# 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。我们的方法基于在文档附加信息和结构中提取特征。使用真实数据库，我们展示了（1）如何使用文档属性去满足恶意软件检测要求；（2）这些特征对于未知样本的耐久性。由我们的分析可知，随机森林方法，一种从各单独分类树中随机选取特征的整体分类器，即使在未知样本下，依然取得最高分类准确度。我们的分类准确率在99%以上，同时误报率控制在0.01%内。

## 3.1 数据集

PDF数据集共分成良性（28332）和恶意（173036）两类，目前样本总数为201368个。我们有156035个是从VirusShare下载下来的，大小有约6.8G，另有9000个正常良性样本来自于Contagio，其他的良性数据集是在搜狗和百度上通过爬虫抓取下来的。还有mimiucus中的数据集用于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，包括有contagio的5,000个良性样本和5,000个恶意样本，还有google的5000个良性样本和VirusStotal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，会近可能少地使用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](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E8%A1%A8%E7%A4%BA)、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 |