**基于机器学习的PDF文件检测与分类器逃逸**

作者A1，3，作者B2

（1.单位名1,城市名1 邮政编码1）

（2.单位名2,城市名2 邮政编码2）

**Abstract -** Nowadays, with the highly rapid development of information, it is becoming more and more important to detection based on malicious documents (such as PDF). A attack for document type is targeted and more vulnerable . Because of the diversity of its structure and form, attackers have various accesses and its technique is hard to be retrieved. Therefore, this research project aims to provide document (PDF) classifier based on AI for industry and academia. The study has collected 200,000 samples data and analyzed training process. This results show that the accuracy for training model highly rated 99.82% while the false positives only rated 0.01%; and through the study of antagonism, the model has certain ability to resist attacks and good robustness. This model can be widely used in the Terminal Security Products or mail servers when a file is detected at a millisecond level. It is a feasible model for AI is not limited to use in PDF document classifier, also can be extended the structure of documents such as word, excel etc.

**KeyWords：**AI Application Security; Machine learning; Maldoc Detection; Adversarial ML

**摘 要：**如今信息发展迅速，基于恶意文档（如pdf）的检测越来越重要，文档类型的攻击具有针对性，并且更容易攻击成功。因为其结构与形式的多样，攻击者手段之丰富且容易隐藏。此研究项目的目标旨在为工业界和学术界提供一个基于AI的文档（PDF）分类器，我们收集了20万个样本，对其进行解析训练，最后训练模型检测率99.82%，误报率0.01%，并通过对抗性学习，证明模型具有一定的抗攻击能力，与良好的鲁棒性。当个文件检测时间为毫秒级别。此模型可广泛应用于终端安全产品，或邮件服务器，均是非常有意义的应用场景，AI文档分类器不仅限于pdf，如可扩展到word，execl等有结构的文档。

**关键词：** 人工智能化应用安全；机器学习；恶意文件检测；对抗性学习

作者简介：作者1姓名（19XX－ ），性别，籍贯，职称，最后学位（或在读学历），主要研究方向，通信地址和邮编，手机，电话，Email;作者2姓名（19XX－ ），性别，籍贯，职称，最后学位（或在读学历），主要研究方向，通信地址和邮编，手机，电话，Email。项目基金：名称、编号

1. **简介**

随着时间的推移，PDF规格和样式都在改变，变得丰富。增加的脚本功能可以使文档与可执行文件几乎相同的方式工作，包括连接到Internet的能力，运行进程以及与其他文件/程序进行交互。内容复杂性的增长为攻击者提供了更多的武器来发动强大的攻击，并且更灵活地隐藏恶意有效载荷（例如，加密，隐藏为图像，字体或Flash内容）并逃避检测。由于企业和个人用户系统未采用最新的浏览器插件（例如，Adobe的 Flash Player 和 Acrobat Reader）修补程序进行更新，这些攻击得以取得成功。

在2012年，开创性的漏洞攻击工具Blackhole专门针对Java，PDF和Flash文件。2013年，通过网络传输的不可执行恶意软件主要是针对Adobe Reader和Microsoft Office应用程序的PDF和Flash文件。在2014年发现的24个0day中，有16个是针对Adobe Reader和Flash Player的）如图1 是近几年来Adobe Reader在CVE上公布的一些漏洞统计，由此可以看出，从2015年开始关于Adobe Reader发现的漏洞就在线性增长，就表明了正对与PDF文档的攻击开始增多。

针对于近几年来遇到的多种关于文档类型的攻击，传统的恶意PDF 检测方法有基于Shellcode[58]检测、基于签名的的检测方法[31]等，这些方法存在识别率不高、无法及时更新恶意代码等问题。机器学习技术为恶意PDF 检测提供了新方向，PDF文档的检测研究大多采用PDF 文档内容或结构为特征，利用随机森林、SVM、决策树等分类器构建PDF 检测器。在本文中经过多次实验对比，选取了识别率最好的随机森林构建模型，使其准确率达到了99.82%。

本文的主要贡献如下：

* 目前数据集从业界通用的万级别 到 达到 二十万 级别以上，其中恶意样本173036个，正常样本28332个。
* 选取了一组有用的和综合的静态特性（133个）以进行检测，使其可以有效的区分PDF 恶意与良性的样本
* 训练模型准确度高达99.81%，误报率0.16%
* 使用逃逸的方法生成7000个恶意文件成功逃逸工业界认可的PDFrate
* 成功实现了在几种情况下进行的回避攻击，并重新训练出一个新的模型（有一个3.1版本支撑）

1. **相关工作**

如今检测PDF文件的方法有很多，主要的方法可以分为两类，动态检测和静态检测，纵然这有一定程度上的重合。动态方法需要把文档放在某一个特定探针环境（instrumented environment）中打开运行；而静态方法则无需打开运行仅通过静态特征即可完成检测。表1 列出了现有方法的总结

表 1 PDF分析技术对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 方法 | 分析重点 | 检测技术 | 是否有解析器 | 是否是ML | 是否依赖 | 逃逸 / 缺陷 |
| 静态分析 | JavaScript | Lexical 分析 [27] | 是 | 是 | 是 | 代码混淆，加密 |
| JavaScript | Token 聚类 [59] | 是 | 是 | 是 |
| JavaScript | API 调用分类 [14] | 是 | 是 | 是 |
| JavaScript | Shellcode and opcode 签名 [31] | 否 | 否 | 是 |
| Metadata | Linearized object path [36] | 是 | 是 | 是 | Mimicry[53], Reverse mimicry[35] |
| Metadata | 分层结构检测 [33, 52] | 是 | 是 | 是 |
| Metadata | 基于内容和Metadata [46] | 是 | 是 | 是 |
| Content | 基于结构和内容解析2015[10] | 是 | 是 | 是 |
| 综合 | 结合上述几种技术解析分类 [34] | 是 | 是 | 是 |
| 动态分析 | JavaScript | Shellcode and opcode签名检测 [58] | 是 | 否 | 是 | 多元化的JS引擎分析 基于Non-script的攻击 |
| JavaScript | 已知的攻击模式 [45] | 是 | 否 | 是 |
| JavaScript | 内存访问模式 [48] | 是 | 否 | 是 |
| JavaScript | 常见 maldoc 行为分析 [29] | 否 | 否 | 是 | Zero-day 漏洞利用 ROP 和 JIT-Spraying |
| 文档类型 | 异常内存访问约束变量 [62] | 否 | 否 | 否 |
|  | 平台多样性 | 系统平台多样性的漏洞利用[30] | 否 | 否 | 是 |  |

### 2.1 动态分析

一些核心想法促成了动态分析方法的兴起。最开始的工作是基于模拟的方法，此方法把可疑包（suspicious payload）使用抽象包执行（abstract payload execution）[36]或使用软件模拟[1，26]。然而，软件模拟并不能有效覆盖全部指令集。正因如此，有的文件就会逃逸检测。为了解决上述问题和增强可扩展性，最近提出的系统SHELLOS在可控shellcode执行的过程中[34]中使用硬件虚拟化技术以取代模拟方法。在一些常用的操作系统核心上，SHELLOS能有效地检测出shellcode（此shellcode在被程序任意分配的缓存之中）。然而，这么好的效果有一定的代价。尽管SHELLOS在检测网络层攻击时显示出优秀的吞吐量，但是检测恶意文档的时却遭受严重的延迟（在秒级别）。这种延迟造成的原因是检测位于内存缓存（memory buffers）中，此类检测需要程序首先开辟内存缓存，才能进行检测。

另外一种动态分析的方法聚焦于检测在运行JavaScript时的恶意行为。JSAND使用10个仔细提取的启发性特征去训练良性的JavaScript模型，然后使用此模型去检测与此模型基线有较大偏移的攻击[9]。一个相似的方法也被成功运用于ActionScript3恶意软件的检测当中[24]。CUJO建立在专门针对JavaScript的沙箱上，且可自动学习事件序列模型（models of sequences of events），这些模型均对JavaScript的解析器产生影响[31]。专门的Java Script动态分析方法与shellcode检测方法相比，有很大的性能提升（单个文件的检测时间控制在数百毫秒内），同时也保持了高准确度和低误报率。

### 静态分析

PJScan是第一个，成功实现基于JavaScript内容的PDF恶意软件检测[19]。为了提高效率，PJScan的JS提取器只搜索那些PDF标准预先指定好的位置。不幸的是，这种提取方法会在以下情况下有局限，攻击者可以将JS代码放在其他位置，通过PDF JavaScript API访问，并使用eval()函数调用来获取，以致来逃逸检测。

另外Maiorca等人提出的系统MALWARE SLAYER[23]，基于模式识别。这里的模式是指使用PDFID工具从PDF文档中提取的文本关键字。它在真实PDF数据集上展现出高准确度和低误报率。这种方法被PDFID自身的函数所局限，且不能处理多版本（multiple revision numbers）和在对象流（object stream）中的隐藏对象（object hidden）。

PDFrate是一个相对成熟基于机器学习的静态PDF分类器。此分类器使用简单的PDF附带信息（meta info）和字节层面（byte-level）的文件格式。此PDFrate在大数据集中有出色的分类表现[33]。然而，此分类器不能提取对象流（object streams），在检测时这个特征就不能加入到训练样本中。

Maiorca等人。也研究了对PDFrate和其他PDF文档分类器的逃逸[26]，[27]。他们提出 反向模仿技术。为了使内容看似良性（如Mimicus所做的那样），他们将恶意内容嵌入到良性PDF中，而不是将内容添加到恶意文档中，而是尽可能少地修改。反向模仿攻击实施针对PDFrate的独立逃避方法, Maiorca等人提出了三种不同的逃避方案。在EXEembed方案中，恶意可执行文件被植入到现有的良性PDF文档中。打开文档时会执行恶意软件。这些文档利用CVE-2010-1240。在PDFembed方案中，恶意PDF被嵌入到良性PDF中。这些嵌入式文档在文档打开时自动呈现。为了评估，Maiorca等人 将利用CVE-2009-0927的文档嵌入到现有的良性PDF文档中。最后，在JSinject场景中，恶意JavaScript（与PDFembed嵌入式文档中使用的相同）直接注入根良性文档。

### 动静结合

另外两个须提及的贡献是把静态和动态分析技术结合起来。MDScan[37]对PDF文件进行静态分析，目标是提取所有的JS代码簇（chunk），这些代码簇可以作为执行的入口（entry point）。直到现今为止，一个特定用途的解析器（parser）被MDScan开发，这个解析器被用于抓取在文件里面的其他信息，这些信息包括被交叉引用图表（cross-reference table）所忽略的对象以及潜在的恶意对象。被抓取的代码会在JavaScript引擎中被执行（此引擎会模拟Acrobat Reader引擎）。在控制执行中（controlled execution），所有的内存缓存都会被一个工具所检测，此工具基于二进制模拟（binary emulation NEMU）以用于shellcode检测。

在ZOZZLE[10]一文中，静态与动态部件的角色发生了变化。ZOZZLE的动态部件会在运行前提取JavaScript的片段，这些JavaScript源于IE浏览器中的JavaScript引擎，此引擎能自然而然地解决JavaSript代码混淆（obfuscation）的问题。ZOZZLE的静态分析部分使用Bayesian分类器，此分类器建立在对检测到的JavaScript源代码进行混合分析（syntactic analysis）的基础上。

### 机器学习

对相关工作的比较表明，在当前用于文档恶意软件检测的静态和动态系统间有明显的权衡（trade-off）。纵然动态分析系统表现出高准确度和低误报率，这种优势是以高延迟、明显的性能瓶颈和需要特定的受控制系统（instrumented environments）为代价。在本文中提出的新方法，旨在从静态方法填补空隙。此技术通过从静态方法开始，增强检测性能同时保持系统设计的简约和计算的高效。为了达到这样的目标，我们使用了一个现成的解析器，构筑关于“Comprehensive Static Analysis of PDF”的方法论。更进一步地，我们对样本潜在的逃逸策略有特别的关注，且用实验去评估所提及的方法对特定攻击策略的健壮性。

Charles 等人提出通过随机森林检测含恶意代码的PDF 文件技术PDFrate，作者从PDF 文档元数据以及文档结构中提取了135 个特征，使用已标记特征的训练数据，并采用10 倍的交叉验证，生成具有多个分类树的分类器，从待测PDF 文档提取特征，评估森林中的每个树，最后投票决定其分类。该方法初始训练过程计算开销较大，但一旦分类器构建完成，对待测的PDF 文档的分类速度很高，我们同时也对这135个特征进行了提取，利用随机森林的算法对文档进行预测分类。

### 对抗性学习

机器学习在一定程度上可以检测到很多关于新奇恶意代码的攻击，但是同时也会有一些攻击是基于分类器的，许伟林2016发表的关于自动逃逸分类器的一文中[7],就针对两种PDF恶意软件分类器（**PDFrate和Hidost**）的技术进行实验为*所有的*分类器自动找到规避的变体，这样一来即使分类器的检测率为99%，也可以通过这个方法，多次变异后成功逃逸分类器。这对检测PDF的模型来说是一个值得关注的问题，也就是模型的鲁棒性。Lux0r [14]构造了两组分别在良性和恶意文档中找到的API参考模式，并将其用于对maldoc进行分类。Maiorca等人。 [34]关注JavaScript和元数据，并将许多上述启发式算法融合到一个程序中，以提高逃避弹性。相反，反向模仿[35]攻击尝试将恶意内容嵌入到良性PDF中，注意尽可能少地修改恶意内容。Mimicus [53]实现了模仿攻击并通过向文档添加空的结构和元数据项而对现有的maldoc进行更好的显示，使其看起来更像良性，而对渲染没有实际影响。

由于Adobe Reader的不断开发，恶意文件（maldoc）检测已经成为一个紧迫的问题。尽管已经提出了许多解决方案，但最近的作品突出了一些常见的缺陷，就如上面所提到的几种攻击方法，针对这一点Meng Xu等研究者就提出了一个检测PDF恶意文件的新视角：平台的多样性，通过操作系统设计和恶意样本与良性样本运行时的行为差异，从系统语义到堆对象元数据结构，进一步展示如何阻止攻击者发现错误，避免其逃逸。

1. **分类评估方法**

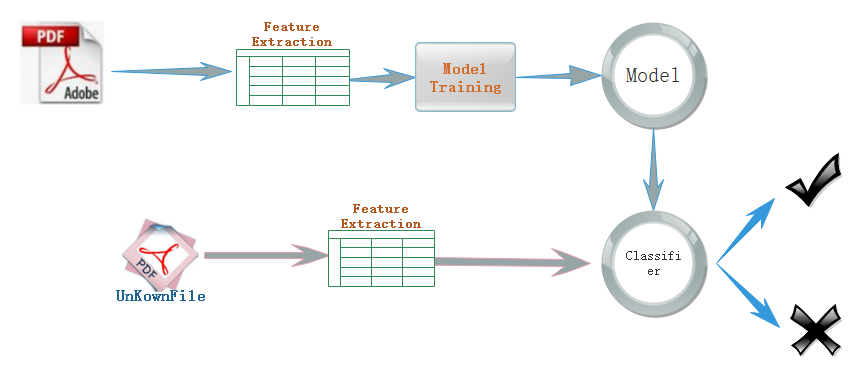
### 训练分类模型构建

所提出的用于基于结构的检测恶意PDF文档的方法包括以下两个步骤，如下图所示：

1. 提取文件特征。作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，并做一些相应的向量计算，提取为一个二维的特征集，使得这些特征可以进入到机器学习中进行训练分类。

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4 机器学习的基本框架



在处理大量PDF数据时，高吞吐量是一个重要的考虑因素，就像V IRUS T OTAL 那样。我们的系统旨在处理这种负载，并利用现代计算机系统的并行处理能力。我们测量了使用随机森林，决策树和SVM 对十万的数据集进行特征提取，学习和分类所花费的时间如 表7 。训练预测是在一个4核4G的CPU的一个Ubuntu 的系统下进行的。

特征提取是最耗时的操作，因为它需要从硬盘加载所有的PDF文件。并对文件进行逐个解析，我们测试的数据有10万，一共5.8G。对这些数据进行了两次解析，一共耗时22分钟36秒。然后对文件进行训练和预测，如图7 是不同算法之间的训练时间和预测时间，以及精度的统计。由表我们可以看出随机森林和决策树的预测时间都快于SVM，但是决策树的训练时间会比随机森林快很多，这个是由算法本身的因素决定的，并且随机森林的准确率是明显高于其他算法的。

表7 训练时间与预测时间

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training | Classification | accuracy |
| Random Forest | 56s | 1s | 99% |
| Decision tree | 4s | 1s | 97% |
| SVM | 58m 18s | 12s | 75% |

### 算法选择

AI引擎的核心就是算法，我们选取了几个实用性较好的几个算法作为计算，这几个算法包括KNN邻近算法，NNET(神经网络)，RF（随机森林）SVM（支持向量机）经过多次的训练与分类实验，最后随机森林以其准确率较高，误报率极低，和低延时，并且技术时间快的优势，被用于我们最后模型的核心算法。

我们通过对四种算法的准确率和鲁棒性进行比较，发现随机森林的准确率和鲁棒性要好很多

表8 算法准确率对比

|  |  |  |  |
| --- | --- | --- | --- |
| KNN | NNET | RF | SVM |
| 0.971292 | 0.824163 | 0.996411 | 0.752392 |

我们使用了KNN邻近算法，NNET(神经网络)，RF（随机森林）SVM（支持向量机）四个算法来做训练，如表6 可以看出RF预测精度会比其他三个算法高，并且随着特征减少依然可以有很好的一个准确率。为了评估这个几个算法

### 分类结果

在我们的实验当中一共对模型进行了4次更新，其中最开始的模型（model1）是通过使用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)的一些静态属性,并通过多次实验核对比较选取性能和精度最好的算法(随机森林RF)此时准确率就达到了99.30%，可是经过我们的研究发现，因为在一开始我们使用peepdf进行解析的时候，并不所有文件都解析到参与到，所以在第二阶段我们重新选取了解析器mimicus[5],这个工具可以弥补之前因为结构缺陷或混淆使文件不能被解析器正常解析的缺陷，同时这个工具也已经开源<https://github.com/srndic/mimicus>，到目前为止也是很多基于PDF文件检测使用最多的一个解析工具。我们使用了mimicus 对之前的文件进行解析，到目前为止可以对所有的数据（20万）进行解析，并做特征提取。

于是在模型2的训练中，我们先使用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，在总的数据集中进行随机抽选一共4万个样本用于model2 的训练与预测。并且从Model2 开始我们就使用mimicus 对文件进行特征提取，其中包括有基于结构的，基于内容的，基于javascript 的，基于Metadata 等等，一共提取特征135个。我们的机器学习的核心算法还是使用准确率最高的随机森林。通过算法调优后对数据进行训练与预测，使Model2的检测率提高到99.99%，误报率降低为0.012%，并集过多次随机选择数据进行训练预测后，准确率依然可以维持在99.99%。

经过Model2 的算法调优和数据筛选后，我们把训练数据增加到十万级别，来对模型进行训练，我们的训练是在本地进行的，在4核4G的CPU上，训练10万个样本仅需要56s,一分钟都不到，其中一个原因是我们在最开始做好了特征提取的工作，作为一个AI流水线的节点，毕竟文件解析与特征提取花费的时间会比训练检测的时间会多，将文件解析与训练分开出来，作为中间结果保存，不经可以减少CPU 内存占用，同时也可以使模型在更新训练时更为快速。当训练样本达到十万级别的时候，Model2.1准确率持续稳定在99.81%，误报率为0.086%。

此时我们对模型的鲁棒性做了一些测试与对抗性学习，具体的来说就是通过一些假设攻击者已知信息，比如说攻击者知道我们所提取的特征，那么他就可以修改恶意文件的特征，使其看起来更像良性，从而逃逸分类器。又或者假设攻击者知道模型提取的特征和算法，通过一些加密，混淆等方法逃逸分类器，按照一定的逻辑关系，当攻击者知道模型的信息越多，越容易逃逸。我们主要利用Nedim Smdic 在IEEE 的论文中Practical Evasion of a Learning- Based Classifier: A Case Study[53]中提到的几种方法，对模型进行对抗性学习，这些会在第六章会具体介绍。

在通过一系列的对抗学习攻击之后，我们对模型重新训练，使模型对那些经过特定技术伪装的恶意文件具有一定的免疫能力，并且我们将Model3的训练数据升级到20万，添加了一些新奇的样本，如通过模仿良性样本的恶意文件，和反向模仿生成的变异文件，一同加入到了Model3 的训练当中，并对整个机器学习的参数 重新调优，经过我们的多次训练，模型3 在数据达到20w 万的情况下，更新模型花费了3分42秒，预测时间需要2秒，最后使模型准确率达到99.82%，误报率小于0.01%。

如下表是更新后模型的一个改进与更新的一个图标对比图，由表可以清晰的看出四个模型的特征数量，训练数据大小，检测率，及误报率。

图9 各个模型参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| model | 特征数目/个 | 数据大小/万 | 检测率 | 误报率 |
| model 1 | 133 | 10w | 99.30% | 0.054% |
| model 2 | 135 | 4W | 99.99% | 0.012% |
| model 2.1 | 135 | 10w | 99.81% | 0.086% |
| model 3.0 | 135 | 21W | 99.82% | 0.018% |

### 模型评估：

为了评估我们的检测模型的预测性能，我们将整个数据集随机分为两部分：训练（90％）和测试（10％）。 我们对每个组合重复该过程10次以上来验证结果。 这个方法有助于评估模型的鲁棒性。

同时我们还通过重要特征自减来评估模型的鲁棒性，对特征有效性进行研究，将所有特征进行一个重要性排序，然后依次将最重要的特征删减，后进行重新训练预测，图9是特征在自减之后的识别率的图示，我们可以观察到当特征减少了100个之后，模型检测率依然可以达到90%，说明模型具有很好的鲁棒性和抗攻击能力。

图9 特征自减后的识别率



我们通过改变输出类别概率的阈值来生成ROC曲线。ROC曲线下面积约为1，表明有效的特征和鲁棒性模型。图10所示，我们的检测率超过99％，RF分类器的误判率低于0.01％。换句话说，功能的静态挖掘在区分良性和恶意文件方面非常有效。 我们可以使用静态分析有效识别未知文件的好坏，以发现可疑信息。

图10 ROC曲线图



1. **对抗性试验**

为了系统地探索针对分类器系统的逃避攻击，基于分类器系统三个组成部分（特征提取，训练，预测），利用如下的逃避场景分类。

表7 攻击过程中的模型变化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attack method | Test file | attaktMoel2.1 | FN | recall |
| F | 2157 | 71.18% | 27.11% | 71.18% |
| FC | 240 | 2.92% | 97.08% | 2.92% |
| FT | 4196 | 84.25% | 15.30% | 84.25% |
| FTC | 600 | 15.83% | 84.16% | 15.83% |

我们复现了Nedim Srndic 2014年在IEEE 会议的论文中[53]提到的4种攻击场景，。他们的名字描述了对手可用的信息，在F情况下：表示只有特征集可用于敌手；在FT情况下，除了已知的特征之外，攻击者还可以利用目标分类器训练数据集的知识。在场景FC中，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如果知道所有三个分类器组件的细节，那么对手就有最大的机会躲避目标分类器。

我们通过使用这四种方法找到关于PDFrate的回避变体后，对我们的模型进行预测，发现仍然有很大一部分，既可以逃逸PDFrate ,也可以逃逸我们的分类器，其中在FC情况比FTC情况的逃逸更高，这就说明了，我们想训练的模型与PDFrate的一个差异性，同时在特征提取方面比较接近，所以容易逃逸，而在后面算法训练这部分，就没有全部逃逸。

针对于这些样本逃逸，我们通过改变特征集，修改权值，或删除重要特征等操作，将样本重新训练测试，因为我们的模型的鲁棒性很好，如图9测试结果一样，所以在我们经过重新计算训练后，模型最后也可以达到99.82的预测精度。

1. **应用场景**

嵌入AI PDF分类器的智能防火墙

按照模块化的思想，我们把基于AI的PDF分类器看作一个独立的检测模组，并把此检测模块集成到安全产品防火墙中。一个面世30多年的老式安全产品，如何为其插上AI安全的翅膀，蓝盾在这方面走在业界的前列。

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PDF文件的检测准确度需在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，在硬件层，是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可容忍的。这些年，随着恶意软件的急速增加，先前基于规则匹配的引擎从 标配 开始力不从心：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和旧设备的规则库；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。

综上所述，我们把基于AI的PDF分类器串联集成到防火墙中，成功替代了原本基于规则的引擎。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，据实验数据，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。

在此真实防火墙应用场景中，我们根据AI模型不同的输出概率以及原因，分情况触发后续多样的动作。举例说，当输出概率大于0.9时，表明基于静态分析方法有较大把握预测此样本为正（即恶意），故会第一时间触发安全产品，如防火墙，对此所属链接进行封锁（block）操作；相反地，当概率小于0.1时，表明模型有较大把握预测此样本为负（即善意），安全产品默认允许此所属链接。

有趣的是当输出概率在灰色区间时如[0.1, 0.9]所触发的行为，我们会把样本上传到云中作动态分析，此分析包括沙箱，安全分析师和威胁情报等。静态分析的优势是速度快，但对于某些具有强逃逸能力（如混淆加密等）的样本会有一定的局限；动态分析的优势是准确度高，分析深入但速度慢，占用资源较高。

综上所述，我们可知，基于AI的静态分析和动态分析技术各有千秋，我们通过把两种技术合理运用，能较好地满足真实应用场景下的产品需求。

1. **总结**

在本论文中，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，并在此基础上，结合具体真实应用场景，成功搭建了一个基于动静态分析的恶意文档立体检测平台。实验数据表明，在十数万级文档数据集中，我们能达到大于99%的准确度和小于0.01%的误报率。更进一步的，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描述 安全AI（apply AI tech. to security），把 AI安全(securing the model & the associate) 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（如添加好的PDF对象），以混淆分类器，大量逃逸的目的；（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（feature set been exploited），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. 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
2. 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
3. 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
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. Nedim ˇ Srndic and Pavel Laskov. Mimicus: A Library for Adversarial Classifier Evasion. <https://github.com/srndic/mimicus>.
6. 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
7. 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/
8. VirusTotal. Free Online Virus, Malware and URL Scanner.https://www.virustotal.com/.
9. Stephan Chenette. Malicious Documents Archive for Signature Testing and Research - Contagio Malware Dump. http://contagiodump.blogspot.de/2010/08/ malicious-documents-archive-for.html.
10. D. Maiorca, D. Ariu, I. Corona, and G. Giacinto, “A Structural and Content-Based Approach for a Precise and Robust Detection of Malicious PDF Files”, in Proceedings of the 1st International Conference on Information Systems Security and Privacy. ScitePress Digital Library,2015, pp. 27-36.
11. Charles Smutz and Angelos Stavrou. Malicious PDF Detection Using Metadata and Structural Features. Technical report, 2012.
12. C. Cortes and V. Vapnik. Support vector networks. Machine Learning, 20:273–297, 1995.
13. <https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf>
14. 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
15. <https://github.com/srndic/mimicus>

[34] 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.

[35] DavideMaiorca,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.

[53] Nedim Srndic and PavelLaskov. 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.

[27] Nedim ˇ Srndic and Pavel Laskov. Mimicus: A Library for Adversarial Classifier Evasion. https://github.com/srndic/mimicus.

[29] Nedim ˇ Srndic and Pavel Laskov. Practical Evasion of a Learning-Based Classifier: A Case Study. In 35th IEEE Symposium on Security and Privacy (Oakland), 2014.

[30] Meng Xu and Taesoo Kim, Georgia Institute of Technology:PlatPal: Detecting Malicious Documents with Platform Diversity . 26th USENIX Security Symposium 2017