**基于机器学习的恶意文档检测与对抗性学习研究**

柯宗贵1，王凤娇2，江纬3，杨育斌4

（1，2，3，4：蓝盾股份，广州 ，510000）

**Abstract -** Nowadays, with the highly rapid development of information technology, it is becoming more and more important to perform detection on malicious documents (such as PDFs). But due to the diversity of the document structure, attackers can gradually have larger attack vector. This research project aims to construct a robust AI document classifier both for industry and academia. The experimental results show that the Accuracy of the model is as high as 99.82% while the False Positive Rate is only as low as 0.01%. More, through the study of adversarial ML, the model has certain capability to resist attacks and enjoys good robustness.

**Key Words：**Information Security; Machine Learning; Maldoc Detection; Adversarial ML

**摘 要：**在当今信息安全领域，基于AI的恶意文档检测越发重要，基于文档的攻击通常具有针对性（targeted attack），加上其文件结构的多样性，攻击手段变得丰富且隐蔽，因此更容易成功。本文的目的是为工业界和学术界提供一个基于AI的恶意文档分类器原型。目前我们收集良性和恶意样本共20万个，经过对恶意样本静态解析，抽取具有显著分类能力的特征，训练生成AI模型。实验数据表明，模型的准确率达到99.82%，而误报率却只有0.01%。进一步地，我们研究了部分对抗性学习的方法，并用实验数据证明模型具有良好的抗逃逸能力和鲁棒性。

**关键词：** 信息安全；机器学习；恶意文档检测；对抗性学习

作者简介：作者1：柯宗贵（1969－至今），男，中国，副董事长兼总经理，学士学位，研究方向：信息安全，广州市天河区科韵路16号信息港A栋20-21楼，邮编：510500，电话：13802736488，kzg@chinabluedon.cn ;

作者2：王凤娇（1992－至今），女，中国云南·大理，研究员，学士学位，研究方向：信息安全，广州信息港，邮编：510500，电话：15602209397，[yonahwang@foxmail.com](mailto:yonahwang@foxmail.com);

作者3：江纬（1986－至今 ），男，广东·广州，研究员，硕士学位，研究方向：信息安全，广州信息港，手机：18802014980，[weijiang2009@gmail.com](mailto:weijiang2009@gmail.com)；

作者4：杨育斌（1974－至今），男，广东·广州，高级工程师，博士，研究方向：信息安全、云计算、移动互联网、网络应急体系，广州信息港，手机：139 2601 3338，电话：（86-20）85526663，Email: yyb@chinabluedon.cn。

1. **简介**

随着时间的推移，PDF规格和样式变得更加丰富。新版本增加脚本的功能使文档与可执行文件几乎能以相同的方式工作，如连接到Internet，运行进程以及与其他程序进行交互等。这种复杂性的增长为攻击者提供了更多的武器来发动攻击，并且能更灵活地隐藏恶意有效载荷逃逸检测。由于企业和个人普遍对此安全意识不足，导致大量系统未采用最新版本更新，最终使这些攻击取得成功。

针对于近几年遇到的多种基于文档的攻击，传统的PDF恶意文件检测方法有基于Shellcode的检测[1]、基于签名的检测[2]等。这些方法均存在无法及时更新，导致无法检测到新的攻击。基于机器学习的方法为此提供了新的方向，机器学习在网页分类技术[3]，入侵检测[4]等方面取得很好的结果，许多研究者也使用机器学习对PDF文件进行分析[5,6]，之后许多研究者基于内容和结构提取文件的静态特征[7]，或是基于元数据与结构提取文件特征[8]进行分类，经过AI算法调优后，可以达到很好的效果。Nedim Srndic等人在后来的研究中关注的不止是模型的预测能力[9]，更多的是AI模型的抗逃逸与鲁棒性，他们针对AI模型的逃逸提出了几种假设和方法[10]，最后以高概率成功逃逸分类器。在我们的实验当中，我们也使用了4种方法来验证我们的模型是否具有抗逃逸的能力，通过对模型进行改进，发现可以检测更多之前成功逃逸的样本，说明我们的模型具备一定的抗逃逸能力。

我们的主要贡献如下：

* 一个PDF文档标签数据集，总样本数201368 个，其中恶意样本173036个，良性样本28332个；
* 精心选取了133个静态特征以用于刻画PDF恶意文档形象，以用于区分恶意与良性样本；
* 模型准确率高达99.82%，误报率0.01%，单个文档预测时间维持在毫秒级别；
* 成功使用生成的恶意文档变异样本对分类器发动逃逸攻击，并促使分类器根据攻击重新训练，得出一个鲁棒性和抗逃逸能力更强的模型。

1. **相关工作**

目前PDF恶意文档检测技术大约可分为两大类：静态分析和动态分析。两种方法在业界均有大量成功应用案例，更高级的解决方案可以把静态和动态分析相结合[11]。表1基于之前的研究列出了对现有方法的总结。

表1 相关工作对比

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 方法 | 分析重点 | 检测技术 | 工作 | 年份 | 外置解析器 | ML | 可检测的不同性 |
| 静态分析 | JavaScript | Lexical 分析 [5] | PJScan | 2011 | Y | Y | Y |
| JavaScript | Token 聚类 [12] | Vatamanu et al. | 2012 | Y | Y | Y |
| JavaScript | API 调用分类 [13] | Lux0r | 2014 | Y | Y | Y |
| JavaScript | Shellcode and opcode 签名 [2] | MPScan | 2013 | N | N | N |
| Metadata | Linearized object path [14] | PDF Malware Slayer | 2012 | Y | Y | Y |
| Metadata | 分层结构检测 [15] | Srndic et al. | 2013 | Y | Y | Y |
| Metadata | 基于结构和Metadata [8] | PDFrate | 2012 | Y | Y | Y |
| Both | 基于结构和内容解析 [7] | Maiorca et al. | 2015 | Y | Y | Y |
| Both | 结合上述几种技术解析分类 [11] | Maiorca et al. | 2016 | Y | Y | Y |
| 动态分析 | JavaScript | Shellcode and opcode签名检测 [1] | MDScan | 2011 | Y | N | N |
| JavaScript | 已知的攻击模式 [16] | PDF Scrutinizer | 2012 | Y | N | N |
| JavaScript | 内存访问模式 [17] | ShellOS | 2011 | Y | N | Y |
| JavaScript | 常见 maldoc 行为分析 [18] | Liu et al. | 2014 | N | N | Y |
| JavaScript | 独立平台的tap point标识技术[19] | tap point | 2016 | N | N | Y |
| 文档类型 | 异常内存访问约束变量 [20] | CWXDetector | 2012 | N | N | N |
|  | 平台多样性 | 系统平台多样性利用[21] | PlatPal | 2017 | Y | N | Y |

由表1可见，静态分析一般聚焦于JavaScript[2]或Metadata[11]，动态分析技术一般聚焦于提取嵌入在PDF文档中的JavaScript代码[19]，再通过实际试运行这些代码片段PlatPal[21]，判定其恶意行为。其中有12/15的工作使用外置的PDF解析器，这使得外置PDF解析器的健壮性成为研究焦点。由表可知，几乎所有的静态分析工作，都使用了机器学习技术如PDFrate[8]。Xu et al[22]通过构建一个能自动生成恶意样本变种的框架，经过遗传算法把良性PDF对象加入到恶意样本的文件结构中混淆分类器，这种专门针对分类器的攻击及其框架被称为分类器逃逸攻击（Classifier Evasion Attack）。

综上所述：对于外置PDF解析器，现有攻击手段是解析器混淆攻击（Parser-Confusion Attacks）；对于机器学习模型，现有的攻击手段是自动化分类器逃逸攻击（Classifier Evasion Attacks）；这些攻击手段对于模型本身的安全提出了很大挑战，我们不仅需生成准确度高的模型，并且在模型健壮性方面也需要有所突破。

# 恶意文档检测器的设计与实现

在本节中，我们聚焦于设计一个基于机器学习的恶意文档检测框架。实验中的数据样本大小约为20万，我们主要对这些文件的内容和结构进行解析，选取具有良好分类效果的特征，然后对提取到的特征用机器学习进行分类。在本节中，我们主要关注以下几个方面：

1). 通用机器学习分类框架

2). 数据集

3). 特征工程

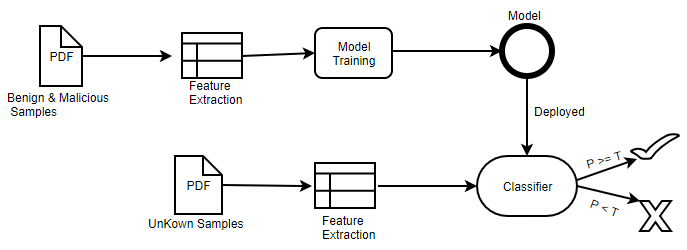
4). 分类算法

## 3.1 通用机器学习分类框架

通用机器学习的分类器框架如图1所示，我们的目标是训练一个健壮性强的模型以用于恶意文档分类。首先，在数据收集阶段，我们需要抓取大量有标签的恶意/良性文档；第二，在特征工程阶段，我们通过对比恶意样本与良性样本的差异，手工提取上百个文档的典型和有效特征；最后，我们使用机器学习来训练模型，其目的在于训练模型拟合训练数据。

我们的训练是离线的，并且需要相对较长的时间和较高的内存占用。预测过程是在线的，模型以服务的形式对外输出预测能力。当新样本提交到模型时，模型在短时间内可返回一个预测分数，分数越高表明模型越有信心预测其为恶意样本。如图1展示了一个基于机器学习的分类器框架。

图1 机器学习的基本框架



## 3.2 数据集

目前使用的数据集共有数据样本201368个，分为良性（28332）和恶意（173036）两大类，其中我们收集到的文件数据有167061个，其中有156035个是从VirusShare下载下来的，大小有约6.8G，另有9000个良性样本来自于Contagio，2026的良性数据集是在搜狗和百度上通过爬虫抓取下来的。我们还使用了mimicus[10]数据集中的2万平衡样本。

我们使用VirusShare的数据样本作为源，使用攻击算法生成对抗样本7000个，以作对抗性学习研究。

图 2. 数据集的时间分布图

图2是数据样本的时间分布图，我们可以看到数据集中有很大部分的样本来自2014年到2017年，同时也说明近几年恶意文档的产生在快速增多。这要求我们的检测引擎对未知恶意文档具有较强的检测能力。

## 3.3 特征工程

有效的特征提取方法主要基于结构、元数据（Metadata）、内容和Javascript。实验数据表明，基于结构的特征具有很好的分类能力。我们通过计算数据集中每一个文档特征的平均值，发现良性样本与恶意样本的特征均值在某些特征中存在明显差异（见表2）。

特征如count\_font与count\_box：在良性样本中会有很多关于font ,box这些对象，而恶意文档一般不把展示信息作为其首要功能，通常是直接把JS恶意代码嵌入到文档当中，以运行恶意代码。在统计同一个页面中obj对象的个数时，良性样本和恶意样本会存在约1倍差距，如果obj在同一个页面中突然增多，此文件为恶意文件的概率大增。

特征如count\_endobj 与count\_endstream：良性PDF样本在每个对象结束时会有一个endobj，但PDF恶意文件为了混淆解析器，会尽可能少地使用endobj和endstream。这就导致解析器在解析恶意PDF文件时不能完整获取整个对象，或导致整个PDF文件解析失败，使恶意PDF文件成功逃逸。这是恶意文件最常使用的逃逸解析器的方法。

表 2: 良性样本与恶意样本之间的特征均值对比

|  |  |  |
| --- | --- | --- |
| Feature | Benign File | Malware File |
| **count\_font** | **14.64** | **0.55** |
| **count\_acroform\_obj** | **700** | **1400** |
| **count\_box\_a4** | **12001** | **200** |
| **count\_box\_legal** | **395040** | **0** |
| **count\_endobj** | **95.80** | **9.68** |
| **count\_endstream** | **30.43** | **3.78** |
| **count\_page\_obj** | **8001** | **16003** |
| **count\_image\_small** | **915892** | **12002** |
| **count\_js** | **0.71** | **1.01** |
| **count\_obj** | **100.96** | **12.01** |
| count\_objstm | 1.57 | 0.15 |

## 3.4 分类算法

AI引擎的重要组成部分之一是算法，我们选取了几个典型算法如SVM，决策树和随机森林进行比较。经过多次训练与分类实验结果如表3，随机森林算法具有准确率高，误报率低，低延时和可解析性等优势，我们故将其选定作为默认算法。随机森林总的来说是一个集成分类器，他使用袋装训练数据（bagged training data）随机选择特征子集，其中一个特征子集表示一棵树，利用这些特征数据确定每棵树的最佳分割点，其中每棵树相对独立，最后通过每棵树的投票来确定最终结果。随机森林算法已被广泛应用于信息安全领域中，如入侵检测算法研究[23]，网络舆情文本信息分类方法研究[24]。

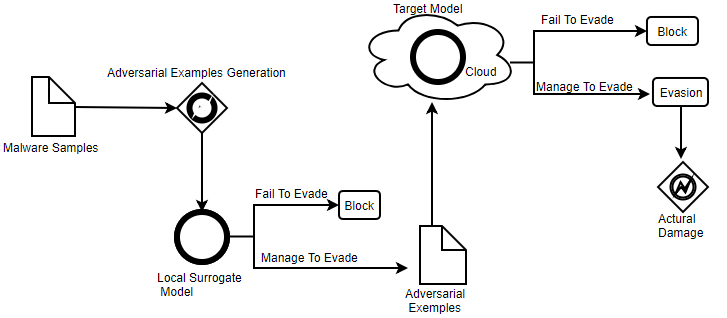
表3：不同机器学习模型的比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | SVM | Decision Tree | Random Forest |
| 检测率 | 75.23% | 82.41% | 99.64% |
| 训练时间 | 58m18s | 4s | 56s |
| 检测时间 | 1.2ms | 0.1ms | 0.1ms |

# 对抗性学习

对抗性学习是机器学习与计算机安全交叉的研究领域[25]。基于机器学习的系统正越来越多地被用于各种恶意数据的检测中[4]。然而，如果模型部署在线上，攻击者可以通过操纵数据（Data manipulation）对模型进行逃逸。

图3.模型逃逸框架



在这个章节中，我们用一个真实的、成功部署线上的model2作为逃逸对象，研究对抗样本的有效性。如图3所示，如果攻击者的目标是逃逸云端模型，他首先会在本地部署一个代理模型（此模型应尽攻击者的能力，与逃逸模型相似），并选取一组具有代表性的恶意样本，利用不同的算法（攻击算法）生成恶意样本，改变文件本身，特征空间等，使其在看起来与良性样本相似但却维持相同恶意行为。我们进而使用这些人工生成的有效的对抗样本去尝试逃逸部署在云端的模型，这背后的假设是对抗样本的可迁移性。云端模型若与本地代理模型相似度高（训练集，特征集，模型参数与算法等），对抗样本则能大概率逃逸此模型，对用户系统造成实在损害。在本节，我们重点关注以下内容：

* 对抗样本生成
* 案例分析（对抗样本迭代原理）
* 有效防御技术
* 模型性能

## 4.1 对抗样本生成

在本节中，我们来讨论特定场景下的对抗性学习。我们假设攻击者已知模型的一些附加信息，如模型所提取的特征，模型的算法等。如果对手知道所有分类器组件的详细信息，他就有最好的机会逃逸检测。在这种情况下，他可以在离线状态完全复制在线分类器，只有在找到足够好的对抗样本时才提交攻击结果。如果逃逸了离线分类器，那么也有很大的可能逃逸在线分类器。

在这里，我们主要参考[10]中所提到的几种场景，对模型进行对抗性学习。F、T、C三个字母分别对应特征集、训练数据集和分类器算法。分别以场景名表示，以下是4种攻击场景对应的信息：

* F（feature）：表示只有特征集可用于攻击者；
* FT（feature and training）：除了已知特征外，攻击者还可以利用目标分类器训练数据集的知识；
* FC（feature and classifier）：攻击者知道特征集以及关于分类器的附加信息，如模型类型，参数或具体实现；
* FTC（all above）：知道特征集，训练集和模型的全部细节。在这种情况下，攻击者可以在线下通过训练完全重现在线分类器，只有在找到足够好的对抗样本时才提交攻击结果。

## 4.2 案例分析

在变异过程中，我们挑选一些典型的样本来做案例分析。一个包含有恶意代码的PDF文件，该文件可以利用漏洞（CVE-2013-0641）远程执行任意代码。我们通过以上四种场景对选取的样本进行变异，然后分别查看样本的VT报告，观察到样本最开始在VT报告中有33个检测引擎可以将其判断为恶意文件，而变异后，识别为恶意文件的引擎减少为22。

表4 样本经过变异后的VT检测结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| File\_HASH | Source | F | FC | FT | FTC |
| 00ba5c43b1cec186c634c24ac21982d3 cve-2013-0641 | 33/61 | 22/60 | 23/60 | 22/60 | 22/60 |

由于大多数的PDF文件检测器是基于结构和内容的，所以只要我们对文件结构和内容做一些改变，比如添加良性样本的一些对象，或改变文件大小等，就可以成功逃逸分类器。于是我们将变异后的文件特征与变异前的文件特征进行比较，如表5所示，变异主要是修改文件的metadata的大小和内容，增加作者信息（author\_lc从0到6，author\_len从0到14），增加count\_javascript的数量（从1到6），同时升级了版本（从4到7）。经过这一系列的改变，样本依然保持其恶意代码，可是已有新增十个分类器不能检测出它的恶意代码。

表5 样本变异前后的特征对比

|  |  |  |
| --- | --- | --- |
| Feature | Source | 变异后 |
| author\_lc | 0 | 6 |
| author\_len | 0 | 14 |
| count\_javascript | 1 | 6 |
| createdate\_ts | -1 | 650616173 |
| moddate\_ts | -1 | 482083775 |
| keywords\_len | 0 | 7 |
| producer\_lc | 0 | 8 |
| version | 4 | 7 |

## 4.3有效防御技术

我们通过分类器找出评分较高的2000个恶意样本作为种子，使用上述的4种方法生成对抗样本，然后使用这些对抗样本来测试Model2。由表6可知，4种攻击方法对于Model2均有较大影响，其中在FC的攻击方法下，Model2对对抗样本的检出率只有2.92%。这促使我们对后续模型2.1、模型2.2和模型2.3和模型3进行探索。

Model 2.1通过增大训练数据集（从10万到20万），以达到增强模型健壮性的目的。由实验数据可知，训练样本数据越多，模型的健壮性越好（在F与FT情况下，模型的对抗样本检出率达到96.7%），同时有研究者认为集合多个分类器可以使模型更稳定[26]；

Model 2.2的改进之处在于在训练过程中添加大量生成的对抗样本。在机器学习中，这种方法叫做对抗性训练（Adversarial Training）。我们把约7000个对抗样本加入到 Model2.2的训练中。实验结果表明，模型2.2准确率有很大的提升（特别是在FC与FTC两种攻击情况下，模型表现出众）。主要原因是该模型在训练阶段已具有对对抗样本进行识别的能力，故只要训练数据集和测试数据集具有相似的数据分布和样本特征，该模型在对抗性学习的测试应该会有良好的性能；

模型2.3的改进是调整阈值。如果我们仍然使用默认的阈值(0.5)，由于对抗样本的平均概率已经大幅度降低，大多数对抗样本将能成功逃逸，对计算环境会造成潜在损害。通过把阈值降低（如设定为0.4），这增加了模型灵敏度（sensitivity），把更多可疑文件作进一步更加全面的分析。

最后，模型3是模型2.1、模型2.2和模型2.3的结合，该模型由一个更大的数据集(包括大量的对抗样本)训练，并为优化检测性能调整阈值。显而易见的，Model3是目前我们评估的最好模型。

表6. 不同攻击场景下模型的准确率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 攻击场景 | 对抗样本数目 | Model 2 | Model 2.1 | Model 2.2 | Model 2.3 | Model 3 |
| **F** | 2000 | 71.18% | 96.71% | 89.43% | 71.03% | **98.65%** |
| **FC** | 240\* | 2.92% | 12.50% | 81.95% | 2.85% | **88.89%** |
| **FT** | 2000 | 84.25% | 96.76% | 98.58% | 85.89% | **98.98%** |
| **FTC** | 2000 | 15.83% | 18.71% | 86.21% | 36.82 | **92.83%** |

\*在理想实验环境下，这个数字应为2000而不是240，但使用240也可以得出同样的结论。

在实验中，我们研究了防御机制的有效性。如果我们的特征集已被攻击者利用，我们可通过改变特征集，如修改权值或删除重要特征等操作，重新训练模型。如图4所示，这是Model3按照特征权重排序的前30个特征，我们可以看到count\_font，count\_javascript，size，count\_obj，count\_endobj这几个特征占有较大权重，同时也很容易被攻击者利用对解析器和分类器逃逸。因此，我们可以在训练时删除这几个已暴露特征，重新训练模型，预测结果如表7 所示。

图4 前30个重要特征排序

表7 是对前5个特征进行依次删除后的模型准确率。当分类器使用全部特征进行训练时，模型准确率达99.82%。当我们将第一重要的特征在训练中删除时，检测率基本没有太大波动，当删减到前5个特征时，模型准确率依旧没有明显变化，维持在高位。

表7 对前5个特征依次删除后的模型准确率

|  |  |
| --- | --- |
| Features Deleted | 准确率 |
| []\* | 99.82% |
| [1] | 99.52% |
| [1,2] | 99.52% |
| [1,2,3] | 99.64% |
| [1,2,3,4] | 99.64% |
| [1,2,3,4,5] | 99.64% |

\*前五个特征分别是(1, count\_font), (2, count\_javascript), (3, size), (4, count\_obj) and (5, count\_endobj)

总的来说，我们针对模型健壮性问题，提出了5种行之有效的防御技术。它们是:

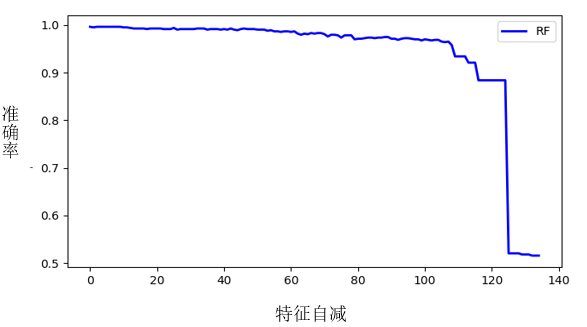
1. 增加多样化和新样本进行训练 (Model 2.1)
2. 对抗性训练 (Model 2.2)
3. 调整模型阈值 (Model 2.3)
4. 及时删除被对手利用的特征并重新训练模型
5. 隐藏分类器的相关参数，如特征，阈值及算法

## 4.4 模型性能

我们通过对特征有效性进行研究，来评估模型的健壮性。图5是特征在不断自删除时所对应的准确率曲线。从图可知，当特征减少至100个时，重新训练后的模型准确率依然高达90%，这说明：

* 单个特征纵然权重高，当此类特征被删除时，模型准确率会下降，但影响不大；
* “中等权重”特征能使模型持续有效，抵御被删除“高权重”特征造成的缺失；

图5 特征自删除后的检测率（模型3）



为了评估模型的预测性能，我们把数据集随机分为训练（90％）和测试（10％）两部分，并采用10-Fold 交叉验证来评估模型有效性。图6为ROC曲线图，曲线下面积约为1，这表明模型具有良好的预测性能。

图6 ROC曲线图



1. **总结**

在本文中，我们详细介绍了基于AI的PDF恶意文档分类器的设计与实现。实验数据表明，在二十万数据集中，我们的模型拥有大于99%的准确率和小于0.01%的误报率。除了研究如何把人工智能应用到恶意文档检测中，也把模型健壮性放在同等重要的位置。我们为此提出了5种增强模型健壮性的防御技术，并使用大量实验证明其有效。

基于AI的恶意文档检测是社会工程、病毒分析等领域的重要研究课题。在未来，我们还会尝试使用动静态分析结合的方法来检测恶意文档，并同时扩展文件检测类型到docx，pptx等。

# 参考文献

1. Zacharias Tzermias, Giorgos Sykiotakis, Michalis Polychronakis, and Evangelos P.Markatos Combining Static and Dynamic Analysis for the Detection of Malicious Documents.[J].In Proceedings of the 4th European Workshop on System Security, 2011, （EUROSEC）(4th):0-15
2. Xun Lu, Jianwei Zhuge, Ruoyu Wang, Yinzhi Cao, and Yan Chen. Deobfuscation and Detection of Malicious PDF Files with High Accuracy[J].In Proceedings of the 46th Hawaii International Conference on System Sciences, 2013,（HICSS）(46th):0-15
3. 孙靖超. 一种基于机器学习的网页分类技术[J]. 信息网络安全，2017，17（9）：45-48.
4. 和湘, 刘晟, 姜吉国. 基于机器学习的入侵检测方法对比研究[J]. 信息网络安全，2018，18（5）：1-11.
5. Pavel Laskov and Nedim Srndic.Static Detection of Malicious JavaScript-Bearing PDF Documents[J]. In Proceedings of the Annual Computer Security Applications Conference, 2011, ACSAC(22):0-15
6. Nedim Srndic and Pavel Laskov.Hidost: a static machine-learning-based detector of malicious files[J]. Eurasip Journal on Information Security, 2016, 22(22):0-15
7. Davide Maiorca, Davide Ariu, Igino Corona, and Giorgio Giacinto. A Structural and Content-based Approach for a Precise and Robust Detection of Malicious PDF Files[J]. In Proceedings of the International Conference on Information Systems Security and Privacy, 2015, ICISSP(1st):0-15
8. Charles Smutz and Angelos Stavrou. Malicious PDF Detection using Metadata and Structural Features[J].In Proceedings of the Annual Computer Security Applications Conference, 2012, （ACSAC）(1):0-15
9. Nedim Srndic and Pavel Laskov. Mimicus: A Library for Adversarial Classifier.[J].IEEE Xplore Digital Library, 2015, IEEE(22):766-777
10. Nedim Srndic and Pavel Laskov. Practical Evasion of a Learning- Based Classifier: A Case Study[J].In Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland), San Jose, CA, 2014, IEEE(35th):0-15
11. DavideMaiorca, DavideAriu, IginoCorona, and GiorgioGiacinto. An Evasion Resilient Approach to the Detection of Malicious PDF Files[J]. In Proceedings of the International Conference on Information Systems Security and Privacy, 2016, ICISSP(1st):0-15
12. Cristina Vatamanu, Dragos GavriluT, and Razvan Benchea. A Practical Approach on Clustering Malicious PDF Documents[J].Journal in Computer Virology, 2012, 1(1):0-20
13. Igino Corona, Davide Maiorca, Davide Ariu, and Giorgio Giacinto. Detection of Malicious PDF-embedded JavaScript Code through Discriminant Analysis of API References[J]. In Proceedings of the Artificial Intelligent and Security Workshop, 2014, AISec(Lux0R):0-15
14. Davide Maiorca, Giorgio Giacinto, and Igino Corona.A Pattern Recognition System for Malicious PDF Files Detection[J].In Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition, 2012, MLDM'12(8th):510-524
15. Nedim Srndic and Pavel Laskov.Detection of Malicious Pdf Files Based on Hierarchical Document Structure[J].In 20th Network and Distributed System Security Symposium, 2013, 20(1):0-0
16. Florian Schmitt, Jan Gassen, and Elmar Gerhards-Padilla. PDF Scrutinizer: Detecting JavaScript-based Attacks in PDF Documents[J].In Proceedings of the 10th Annual International Conference on Privacy, Security and Trust, 2012, （PST）(10th):0-15
17. Kevin ZSnow, Srinivas Krishnan, Fabian Monrose, and Niels Provos. ShellOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks[J].In Proceedings of the 20th USENIX Security Symposium, 2011, （Security）(20th):0-15
18. DaipingLiu, HainingWang, and Angelos Stavrou. Detecting Malicious Javascript in PDF through Document Instrumentation[J].In Proceedings of the 44th International Conference on Dependable Systems and Networks, Atlanta, GA, 2014, （DSN）(44th):0-15
19. Curtis Carmony, Mu Zhang, Xunchao Hu, Abhishek Vasisht Bhaskar, and Heng Yin. Extract Me If You Can: Abusing PDF Parsers in Malware Detectors[J].In Proceedings of the 2016 Annual Network and Distributed System Security Symposium San Diego, 2016, （NDSS）(AN):0-15
20. Carsten Willems, Felix CFreiling, and Thorsten Holz. Using Memory Management to Detect and Extract Illegitimate Code for Malware Analysis.[J].[J].In Proceedings of the Annual Computer Security Applications Conference, 2012, (ACSAC)((1)):0-15
21. Meng Xu and Taesoo Kim, Georgia Institute of Technology. PlatPal: Detecting Malicious Documents with Platform Diversity[J].USENIX, 2017, USENIX(26th):0-20
22. Weilin Xu, Yanjun Qi, and David Evans. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers[J].In Proceedings of the 2016 Annual Network and Distributed System Security Symposium San Diego, CA, February 2016. http:evademl.org, 2016, （NDSS）(1st):0-20
23. 林伟宁, 陈明志, 詹云清, 刘川葆. 一种基于PCA和随机森林分类的入侵检测算法研究[J]. 信息网络安全,2017,17（11）：50-54
24. 吴坚，沙晶.基于随机森林算法的网络舆情文本信息分类方法研究[J].信息网络安全, 2014, 14(11):36-43
25. 顾巧云，孙玉龙，高丰.基于博弈论的网络攻防对抗模型及应用研究[J].信息网络安全, 2013, 13(1):0-0
26. Charles Smutz, Angelos Stavrou . When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. In Proceedings of the 2016 Annual Network and Distributed System Security Symposium San Diego, 2016, （NDSS）(AN):11-26