**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. Around 200,000 samples have been collected and the AI model have been trained and optimized. 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. At last, we demonstrate our model can be widely deployed in typical scenarios such as security products or mail servers.

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

**摘 要：**在当今信息安全领域，基于AI的恶意PDF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%，单个文件的平均检测时间仅需几毫秒。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。最后，我们通过实际应用，表明此模型可广泛部署在终端安全产品，邮件服务器上等，这些均是非常有意义的应用场景。

**关键词：** 人工智能之于信息安全；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，科韵路16广州信息港，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，广州市天河区科韵路16号信息港A栋20-21楼，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，工程师，硕士学位，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，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，广州市天河区科韵路16号信息港A栋20-21楼，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文件/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，并逃逸检测。由于企业和个人普遍对此类安全漏洞反应迟缓，安全意识不足，导致大量的用户系统未采用最新版本进行更新，最终使这些攻击取得成功。

在2014年发现的24个0-day中，有16个是针对Adobe Reader和Flash Player的。在通用漏洞与披露（CVE）也可以明显观测到，从2015年开始关于Adobe Reader发现的漏洞呈现高增长态势，这给基于PDF的文档攻击敞开了大门。

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[15]、基于签名的检测[13]等。这些方法均存在识别率不高、无法及时更新恶意代码等普遍问题。基于机器学习的PDF恶意软件检测为此提供了崭新的方向，最开始使用机器学习的方法是2011年Nedim Srndic等人，主要对javascript进行提取分析，之后很多研究者基于内容和结构提取文件的静态特征[8],或是基于元数据与结构提取文件特征[24]使用SVM和决策树对文件进行分类，经过AI算法调优后，可以达到很好的效果。我们的方法结合他们所提到的一些特征进行提取，包含有结构，内容，javascript,元数据信息等。并通过研究发现，我们使用随机森林的算法比使用SVM 算法的准确率要高。同时我们也考虑到了AI模型的安全问题，Nedim Srndic等人在后来的研究中关注的不止是模型的准确率，而是AI模型的抗逃逸与鲁棒性，他们在IEEE会议上[4]针对AI模型的逃逸提出了几种假设，最后成功逃逸分类器，在我们的实验当中同时也使用了其中的4种方法来验证我们的模型是否具有这种抗逃逸的能力，我们通过对模型的一些特征选取和算法调优，发现可以使之前逃逸的一部分样本被检测出来，说明我们的模型有一定的抗逃逸能力。

本文的主要贡献如下：

* 一个PDF数据集，总样本数201368 个，其中恶意样本173036个，正常样本28332个；
* 精心选取了一套静态特征集（133个）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%，单个文件检测时间维持在毫秒水平；
* 单个文件预测时间维持在毫秒级别
* 成功使用自己生成的变种病毒对分类器发动逃逸攻击，分类器根据攻击进行自我修复，重新训练得出一个鲁棒性强，抗逃逸能力强的模型。

1. **相关工作**

如今PDF恶意文档检测技术可大约分为两大类：静态分析和动态分析。静态分析无需使样本运行，仅通过文件头部格式，二进制层N-gram等静态模式，即可对目标样本进行预测；动态分析，则通过使目标样本运行于受控环境内，以此捕捉其恶意行为。一般而言，静态分析的优点是速度易于部署更新，动态分析速度慢，消耗资源多，但人员参与精度很高。两种方法在业界均有大量成功应用案例，更高级的解决方案如可以把静态和动态分析结合，典型的工作如Maiorca et al. [9]。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 [7] | Lux0r | 2014 | Y | Y | Y |
| JavaScript | Shellcode and opcode 签名 [13] | MPScan | 2013 | N | N | N |
| Metadata | Linearized object path [11] | PDF Malware Slayer | 2012 | Y | Y | Y |
| Metadata | 分层结构检测 [1] | Srndic et al. | 2013 | Y | Y | Y |
| Metadata | 基于结构和Metadata [24] | PDFrate | 2012 | Y | Y | Y |
| Both | 基于结构和内容解析 [8] | Maiorca et al. | 2015 | Y | Y | Y |
| Both | 结合上述几种技术解析分类 [9] | Maiorca et al. | 2016 | Y | Y | Y |
| 动态分析 | JavaScript | Shellcode and opcode签名检测 [15] | 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[20] | tap point | 2016 | N | N | Y |
| 文档类型 | 异常内存访问约束变量 [19] | 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本身或使用Metadata进行分析。代表性的检测技术有基于Shellcode和OPCode签名的MPScan[13]、基于结构与内容两者的分类[9]。动态分析技术一般聚焦于提取嵌入在PDF文档中的JavaScript代码，再通过实际试运行这些代码片段，检测出恶意行为。这类工作的代表有基于maldoc的行为分析[20]和基于平台多样性的PlatPal[21]等。

在以上工作中，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。这是因为外置PDF解析器一般设计和实现均较为简单，恶意样本经少量变异即能轻易逃逸此类解析器。这种攻击在Carmony et al.[20]的工作中被称为解析器混淆性攻击（Parser Confusion Attacks）。

从表1可知，机器学习一般并不适合于动态分析，而几乎所有的静态分析工作，都在某种程度上使用了机器学习的技术。这部分的典型工作有PDFrate[24]、PDF Malware Slayer[11]等。这些工作均声称分类器能在低功耗环境下达到很高的检测精度，但对模型本身的安全性，恶意样本逃逸分类器等对抗性学习的研究内容却鲜有提及。这种攻击在Xu et al[14]的工作中被提出，作者通过构建一个能自动生成恶意样本变种的框架，使得在每一次的样本变异迭代过程中，原始输入样本集会经过某种遗传算法把良性PDF对象加入到恶意样本的文件结构中。在不断的变异过程中系统一方面需保持恶意样本的恶意本来面目不变，另一方面则需要达到迷惑分类器的目的。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）

在上述工作中，有11/15的工作有如下假设：即恶意样本和良性样本间需具有明显特征分辨能力或分界线。换句话说，我们假设恶意样本和良性样本有一超平面能把其很好地在高纬度特征空间中分开。一些有趣的研究性问题是这样的：是否可以通过不改变原文件的恶意属性，用增加良性行为部分的方法，以成功逃逸分类器的检测？是否可以通过不改变原文件的善意行为，用增加恶意行为的方法，使这些样本通过隐藏方式，成功逃逸分类器。Srndic et al. [4]的工作从恶意样本着手，聚焦于前一种攻击，我们把其称为模拟性攻击（Mimicry Attack）；而Maiorca et al.[10]的工作从良性样本着手，我们在这里称其为反向模拟性攻击（Reverse Mimicry Attacks）。

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（Automatic Classifier Evasion Attacks）；对于假设性的“可检测的分辨力”(Detectable Discrepancy)，现有攻击手段为模拟和反向模拟攻击（Mimicry and Reverse Mimicry）。这些攻击手段对于模型本身的安全提出了很大挑战，在我们的工作中不仅生成了一个准确度高的模型，并且在模型本身的安全也有所建树。

# 恶意文档检测器的设计与实现

在本节中，我们集中对分类器的设计，一个基于机器学习的恶意文档检测框架。实验中我们采用的数据有20W，其中包含了所有PDF的文件类型。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的方法进行分类。我们主要关注分类器的以下几个方面：

1). 通用机器学习分类框架

2). 数据集

3). 特征工程

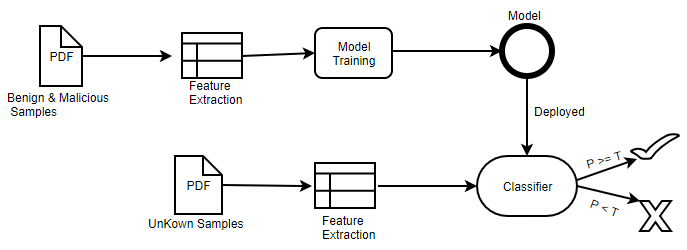
4). 分类算法

5). 模型迭代

## 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鲁棒性的模型用于恶意文档分类，首先，在数据收集阶段，我们需要收集到大量有标签的恶意文件和良性文件；第二，在特征工程阶段，我们通过对比恶意文件与良性文件的差异，手动提取上百个文件的典型特征，这是一个将文档从原始向量转换为特征向量；最后，我们通过机器学习来训练模型，使模型可以很好的拟合训练数据分布。训练通常是离线的，并且需要很多的时间和内存占用，当预测是就可以打包到在线服务进行预测。在这时，我们的模型已经准备好预测了，当新的样本提交到了模型中，它就可以返回出自信的预测分数，来判定这个样本是恶意的还是良性的。如图1 很好的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

目前收集的数据一共201368个，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。

通过我们对对抗性学习的研究，使用VirusShare为源样本，又生成了7000个对抗样本，在最后的试验中用于测试。

还有mimicus中的数据集主要用于PDFRATE实验性评估，可供下载[4]。mimicus 开源数据集有2万的平衡样本，包括来自于contagio的5,000个良性样本和5,000个恶意样本，以及来自于google的5000个良性样本和VirusTotal的5000个恶意样本。

图 4. 数据集的时间分布图

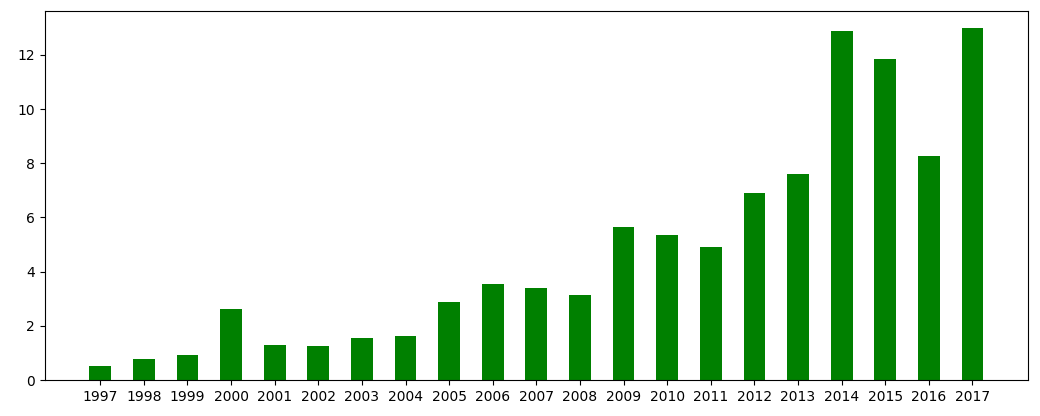


图4是数据集的时间分布图，我们可以看到数据集中有很大部分的样本来自2014-2017，同时也说明随着时间的增长，恶意文件的产生也在逐渐的增多，这就要求我们检测引擎对未知的文件具有一定的检测能力。稍后，我们可以看到我们的ML技术可以有效地对抗多样化的数据集，特别是对于不可见的样本，这证明了我们所提出的技术的通用性和预测能力。

## 3.3 特征工程

有效的特征提取方法主要基于结构、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。我们通过计算样本集中每一个文件特征的平均值，发现正常样本与恶意样本的特征均值在某些特征中存在明显差异差（具体见表2）。

特征如count\_font、count\_box：在正常样本中会有很多关于font ,box这些对象，是因为PDF文件主要功能在于用这些对象来描述信息。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。

特征如count\_page\_obj和count\_obj：一般来说，良性样本的obj对象比恶意样本多很多，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，或者导致整个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。

特征如count\_js:恶意文件的主要攻击手段是嵌入JS代码来执行恶意行为。因此，一个恶意文件所含JS代码量会比良性样本的代码量多。还有一部分用于混淆和加密的JS的大小与良性样本间也存在一定的差异。

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\_box\_letter | 7291529 | 866773 |
| count\_box\_other | 32.18 | 1.74 |
| count\_box\_overlap | 1000 | 0 |
| **count\_endobj** | **95.80** | **9.68** |
| **count\_endstream** | **30.43** | **3.78** |
| **count\_page\_obj** | **8001** | **16003** |
| count\_image\_large | 110711 | 400 |
| count\_image\_med | 465247 | 6401 |
| **count\_image\_small** | **915892** | **12002** |
| count\_image\_total | 36.56 | 0.30 |
| count\_image\_xlarge | 300 | 0 |
| count\_image\_xsmall | 21.64 | 0.11 |
| **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，我们选取了几个实用性较好的算法来比较如表3，这几个算法包括SVM支持向量机，决策树和RF随机森林，经过多次的训练与分类实验，发现随机森林准确率高，误报率低，低延时鲁棒性良好，和可解析性等优势，于是我们将其选定作为默认算法。并且经过我们的一百多次的验证，在特征发生改变的时候，随机森林准确率依然趋于一个稳定的值。

随机森林分类方法给出的结果是基于很多棵分类树判断结果的集合，每一个决策树都是在训练数据中随机选择生成的。因此，随机森林总的来说是一个集成分类器，他使用 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，每棵树都如此来确定树的一个节点，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。

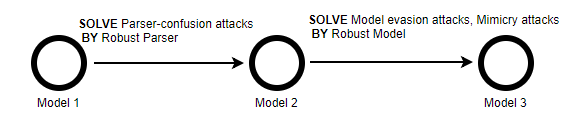
表3：比较不同机器学习的模型

|  |  |  |  |
| --- | --- | --- | --- |
|  | SVM | Decision Tree | Random Forest |
| Accuracy | 75.23% | 82.41% | 99.64% |
| Training Time (the whole dataset) | 58m18s | 4s | 56s |
| Prediction Time (each sample) | 1.2ms | 0.1ms | 0.1ms |

## 3.5 模型迭代

在实验过程中，我们主要对模型进行了两次主要的迭代，每个模型都提供了文件的恶意程度的概率值，即，区间[0,1]中的实际值。我们在预测时默认的阈值(0.5)，但会相应地调整阈值。由表5 可以清晰的看出来，两次模型模型更新解决的一些问题。其中Model 2更新时解决了解析器攻击，使用了鲁棒性强的解析器。Model 3 更新时，提出了一个鲁棒性强的分类器，主要针对模拟攻击和模型逃逸，详见本文第四章。

表 5. 模型迭代



Model 1 : 此模型使用peepdf为解析工具(https://github.com/jesparza/peepdf)，然后通过一些特征计算与特征量化，使其可以用于机器学习训练与预测，当我们的特征数提取到133个的时候，这些特征包括有基于结构的（count\_font、size、count\_startxref），内容信息的（title\_oth、subject\_lc）和metadata(producer\_oth、producer\_len)等一些静态属性,并通过多次实验比较选取性能和精度较好的算法(随机森林RF)。但这样存在问题：由于在一开始使用peepdf进行解析的时候，只有一半的文件可以被解析到。换句话说就是，有一半的文件没有经过分类器，直接逃逸。

Model 2 : 在Model 2 中，重新选取了解析器mimicus[2]，这个工具可以解决之前因为结构缺陷或混淆而不能正常解析的问题，我们使用了mimicus 对之前的文件进行解析，对所有的数据（20万）均能正常解析，一共提取特征135个。在Model 2的训练中，随机抽选平衡的数据集进行训练与预测，其中包含2万恶意样本与2万良性样本。主要算法还是使用准确率较高的随机森林。通过参数调优后，使Model2的检测率提高到99.99%，误报率降低为0.012%。当数据集增大到十级别的时候，准确率下降了0.18%，这是因为测试数据集也随之增加，文档分类与代码嵌入的方式也会更多，这时就考虑到模型的健壮性。我们会在下一章中讨论关于模型健壮性的问题。

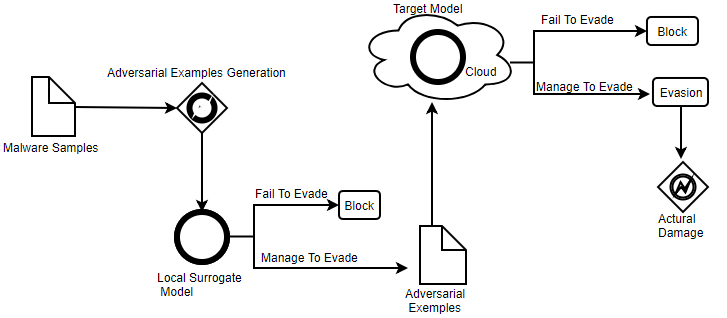
Model 3 : 该模型与前两个模型最大的不同点就是鲁棒性。针对于模型解析与模型逃逸两方面都做了改进，并使用对抗性样本对模型进行攻击后，做了相对应的防御策略，最后模型不仅对一般的文件具有良好的分类效果，对于那些混淆的，对抗样本也具有一定的分类效果。我们提出了5钟有效的防御策略，具体方法在下一章节会详细介绍。

# 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[28]。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。然而，如果模型部署在线上，攻击者可以通过操纵数据（Data manipulation）对其进行逃逸。此类攻击在以前的工作中也有所研究，但其假设是攻击者对所部署的分类器有100%的知识（full knowledge）。在实际中，这种假设是极少成立的，特别是对于部署在线的系统。

在这个章节中，我们用一个真实的、部署成功的model2作为测试用例，研究分类器逃逸技术的有效性。如图6所示，在实验中选取了具有代表性的恶意样本，这些恶意样本在本地分类器中均被判断为1 。利用不同场景中对应的攻击算法，改变文件的特征空间，使其变得更像良性样本。然后在本地的训练模型进行预测，当预测为1，说明Fail to evade；当预测为0 ，说明Manage to evade ,也就是说生成的对抗样本已经成功逃逸了本地分类器，我们就可以使用这些已经逃逸的样本去测试我们部署在云端的分类器。云端分类器就会预测出两种结果，一种是Fail to evade ,另一种是manage to evade,前者直接被Block,而后者就会逃逸，最后对系统造成Actural Damage。

图6.模型逃逸框



我们为实际逃逸策略建立了一套科学体系，并且适配了一些逃逸算法用于实际的应用场景中。同时研究了一些潜在的面对分类器逃逸攻击的防御策略。实验结果表明，模型2即使在受到简单的攻击时，其检测准确率就有明显的下滑。研究结果表明。我们提出的5种潜在的技术可以增强模型的鲁棒性，以对抗数据操纵。以下是本章的研究场景和对抗策略：

* 对抗样本生成
* 案例分析（对抗样本迭代原理）
* 潜在的防御技术
* 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。具体来说，我们假设攻击者已知模型的一些信息，如模型所提取的特征，模型的算法等。当攻击者知道模型的信息越多，他所设计的逃逸样本会越容易逃逸。在这里，我们主要参考Nedim Smdic [4]中所提到的几种场景，对模型进行对抗性学习。

为了有效地探索攻击者的攻击思路，我们定义了一组反映不同程度的正交规避策略。F、T、C三个字母分别对应于特征集、训练数据集和分类器算法，分别以场景名表示。我们描述了在以下4种攻击场景：

* F（feature）：表示只有特征集可用于敌手；
* FT（feature and training）：除了已知的特征外，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一些细节，例如类型，参数或具体实现；
* FTC（all above）：如果知道所有分类器组件的细节，在这种情况下，攻击者可以在线下完全重现在线分类器，只有在找到足够好的规避样本时才提交攻击结果。

如果对手知道所有分类器组件的详细信息，他就有最好的机会避开检测。在这种情况下，他可以在离线状态中完全复制在线分类器，只有在找到足够好的规避样本时才提交攻击结果。如果击败了离线分类器，那么也有很大的可能击败在线分类器。

## 4.2 案例分析

在变异过程中，我们精心挑选一些典型的样本来做案例分析。我们选取了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报告中可以被61个检测引擎分析到，其中有33个检测引擎可以将其判断为恶意文件，而经过不同的方法变异后，可解析的引擎减少至60，可识别为恶意文件的引擎减少为22。

表5 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6所示，我们可以看出，变异主要是改变了文件的metadata的大小和内容，增加了Count\_javascript的数量、一些Keywords的内容，并且增加的都是良性样本的对象，同时还对其版本进行了修改。经过这一系列的改变，样本依然保持有其恶意代码，可是已经有十个分类器不能检测出它的恶意代码。

表6 样本变异后的特征对比

|  |  |  |
| --- | --- | --- |
| Feature | Source | 变异后 |
| author\_lc | 0 | 6 |
| author\_len | 0 | 14 |
| author\_uc | 0 | 6 |
| count\_javascript | 1 | 6 |
| createdate\_ts | -1 | 650616173 |
| createdate\_tz | -1 | 10020 |
| moddate\_ts | -1 | 482083775 |
| keywords\_lc | 0 | 4 |
| keywords\_len | 0 | 7 |
| producer\_lc | 0 | 8 |
| producer\_len | 0 | 19 |
| version | 4 | 7 |

## 4.3 潜在的有效防御技术

我们框架中最重要的部分之一是对抗样本生成算法。目标是生成PDF文件，其特征向量可能获得较低的分类分数。

我们通过分类器找出评分较高的2000个恶意样本作为病毒母体，使用上述的四种方法生成可PDFrate、且依然保持有恶意行为的恶意样本变种，然后使用这些变种来攻击Model2。由表4我可以观测到 这种攻击方法对于Model2有很大的影响，其中在FC的攻击方法下，Model2对变种的准确率只有2.92%。就是说有90%以上的恶意文件通过变异后逃逸分类器。这促使我们对模型2.1、模型2.2和模型2.3进行了开发，最后从防御方面对模型3进行了开发。

Model 2.1 通过将数据集的大小从10万增加到20万训练。与模型2相比，增加了样本多样性与数据量后，模型2.1相对于4种攻击场景都有显著的改进。一定程度说明训练样本数据越多，模型的通用性就约好。

Model 2.2 改进的地方在于添加了大量的对抗性样本训练。在机器学习中，这一种方法叫做对抗性训练。我们使用大约7000个对抗样本加入到 Model 2.2的训练中。由表6我们可以看出，Model 2,2在测试新的对抗样本时，准确率有很大的提升 ，特别是在FC和FTC这两种场景中。主要原因是该模型在训练阶段具有对敌对样本进行判别的经验，只要训练集和测试集具有相同的数据分布，该模型在测试中就会有良好的性能。

模型2.3的改进是其调整的阈值。如果我们仍然使用默认的原始阈值(0.5)，大多数对抗性的例子将被忽略，从而增加了误报率。通过手工将检测阈值从0.5降低到0.4，我们增加了模型的灵敏度，以包含更多可疑的文件以进行进一步的研究。

最后的模型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(包括大量的对抗式例子)训练，并为优化性能调整了阈值。显而易见的，Model 3比我们目前评估的所有模型都要好。在3种防御技术中，技术对抗训练的贡献最大也就不足为奇了。

表6. 不同攻击场景下模型的准确率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attack Scenarios | Adversarial Examples | 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%** |

\*在理想的实验环境下，这个数字应该是2000而不是240，但是240也可以得出同样的结论。

## 4.4 模型性能

在实验中，我们研究了防御机制对规避技术的鲁棒性。提出了5种增强机器学习模型鲁棒性的防御技术。

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(feature set)，如修改权值或删除重要特征等操作，重新训练模型。如图2 所示，是Model3训练后，按照特征重要性排序的前30个特征，我们可以看到count\_font，count\_javascript，size，count\_obj，count\_endobj这几个特征在分类中占有较多的权值，同时也是非常容易被攻击者利用，来对解析器和分类器进行逃逸，于是我们在训练时就删除了这几个特征，然后重新训练Model3，预测结果如表7 所示。

图2 前30个重要特征分布图



表7 是对前5个特征修改后的Model3准确率。由表中可知，当分类器使用全部特征进行训练时，模型准确率高达99.82%。当我们将第一个重要的特征在训练的时候删去，检测率基本没有太大的波动，当减到count\_endobj前5个特征时，模型准确率的波动可忽略不计。这也说明了我们的模型可以对抗一些基于特征的攻击，即使对手知道我们分类器使用的特征，模型同样可以达到99%的精度 。

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)

同时我们还通过对特征有效性进行研究，来评估模型的鲁棒性。我们将模型的特征进行重要性排序，然后依次将最重要的特征逐一删减，并重新使用新特征集重新训练模型。图3是特征在不断自减时所对应的准确度曲线。从图中可知，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，模型准确度会下降，但降幅不大；
* “中等权重”特征的互相作用和叠加，可以使模型健壮，且抵消单个重要特征的缺失影响；
* “中等权重”特征能有效抵御通过改变特征数值的分类器逃逸攻击；

图3 特征自减后的识别率（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(Cross Validation）的方法来评估模型。 图4为ROC曲线图，由图可知，ROC曲线下的面积约为1，这表明Model3具有良好的预测性能。

图4 ROC曲线图



总的来说，我们针对模型鲁棒性问题，提出了5种增强模型鲁棒性的防御技术。它们是:

1. 增加多样化和最新的样本进行训练 (Model 2.1)
2. 对抗性训练 (Model 2.2)
3. 调整模型的阈值 (Model 2.3)
4. 删除被对手利用的特征训练模型 (如Model 3的图3)
5. 隐藏分类器的相关参数，比如特征，阈值，算法

1. **应用实例：AI防火墙**

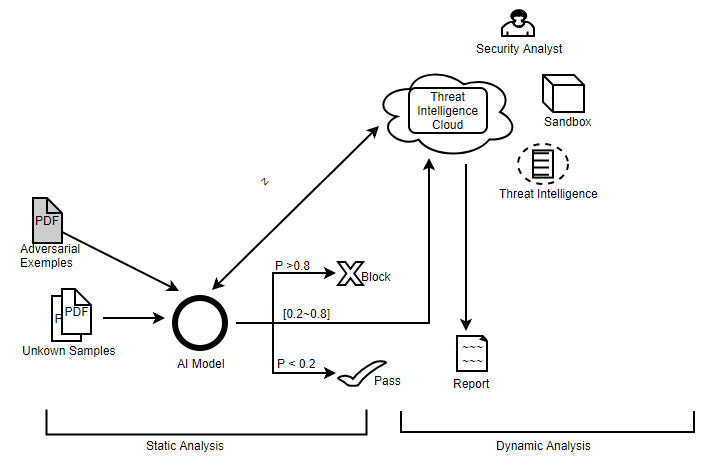
按照模块化的思想，我们把基于AI的文档分类器作为一独立检测模组，集成到边界安全产品如防火墙中。一个拥有30多年历史的老式安全产品，该如何为其插上AI的翅膀？

在当今边界安全产品中（如防火墙），对于网络应用层（第七层）的恶意文件扫描已经是国际标准。业界对此功能的需求极为严苛，一个优秀的功能模组通常要求单个文件检测（扫描）时间（即延迟）维持在毫秒级别，对文件的检测准确率需在99%以上，而误报率则不能大于0.01%。

低延迟的功能要求是因为文件扫描功能需串联于整个文件检测的流水线中，高延迟会显著增加丢包率，造成严重数据丢失，这对于安全设备是不允许的。这些年，随着恶意文档的急速增加，先前基于规则匹配的引擎从标配开始变得力不从心：一方面为了达到产品需求的高准确度，需要大量的安全分析人员编写规则以更新之前的规则库；另一方面随着核心规则库的扩大，规则匹配算法的时间也呈现指数级别增长。这两方面都促使我们探索更新更好的AI引擎，并把此技术运用于实际工程中。

综上所述，我们把基于AI的PDF分类器串联集成到防火墙中。纵然上述两引擎均基于静态分析技术，AI引擎带来的优势却是非常明显的。一方面，此引擎本无需频繁更新，据实验数据，正常AI引擎的平均更新周期为半年，而规则引擎则为2周；第二方面是AI引擎卓越的低计算资源消耗，据实验表明，AI引擎能在预测时稳定于约1/3的CPU占用，约50%的内存占用。CPU计算力的消耗主要在特征提取和计算最终结果概率上，内存的消耗则来自AI模型自身在预测时需位于内存中。

Figure 10. Overview of Static & Dynamic analysis



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，在十万级文档数据集中，我们的模型均大于99%的准确率和小于0.01%的误报率。且在实际运行时，CPU和内存的时空效能比（Time & Space Performance）比旧有基于规则的模型，有显著提升。

本文除了使用大量数据研究人工智能化应用安全，并且把人工智能本身的安全也放在了同等重要的位置上。我们通过大量的实验，模拟了（1）攻击者通过对恶意样本的增删改（如变更特征的值），以混淆分类器，达到逃逸的目的；（2）分类器经自身修正，通过重新训练模型，去除已被攻击者所利用的特征，以维持模型的健壮性。

基于AI的文档分类器是社会工程学、病毒分析等领域的重要研究课题。在未来，我们还会尝试解决以下研究问题：

* 基于深度学习的恶意PDF文档检测
* 动静态分析引擎的调优
* 对于Microsoft Office等其他文件格式的支持（如docx，pptx等）

# 参考文献:

1. Nedim ˇ Srndic and Pavel Laskov. Detection of Malicious Pdf Files Based on Hierarchical Document Structure. In 20th Network and Distributed System Security Symposium (NDSS), 2013
2. Nedim ˇ Srndic and Pavel Laskov. Mimicus: A Library for Adversarial Classifier Evasion. [https://github.com/srndic/ mimicus](https://github.com/srndic/mimicus).
3. Nedim Šrndic and Pavel Laskov . Hidost: a static machine-learning-based detector of malicious files, Šrndi′c and Laskov EURASIP Journal on Information Security (2016) 2016
4. Nedim Srndic and Pavel Laskov. Practical Evasion of a Learning- Based Classifier: A Case Study. In Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland), San Jose, CA, May 2014
5. Pavel Laskov and Nedim Srndic. Static Detection of Malicious JavaScript-Bearing PDF Documents. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), 2011
6. Davide Balzarotti, Marco Cova, Christoph Karlberger, Christopher Kruegel, Engin Kirda, and Giovann Vigna. Efficient Detection of Split Personalities in Malware. In Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February–March 2010
7. Igino Corona, Davide Maiorca, Davide Ariu, and Giorgio Giacinto. Lux0R: Detection of Malicious PDF-embedded JavaScript Code through Discriminant Analysis of API References. In Proceedings of the Artificial Intelligent and Security Workshop (AISec), 2014.PDFrate
8. Davide Maiorca, Davide Ariu, Igino Corona, and Giorgio Giac- into. A Structural and Content-based Approach for a Precise and Robust Detection of Malicious PDF Files. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*, 2015.
9. DavideMaiorca, DavideAriu, IginoCorona, andGiorgioGiacinto. An Evasion Resilient Approach to the Detection of Malicious PDF Files. In Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP), 2016.
10. Davide Maiorca,IginoCorona,andGiorgioGiacinto.Lookingat the Bag is not Enough to Find the Bomb: An Evasion of Structural Methods for Malicious PDF Files Detection. In Proceedings of the 8th ACM Symposium on Information, Computer and Commu- nications Security (ASIACCS), Hangzhou, China, March 2013.
11. Davide Maiorca,Giorgio Giacinto,and Igino Corona. APattern Recognition System for Malicious PDF Files Detection. In *Pro- ceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)*, 2012.
12. Cristina Vatamanu, Drago¸s Gavrilu ¸ T, and R˘azvan Benchea. A Practical Approach on Clustering Malicious PDF Documents. Journal in Computer Virology, June 2012.
13. Xun Lu, Jianwei Zhuge, Ruoyu Wang, Yinzhi Cao, and Yan Chen. De-obfuscation and Detection of Malicious PDF Files with High Accuracy. In *Proceedings of the 46th Hawaii International Con- ference on System Sciences (HICSS)*, 2013.
14. Weilin Xu, Yanjun Qi, and David Evans. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers. In Proceedings of the 2016 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2016. http://evademl.org/
15. Zacharias Tzermias, Giorgos Sykiotakis, Michalis Polychronakis, and Evangelos P. Markatos. Combining Static and Dynamic Analysis for the Detection of Malicious Documents. In Proceedings of the 4th European Workshop on System Security (EUROSEC), 2011.
16. Florian Schmitt, Jan Gassen, and Elmar Gerhards-Padilla. PDF Scrutinizer: Detecting JavaScript-based Attacks in PDF Docu- ments. In *Proceedings of the 10th Annual International Confer- ence on Privacy, Security and Trust (PST)*, 2012.
17. Kevin Z. Snow, Srinivas Krishnan, Fabian Monrose, and Niels Provos. ShellOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks. In *Proceedings of the 20th USENIX Security Symposium (Security)*, San Francisco, CA, August 2011.
18. DaipingLiu,HainingWang,andAngelosStavrou.DetectingMa- licious Javascript in PDF through Document Instrumentation. In *Proceedings of the 44th International Conference on Dependable Systems and Networks (DSN)*, Atlanta, GA, 2014.
19. Carsten Willems, Felix C. Freiling, and Thorsten Holz. Using Memory Management to Detect and Extract Illegitimate Code for Malware Analysis. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), 2012.
20. Curtis Carmony, Mu Zhang, Xunchao Hu, Abhishek Vasisht Bhaskar, and Heng Yin. Extract Me If You Can: Abusing PDF Parsers in Malware Detectors. In Proceedings of the 2016 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2016
21. Meng Xu and Taesoo Kim, Georgia Institute of Technology:PlatPal: Detecting Malicious Documents with Platform Diversity . 26th USENIX Security Symposium 2017
22. VirusTotal. Free Online Virus, Malware and URL Scanner.https://www.virustotal.com/.
23. Stephan Chenette. Malicious Documents Archive for Signature Testing and Research - Contagio Malware Dump. http://contagiodump.blogspot.de/2010/08/ malicious-documents-archive-for.html.
24. Charles Smutz and Angelos Stavrou. Malicious PDF Detection using Metadata and Structural Features. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), 2012.
25. Symantec 2017年安全威胁报告https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf
26. M.Polychronakis,K.Anagnostakis,andE.Markatos.Com- prehensive shellcode detection using runtime heuristics. In Annual Computer Security Applications Conference (AC- SAC), pages 287–296, 2010.
27. Charles Smutz, Angelos Stavrou . When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. C Smutz, A Stavrou - NDSS, 2016
28. Adversarial Machine Learning. <https://en.wikipedia.org/wiki/Adversarial_machine_learning>
29. 例如文伟平，郭荣华，孟正，等. 信息安全风险评估关键技术研究与实现[J]. 信息网络安全，2015，15（2）：7-14.
30. 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.
31. 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.
32. 张家旺, 李燕伟. 基于N-gram算法的恶意程序检测系统研究与设计[J]. 信息网络安全，2016,16（8）：74-80.
33. 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
34. 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
35. 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，