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;
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；
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；
- 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。

2. 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]	LuxOr	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。

3. 恶意文档检测器的设计与实现

在本节中，我们展示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。实验结果表明，通过我们提取的特征和分类方法，可以使模型准确率在 99%以上，同时误报率控制在 0.01%内。

3.1 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。

3.2 特征提取

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。

还有一个重要的差异是特征 count_acroform_obs：在 PDF Specification 1.2 中引入 AcroForm。这种表单从用户处通过交互方式收集信息。表单支持包括数据表示、数据捕捉和数据编辑等功能。它还可以进行动态交互，从具有动态计算、验证及其他特性的交互式、可编辑的表单，到由服务器生成的、机器填充的表单等。同时动态布局表单可以自动重新调整自身以适应用户或外部数据源（如数据库服务器）提供的数据。基于以上几个特点，表单很容易成为攻击者混淆和加密的地方，故在计算 AcroForm 值的时候，恶意样本比正常样本高约一倍。

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

3.3 分类算法

首先对收集到的文件进行分类，将提取出来的特征作为训练数据集，在这个时候，随机森林（random forests）在分类上表现了很好的优势，有效且误报率极低，并易于使用，可以很快的对数据进行分类。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展现，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, 通过随机选择的特征子集,，并使用该节点的训练数据，确定每个节点处的最佳分割来创建树中的每个节点。此外，每棵树都是基于一个独立的特征子集，最后，在分类过程中每一个树的投票来确定最终结果。

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，这几个算法包括 KNN 邻近算法，NNET 神经网络，RF 随机森林和 SVM 支持向量机，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。

表 3　算法准确率对比

SVM	NNET	KNN	RF
75.23%	82.41%	97.12%	99.64%

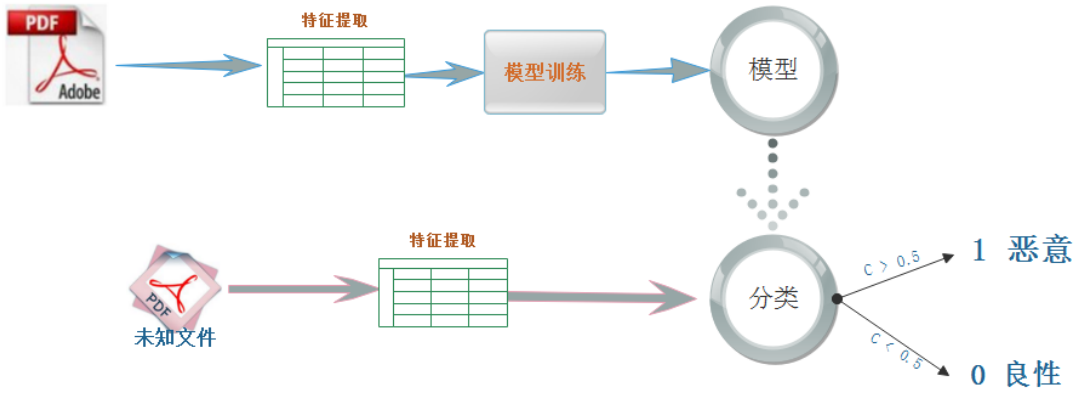
3.4 模型构建

所提出的基于机器学习的恶意 PDF 文档检测的方法包括以下两个步骤，如图 1 所示：

1. 提取文件特征。此为基本的预处理步骤，对 PDF 文件的结构、内容和元数据进行解析，并做相应的向量计算，提取为一个二维的特征集，使得这些特征可以进入到基于机器学习的模型中进行训练分类。

2. 学习和分类。我们会随机选取数据的 80%进行训练，通过训练之后保存训练模型，然后使用 20%的文件进行预测分类，从中计算出模型的准确率，误报率等信息。

图 1 机器学习的基本框架



在我们的实验当中，主要对模型进行了 4 次更新，其中最开始的模型（model1）是使用 peepdf 为解析工具，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：经过我们的研究发现，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，所以我们重新选取了解析器 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，并做特征提取。

在模型 2 的训练中，我们在总的数据集中进行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。并且从 Model2 开始我们就使用 mimicus 对文件进行特征提取，一共提取特征 135 个。我们的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%。

当检测率与误报率趋于稳定的时候，我们用十万级别的样本重新对模型进行训练。在 4 核 4G 的 CPU 上，训练时间仅需要 56s。使用 2 万数据集进行测试，Model2 准确率维持在 99.81%，误报率为 0.086%。在这里我们可以看到，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。

4. 对抗性学习

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。对部署的分类器知识可以通过各种源得到。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。

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。我们的实验结果揭示了即使面对简单的攻击，model2 的检测率有巨大下滑。与此同时我们研究了一些潜在的面对分类器逃逸攻击的防御策略。我们的实验表明有两种技术可以使模型面对此类攻击更为健壮。他们是：（1）增大模型训练的数据集（2）采用不同的特征集重新训练模型。在相关讨论的段落中，我们分析了一些潜在的技术以用于增强这些学习系统在面对对抗性操纵数据时的稳定性。

4.1 样本逃逸

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，其中 4 种运用不同攻击方法的场景如下所示：

- 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。

我们通过分类器找出评分较高的 2000 个病毒作为病毒母体，使用上述的四种方法生成可 PDFrate、且依然保持有恶意行为的病毒变种，然后使用这些病毒变种来攻击 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 对变种病毒的准确率只有 2.92%。就是说有 90%以上的病毒文件通过变异后逃逸分类器。

经过以上的攻击后，我们通过改变特征与样本集对模型重新训练生成 Model3，我们将 Model3 的训练数据升级到 20 万，还添加了一些新全新的病毒变种的样本，如通过模仿良性样本（Mimicry Attack）和反向模仿(Reverse Mimicry Attacks)生成的变异文件。之后再面对以上 4 种攻击方法的时候，Model3 的检测率比 Model2 的检测率有所提高。如表 4 所示：

表 4 不同攻击方法与准确率

攻击方法	病毒变种	Model2 准确率	Model3 准确率
F	2157	71.18%	96.71%
FC	240	2.92%	12.50%
FT	4196	84.25%	96.76%
FTC	600	15.83%	18.71%

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 模型更新

针对逃逸样本，我们采用两种抗攻击的方法来更新之前的模型，1.增加训练样本的个数当训练样本达到一定的数量，就会避免数据过拟合，实现局部最优（Local optimum）的情况；2.重新调整特征集也可以使检测率有所提高。

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，是模型训练后，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，然后重新训练模型，预测结果如表 7 所示。

图 2 前 30 个重要特征分布图

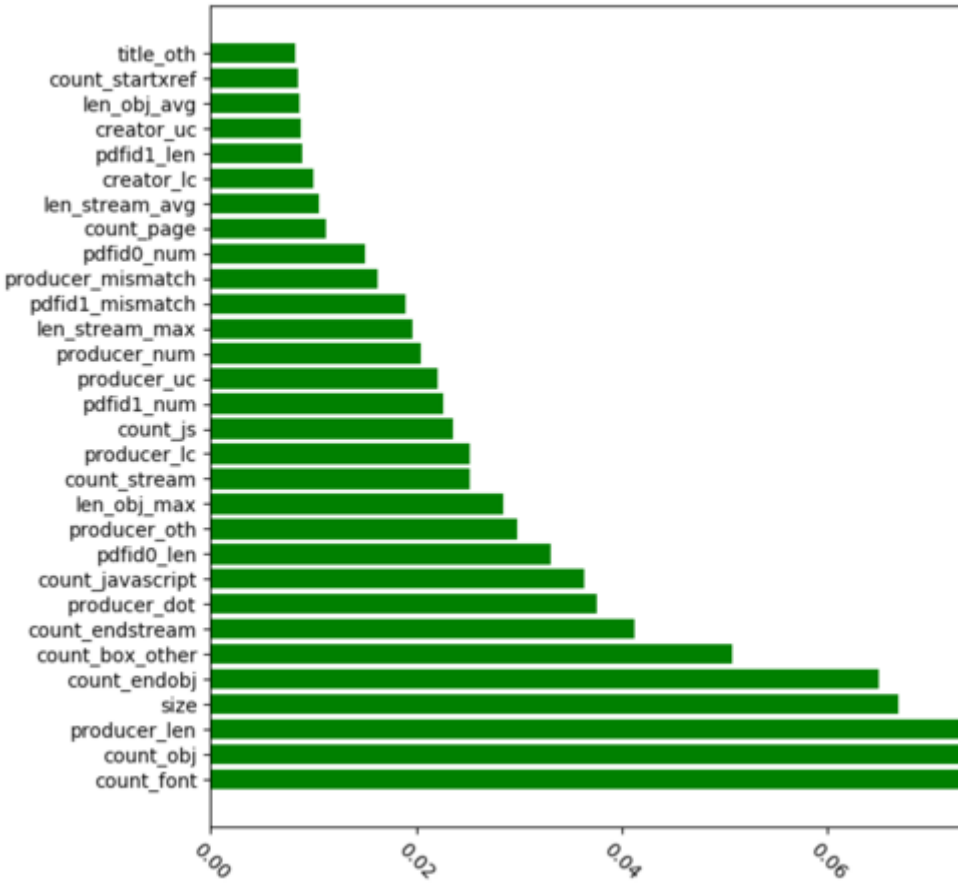


表 7 是对前 5 个特征修改后的模型准确率。由表中可知，当分类器使用全部特征进行训练时，模型准确率高达 99.82%。当我们将第一个重要的特征在训练的时候删去，检测率基本没有太大的波动，当减到 count_endobj 前 5 个特征时，模型准确率的波动可忽略不计。这也说明了我们的模型可以对抗一些基于特征的攻击，即使对手知道我们分类器使用的特征，模型同样可以达到 99%的精度 。

表 7 对前 5 个特征依次删除后的模型准确率

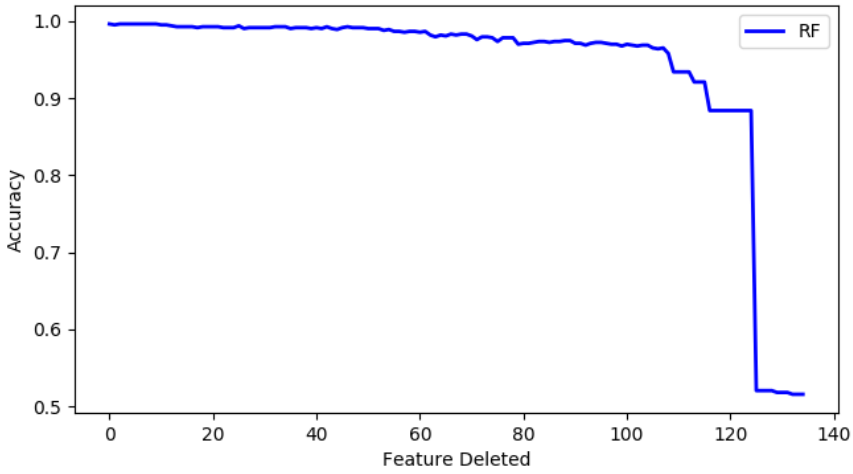
Feature delete train	准确率
None	99.82%
count_font	99.52%
count_javascript	99.52%
Size	99.64%
count_obj	99.64%
count_endobj	99.64%

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；

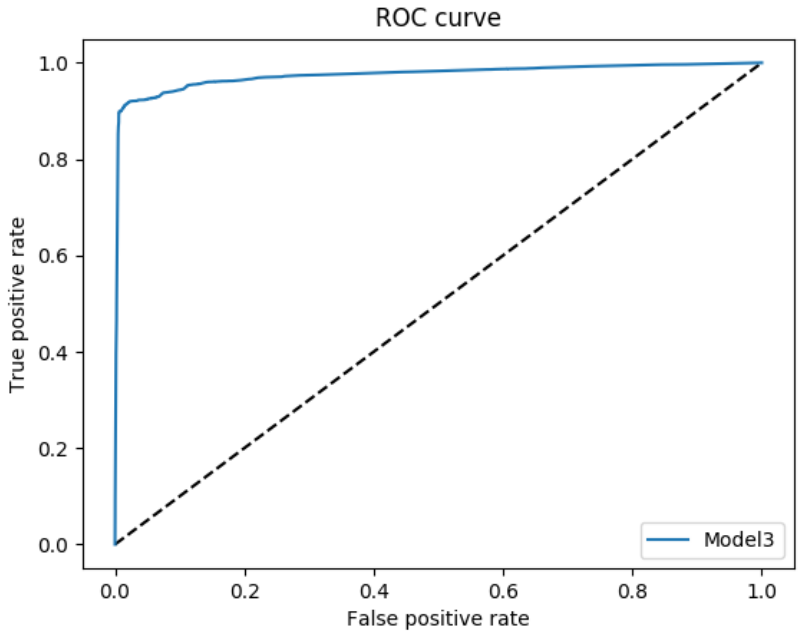
图 3 特征自减后的识别率（模型 3）



4.4 性能评估

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，这表明模型具有良好的预测性能。

图 4 ROC 曲线图



特征提取是最耗时的操作，因为它需要从硬盘加载所有的文件，并对文件进行逐个解析。于是我们将文件解析与训练分步处理，把解析后的特征集作为中间结果保存，不仅可以减少 CPU

内存占用，同时也可以使模型在更新训练时更为快速。对训练样本的解析（十万级别）共耗时约 22 分钟。表 8 是模型使用不同算法之间的训练与预测时间对比，由表可知，使用随机森林的模型在此任务中不止有良好的准确率，并且预测时间仅需要 1s。

表 8 训练时间与预测时间

	训练时间	预测时间	准确率
Random Forest	56s	1s	99%
Decision Tree	4s	1s	97%
SVM	58m 18s	12s	75%

5. 应用实例：蓝盾 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。

6. 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）

7. 致谢

感谢蓝盾为本研究工作提供平台条件，同时也十分感谢李德圆、黄国彬、魏舒敏及王木梯对本工作的支持与讨论

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.

[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