恶意 PDF 检测中的特征工程研究与改进

黄 娜[1*]，何泾沙[2]，吴亚飚[1]

(1. 北京天融信科技有限公司 北京 海淀区 100085；
2. 北京工业大学信息学部 北京 朝阳区 100124)

【摘要】在基于机器学习的恶意 PDF 检测中，现有特征容易引起混淆或逃逸。为了提高特征的准确性和鲁棒性，在现有方法的基础上研究和改进特征提取方法，结合内容特征、结构特征以及逻辑树的间接结构特征，通过分析特征重要性进行特征选择，最后应用分类算法实现恶意 PDF 检测。结构特征包括多个高频次叶子节点数量；内容特征包括元数据特征、字节熵值、流字节比例等特征。收集实验数据集，提取特征并分析，最终选择出 58 维特征，使用 LightGBM 算法训练梯度提升决策树模型，测试准确率为 99.9%，优于其他方法。另外，模拟攻击部分样本的特征，生成对抗样本，检测准确率同样达到 99.2%。

关 键 词 内容特征； DOM 树； 梯度提升决策树； 恶意 PDF 检测； 结构特征
中图分类号 TP311 **文献标志码** A doi:10.12178/1001-0548.2021403

Research and Improvement of Feature Engineering for Malicious PDF Detection

HUANG Na[1*], HE Jingsha[2], and WU Yabiao[1]

(1. Beijing Topsec Technologies Inc. Haidian Beijing 100085;
2. Faculty of Information, Beijing University of Technology Chaoyang Beijing 100124)

Abstract In malicious portable document format (PDF) detection based on machine learning, the existing features are easy to be confused or escaped. In order to improve the accuracy and robustness of features, this paper studies and improves the feature extraction method based on the existing methods. Combining the content features, structure features and indirect structure features of document object model (DOM) trees, the feature is selected by analyzing the importance of features and finally the malicious PDF detection is realized by using classification algorithm. The structural features are the number of leaf nodes with high-frequency. Content features includes metadata features, byte entropy, stream byte ratio, etc. The improved feature extraction method can avoid the problems of confusion and escape, and improve the accuracy and robustness of features. In the experiments, we extracted and analyzed features from the collected dataset, 58-dim features with high-importance were selected. Then we used LightGBM algorithm to train gradient boosting decision tree. The testing accuracy of this model reaches 99.9%, which is superior to the other methods. In addition, the features of some adversarial samples are simulated, and the detection accuracy is about 99.2%.

Key words content feature; DOM tree; gradient boosting decision tree; malicious PDF detection; structural feature

基于文件格式漏洞的攻击行为是网络安全的主要威胁之一。文件格式往往具有跨平台的特点，一旦漏洞被利用，各类目标主机都可被轻易攻破。文档类的文件格式，如 doc、docx、xls、pdf，在日常工作与生活中传播广泛，是藏匿和传播恶意行为的重要媒介，由此引起的安全事件不胜枚举。据 Cisco 发布的《2018 年度网络安全报告》统计，在 2017 年间，恶意邮件附件中最普遍的 3 种文件格式为 Office 文档 (38%)、压缩文件 (37%) 以及 PDF 文件 (14%)。

收稿日期：2021－12－28；修回日期：2022－01－10
基金项目：北京市博士后科研工作经费支助项目 (A 创新研发类 (2021-ZZ-087))
作者简介：黄娜 (1990－)，女，博士，主要从事机器学习、信息与网络安全等方面的研究.
*通信作者：黄娜，E-mail：huang_na@topsec.com.cn

PDF 文件格式是由 Adobe 公司于 1993 年制定的一种电子文档分发开放式标准，具有以下优点：1) 灵活的层次结构，可以封装文字、图像、字体格式、超链接、声音、影像等众多信息；2) 跨平台的特性，在各类操作系统中通用。正是由于这些突出的特点，使得 PDF 文件在为我们带来便利的同时，也为黑客提供了可乘之机。从攻击角度来看，恶意 PDF 文件可分为两种类型：1) 利用 PDF 文档规范本身存在的漏洞，如字典中相同 key 值对应不同 value、对象号错误引起误识别，以及利用 ASCII 编码隐藏关键节点等；2) PDF 文件中携带恶意内容分为 4 种具体情况，即嵌入恶意 JavaScript 代码、嵌入恶意文档、嵌入恶意远程链接以及嵌入恶意软件。

PDF 文件数量十分庞大，且具有统一的文件格式规范，便于提取出结构化特征，因此机器学习技术在恶意 PDF 检测中有良好的应用条件及效果。本文首先回顾恶意 PDF 检测研究现状，对其中存在的混淆和逃逸问题进行阐述；然后针对混淆和逃逸设计完善的特征组合，包括内容特征、结构特征以及逻辑树的间接结构特征，提高检测模型的性能。

1　研究背景

1.1　PDF 规范介绍

从物理意义上看，PDF 文件由文件头、对象集合、交叉引用表以及文件尾组成。文件头中储存该 PDF 遵循的规范版本。对象集合包括文档包含的所有对象，每个对象都以 obj 作为开头标志，endobj 作为结尾标志，中间为对象所包含的字段、子对象、流内容等。交叉引用表是 PDF 文件内部的重要组织方式，用户可以直接访问某对象，以 xref 作为开头标志。文件尾以 trailer 为开头标志，包含一些键值对形式的文档描述信息，如所有对象的数量、文档的作者、创建时间、ID 等。

从逻辑意义上看，PDF 文件为树形结构，如图 1 所示，Catalog 为字典类型的根节点，包含 Outlines、Pages 等子对象节点。其中，Pages 本身为字典型数据结构，是所有页面的集合入口，包括 Count、Kids、Parent、Type 等字段，Kids 包含描述页面信息的 Page 对象或 PagesTree 对象。除 Pages 之外，Catalog 字典中还有 Type、Version、PageLabels、PageLayout、AA 等对象节点，表 1 描述了部分关键对象及其意义。

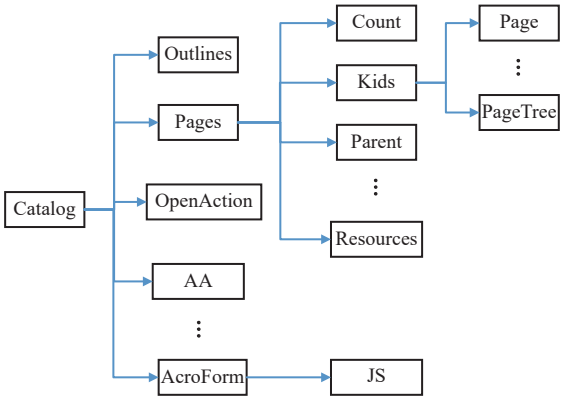


图 1　逻辑结构示例

表 1　Catalog 字典中常见的对象

字段	类型	描述
Type	name	必须为Catalog
Version	name	PDF文件所遵循的版本号
Pages	dictionary	页面集合入口
PagesLabels	number tree	定义Page和Page之间的关系
PageLayout	name	文档的页面布局
PageMode	name	文档的显示方式
Outlines	dictionary	文档目录
Threads	array	文档线索
JS	stream	执行JavaScript代码
JavaScript	stream	执行JavaScript代码
OpenAction	dictionary	自动执行相应动作
AA	dictionary	自动执行的相应动作
Names	dictionary	文档名称
EmbeddedFile	dictionary	打开嵌入的文件
F	dictionary	打开嵌入的文件
URI	dictionary	URL链接
AcroForm	dictionary	交互式表单
XFA	dictionary	交互式表单
Metadata	stream	文档元数据

1.2　相关研究

文献 [1] 结合使用动态特征与静态特征进行恶意 PDF 检测，提取动态 API 调用特征，在选择静态特征时，对文档结构中的关键字出现频次使用 KMeans 聚类，选出正常样本的关键字集合以及恶意样本的关键字集合[1]。在基于机器学习的检测方法中，所用的静态特征可分为逻辑结构特征和物理内容特征两类。物理内容特征包括元数据特征和具体内容特征等；逻辑结构特征包括节点路径特征、节点数量特征等。各类方法的归纳分类如图 2 所示。

元数据包括版本、作者、创建时间、编码方式等直接信息，以及文件大小、页面数目等间接信息。文件具体内容特征是指根据 PDF 文件中具体

内容提取的特征，如流内容的序列特征、熵、JavaScript 代码的特征等。PDF 中的 JavaScript 代码容易成为恶意行为的载体，是一种比较广泛的攻击方式。文献 [2] 通过分析 PDF 文档中的 JavaScript 对象来检测恶意 PDF。这种方法虽具有可靠的准确性，但对于非嵌入 JavaScript 类型的恶意 PDF 检测无效。

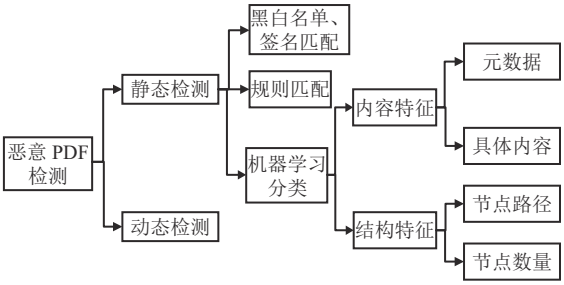


图 2 现有方法分类

目前常用的逻辑结构特征主要包括节点的路径特征和数量。文献 [3-4] 提出了 Bag-of-Path 方法，即使用节点路径作为特征，路径叶子节点的值经过数值化等处理作为相应的特征值。该方法直接利用对象路径反映 PDF 文档包含的对象特性及内容，不容易受混淆影响，但提取过程较为复杂，使得检测效率较低，难以适用于实时检测环境。文献 [2] 提出的 PDFMS(PDF malware slayer)，将 PDF 中存在的对象节点的数量作为特征，提取过程简便，但没有将正常 PDF 文档中一些常见的对象数量作为判别特征。

物理内容特征和逻辑结构特征能从不同的方面反映恶意 PDF 和正常 PDF 的区别，文献 [5-7] 将两者结合，常用的机器学习算法，如支持向量机 (support vector machine, SVM)、决策树 (decision tree, DT) 以及随机森林 (random forest, RF) 等，都有相应的研究及应用。文献 [8] 从 VirusTotal 收集了 2008 年–2019 年间的正常及恶意 PDF 样本，直接将原始的 PDF 文件输入卷积神经网络 (convolutional neural network, CNN) 进行分类检测，取得了 94% 的检测准确率。

随着机器学习安全问题逐渐引起重视，恶意 PDF 检测领域也出现了关于防御对抗的研究[9]，旨在增强检测模型的鲁棒性。文献 [10] 为了防御恶意样本逃逸 SVM 模型的检测，分别提取正常和恶意 PDF 样本集合中的高频节点作为特征，通过增加正常节点对恶意 PDF 进行伪装，将生成的逃逸样本加入 SVM 分类器的训练，经过 3 次迭代后，分类器能够完全检测出这类逃逸样本。文献 [11] 提出了防御特征加法攻击的集成决策树方法，但没有给出特征加法攻击中的具体特征。

2 方法研究及改进

2.1 恶意 PDF 中的混淆和逃逸

除了增加正常节点的逃逸方式外，在对大量恶意 PDF 进行分析之后，发现还存在利用 PDF 规范漏洞的混淆和逃逸手段。利用这几种方式，恶意 PDF 文档中的关键对象能逃逸现有检测方法，但依然能够在目前计算机中正常执行。

1) 对象号重写。对于对象号重复的情况，大部分解析器只读取对象号相同的最后一个对象。

2) 对象隐藏。对象隐藏在文件尾 (trailer) 中，能逃避目前大部分的解析方法，但隐藏在 trailer 中的对象依然能正常执行。另外，在 PDF 文档 body 内，ObjStm可以将对象压缩或加密为流 (stream)，若不对 ObjStm进行解析，则不能读取压缩对象的内容。

3) 关键节点混淆。现有的一种主流静态检测方法，是以关键节点数量作为特征，但关键节点可以通过 Ascii 编码混淆，使检测器提取出错误的关键节点数量。另外，PDF 嵌入文件有两种方式：“/Type/EmbeddedFile”与“/EmbeddedFiles”，但现有的相关方法通常只提取后者的数量作为关键节点数量特征。

4) 对象无结束标志。正常来说，PDF 中每一个对象都是以“obj”作为开始标志，以“endobj”作为结束标志。但有些恶意 PDF 在关键对象中删除结束标志，使得静态检测方法不能正常读取该对象，也就无法提取该对象的特征。

2.2 改进的静态解析方法

本文提出了一种改进的 PDF 静态解析方法，能够更加准确地提取出静态特征，包括预处理和具体特征提取，对抗逃逸手段。先对 PDF 文档预处理：1) 读取整个 PDF 文档的字节流，搜索 ObjStm 对象，其中/filter 节点中储存了所采用的编码算法，如“FlateDecode”，根据相应的算法将对象解码并替换原有内容；2) 搜索标识对象类型的节点，检查是否有 Ascii 编码，若有，对照 Ascii 编码将其还原；3) 匹配每一个对象的开始和结束标志，若没有结束标志，则将“endobj”添加到该对象的结尾处。然后，分别提取 PDF 文档的内容特征、结构特征以及逻辑树间接结构特征。

2.2.1　内容特征

本文设计及采用的内容特征具体见表 2。PDF 规范版本号、文件尾标志 (“EOF”) 数量、尾部所包含字节数是否经过修改，根据这些特征能够初步判断一个 PDF 文档是否符合规范、是否为伪造 PDF。其中，尾部所包含字节数能体现出是否有对象隐藏在尾部。

表 2　内容特征

特征	意义
Version	PDF文件格式版本
EOF	结尾标志的数量
EndBytes	尾部字节数
Modification	是否经过修改
TotalEntropy	总体字节熵
TotalBytes	总体字节数
Ratio	关键节点数量与节点总数的比例
StreamEntropy	流内容字节熵
StreamBytes	流内容字节数
nonStreamEntropy	非流内容字节熵
nonStreamBytes	非流内容字节数
Stream_in_nonStream	流与非流字节比例
objCount_in_size	对象数与文件大小的比例
StreamEntropy	流内容字节熵
StreamBytes	流内容字节数

字节熵 E 的计算方法为：

$$E=\sum_{i=0}^{255}(-\lg\frac{\mathrm{num}(b_i)}{T}) \tag{1}$$

式中，b_i表示一个字节，$\mathrm{num}(b_i)$表示字节b_i的数量；T表示计算内容的字节总数。

采用字节熵值、字节比例以及对象数与文件大小的比例作为特征，能够分辨攻击者向恶意 PDF 中添加的字节流、对象节点及其他无意义内容，防止对检测器产生误导。与正常 PDF 相比，恶意 PDF 的字节熵值、流与非流字节比例、对象数与文件大小比例往往较低，因此将其作为判别正常 PDF 和恶意 PDF 的特征。经过实验证明，关键节点 (如 OpenAction、URI 等) 数量在节点总数中所占的比例，也是一个重要的判别特征。

2.2.2　结构特征

恶意 PDF 的主要攻击方式是嵌入 JavaScript 恶意代码、嵌入恶意文件、嵌入恶意链接或交互式表单，因此，这 4 类相关对象的数量能够反映文档是否具有潜在的恶意行为。现有相关方法通常以 JavaScript、OpenAction、URI 等能够反映出文档具有潜在恶意属性的对象为检测重点。正常的 PDF 文档通常储存较多的文字、图像等，因此包含较多的 Encoding、Font、Resources 以及 MediaBox 等正常属性对象。

Bag-of-Path 方法是将逻辑结构中所存在的叶子节点的路径作为特征，能够表征更加具体的节点信息以及节点嵌套关系，但是该方法的实现效率较低，且提取的特征维度高。本文基于该方法，将具有相同叶子节点的路径合并，提取高频次节点的数量作为结构特征，实现更加简便，且能够降低特征维度。

为了找出对恶意检测意义较大的关键节点，利用 Bag-of-Path 方法提取了数据集中所有样本的路径，将其中的高频节点作为候选特征，节点数量作为特征值。根据本文提出的静态解析方法提取结构特征，遇到具有重复对象号的对象、隐藏在 trailer 中的对象，它们所包含的节点数量也会被提取，避免逃逸；将“EmbeddedFile”与“EmbeddedFiles”两类节点的数量合并，作为“EmbeddedFile”特征值。

2.2.3　逻辑树间接结构特征

完成以上预处理后，再解析 PDF 文件的逻辑结构，如图 3 示例，通过解析一个 PDF 文件的逻辑结构，生成逻辑结构树。从尾部入手，采用递归的方法遍历，生成 XML 格式的 DOM 树，算法详细描述如下。

1) 查找文件尾部的 trailer 对象；

2) 建立一个只包含根节点<trailer> </trailer>的 DOM 树；

3) 令 Parent=trailer，

if Parent 是 PDF 中的对象类型：

　　查找 Parent 包含的所有的子节点，依次插入 DOM 树；

　　依次取 Parent 的子节点，作为 Child：

　　令 Parent=Child，返回执行步骤 3)；

else if Parent 是 PDF 中的节点类型：

　　取出 Parent 节点的值，作为 value：

if value 是对象号 (“数字 数字 R”)：

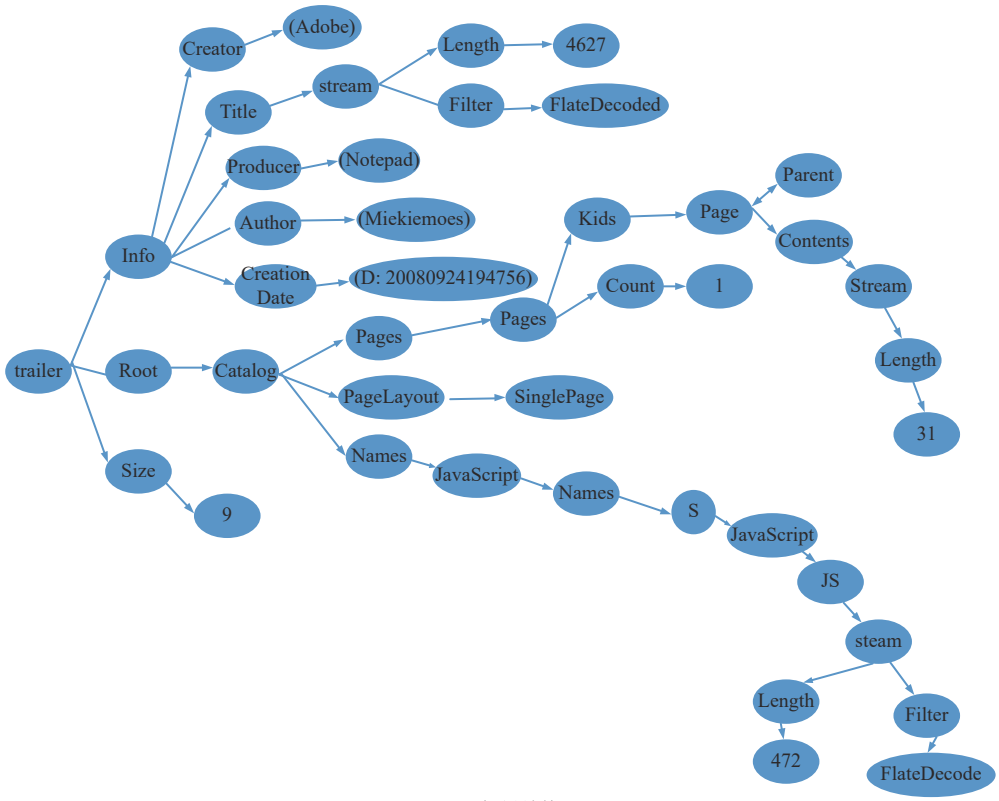
　　查找 objID=value 的对象，作为 obj；

　　令 Parent=obj，返回执行步骤 3)；

else if value 不是对象号：

　　将 value 作为 Parent 的子节点插入 DOM 树。

a. 物理内容



b. 逻辑树结构

图 3 PDF 逻辑结构示例

从生成的 DOM 树提取间接的结构特征，具体包括树的深度和广度、特殊子树的深度和广度、子树平均广度、所有节点的类型数以及各类节点的熵。深度是从根节点到所有子节点的路径中，最长路径的节点数；广度是在树的所有层中，节点最多的一层所包含的节点数。经过对大量样本的分析，发现较多恶意 PDF 会将关键节点藏匿在一个较深的分支中，如图 3 所示，因此 DOM 树的深度和广度是区分 PDF 文档是否为恶意的一个有效特征。特殊子树是根据正常 PDF 和恶意 PDF 所包含的节点对象的区别，或者节点对象梳理的区别，确定一个特殊节点，取出 DOM 树中以该特殊节点为根节

点的子树。如 JavaScript 节点会藏匿可执行的恶意代码，是一个特殊节点，以 JavaScript为根节点的子树即为一个特殊子树，提取它的深度和广度作为特征。子树平均广度是针对 DOM 树中除了叶子节点以外的所有节点，计算这些节点的平均叶子数量作为特征，假设除叶子节点以外共 n 个节点，每个节点所包含的叶子数量为$m_i, i \in [1,n]$，平均叶子数量为$\sum_{i=}^{n} m_i/n$。所有节点的类型数是指不重复统计 DOM 树中的节点，得到的数量作为特征。各类节点的熵计算式为：

$$E = -\sum_{i=1}^{c} P(O_i \lg(P(O_i))) \tag{2}$$

式中，假设 DOM树所有节点的类别数为 c；O_i表示每一类节点，$i \in [1,c]$，O_i出现的概率为$P(O_i)$。

2.3　特征选择

使用决策树算法中的信息增益率评价某个特征在分类任务中的重要性。将信息增益率表示为：

$$G_r(D|A) = \frac{H(D) - H(D|A)}{\mathrm{In}I(D,A)} \tag{3}$$

式中，D 表示数据集；A 表示某个特征；$H(D)$为数据集中样本类别的信息熵；$H(D|A)$为加入特征 A 作为分类依据后类别的信息熵；$\mathrm{In}I(D,A)$为特征 A 内部的信息熵，其计算分别为：

$$H(D) = -\sum_{i=1}^{n} p_i \lg(p_i) \qquad i \in [1,n] \tag{4}$$

$$H(D|A) = -\sum_{i=1}^{n} p_i [p_{(i|A=a_j)} \lg(p_{(i|A=a_j)})] \qquad i \in [1,n] \tag{5}$$

$$\mathrm{In}I(D,A) = -\sum_{i=1}^{n} p_{(i|A=a_j)} \lg(p_{(i|A=a_j)}) \qquad i \in [1,n], j \in [1,m] \tag{6}$$

式中，n 为数据集中的样本类别数；m 为特征 A 的取值个数；p_i表示随机取出一个样本属于类别i的概率；$p_{(i|A=a_j)}$则表示在特征 A 取值为a_j的条件下，随机取出一个样本属于类别i的概率。

2.4　LightGBM 分类器

集成学习是目前常用的一种防御对抗方法，通过融合多个弱学习器来降低攻击风险。文献 [11] 提出使用 Adaboost 算法训练集成决策树，提高恶意 PDF 检测鲁棒性的方法。LightGBM 也是一种采用 Boosting 方式的集成学习算法，以梯度提升决策树 (gradient boosting decision tree, GBDT) 为核心，通过改进的生长策略、分割算法以及并行策略，提高 GBDT 的训练效率，降低内存占用率。与深度学习以及其他集成学习算法相比，LightGBM 在并行拓展与运行效率上具有明显的优势，成为工业界主要应用的机器学习算法之一[12]。本文使用 LightGBM 框架训练 GBDT 模型，根据前面所提取的特征区分正常和恶意的 PDF 文档。

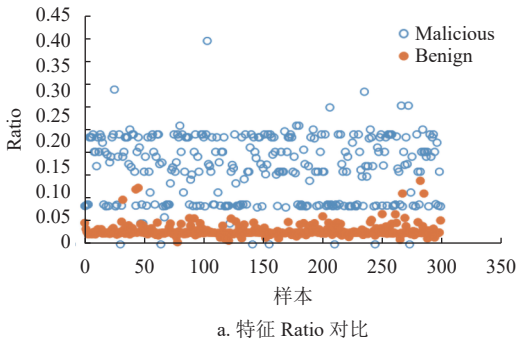
3　实验与分析

3.1　方法验证

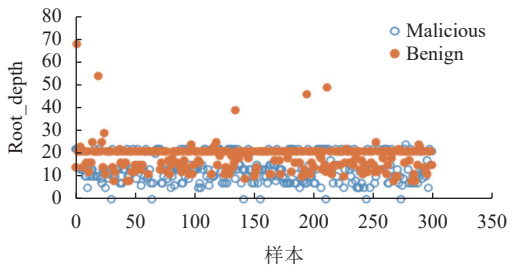
收集了网络环境中近两年的正常和恶意 PDF 样本各 6 000 个，作为实验数据集，如表 3 所示。在数据集上首先使用 Bag-of-Path 方法提取至少出现于 1 000 个样本中的路径，找出这些路径的叶子节点对象；然后提取对象数量作为结构特征。提取本文设计的其他特征，形成 71 维特征。图 4 给出了正负样本其中 4 种特征的分布，可看出明显差异：特征 Ratio 表现为黑样本的特征值较高，白样本的特征值较低；特征 Root_depth 表现为白样本特征值较高，黑样本的特征值较低；特征 ObjCount_by_size 表现为黑样本的特征值高于白样本特征值；特征 Stream_by_nonstream 表现为白样本特征值高于黑样本特征值。

表 3　实验数据集

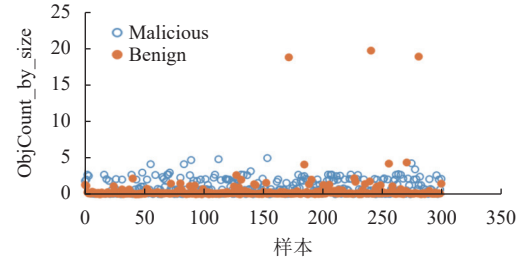
性质	训练集	测试集
恶意	5 000	1 000
正常	5 000	1 000



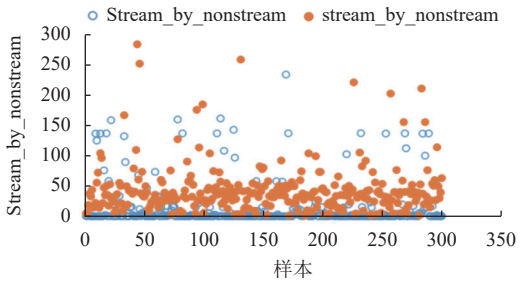
a. 特征 Ratio 对比



b. 特征 Root_depth 对比



c. 特征 ObjCount_by_size 对比



d. Stream_by_nonstream 对比

图 4 黑白样本的特征对比

图 5 展示出所提取的 71 维具体特征及它们各自的重要性。可以看出，本文提出的内容特征以及 DOM 树结构的间接特征，如关键节点数量、节点总数比例、根节点深度、字节熵、流内容字节数、字节比例等的重要性较高；在结构特征中，Resources、ToUnicode、BaseEncoding 等正常属性的对象数量特征，与 OpenAction、AA、EmbeddedFile、JavaScript 等具有恶意属性的对象数量特征都具有较高重要性。结尾标志“%EOF”、JBIG2Decode、colors 等 13 维特征不具有重要性，因此，在后续的模型训练中删除这 13 维特征。

图 5 特征重要性

3.2 与其他特征的性能对比

为了对比改进的特征工程方法，按照文献 [4] 和文献 [13] 使用的特征工程分别提取实验数据集的特征，同样使用 LightGBM 算法训练模型。测试性能的对比如表 4 所示，可以看出本方法比 PDFMS 具有更高的 AUC 值，与 Bag-of-Path 处于同一水平。对比表 5，本方法仍然显示出更高的准确性，而且特征维度较低，有利于简化模型，在效率上得到提高。

表 4 不同方法的 AUC 值对比

方法	AUC
PDFMS[4]	0.97
Bag-of-Path[13]	1.00
本方法	1.00

表 5 不同特征的性能对比

方法	特征维度	Precision	Recall	Accuracy	F1-measure
PDFMS	31	0.909	0.992	0.909	0.949
Falah 2020	13	0.943	0.991	0.965	0.966
Srndic 2016	309	0.986	0.995	0.982	0.991
内容特征+结构特征	346	0.986	0.998	0.993	0.992
特征选择(内容特征+结构特征)	37	0.998	0.998	0.998	0.998
本文方法	58	0.999	0.998	0.999	0.999

3.3 鲁棒性验证

黑客在制造对抗机器学习检测的恶意 PDF 时，既要保留 PDF 中所嵌入的恶意对象，又要误导检测模型，因此大多数的做法是向恶意 PDF 中插入一些正常的节点对象、内容流等，试图使模型将其误认为正常 PDF。如恶意 PDF 中 obj 数量少于正常 PDF，于是黑客通过增加 obj 对象的方式生成对抗样本。在现有恶意样本的基础上，统计恶意 PDF 和正常 PDF 中各个对象的数量差异，通过增加正常对象的数量，生成对抗样本测试集，用于测试、评估模型的鲁棒性。表 6 对比了模型检测无对抗样本和有对抗样本的性能，精确率降低了 0.1%，召回率提高了 0.1%，整体准确率无差别，可以证明该模型防御对抗手段的鲁棒性较为可靠。

表 6 对抗鲁棒性测试结果

测试集	Accuracy	Precision	Recall	F1-measure
无对抗样本测试集	0.992	0.999	0.984	0.991
对抗样本测试集	0.992	0.998	0.985	0.992

4 结 束 语

本文改进并实现了 PDF 静态特征提取方法，能够提取出更加准确的静态特征，防止混淆和逃

逸。实验验证表明，与现有的其他特征工程相比，本文结合使用的结构特征、内容特征以及逻辑树间接结构特征，能够使机器学习检测模型实现较高的准确性和鲁棒性。

参 考 文 献

[1] 杜学绘, 林杨东, 孙奕. 基于混合特征的恶意 PDF 文档检测[J]. 通信学报, 2019, 40(2): 1-11.
DU X H, LIN Y D, SUN Y. Malicious PDF document detection based on mixed feature[J]. Journal on Communications, 2019, 40(2): 1-11.

[2] LASKOV P, SRNDI N. Static detection of malicious JavaScript-bearing PDF documents[C]//The 27th Annual Computer Security Applications Conference, ACSAC 2011. Orlando, FL: [s.n.], 2011: 5-9.

[3] SRNDI N, LASKOV P. Detection of malicious PDF files based on hierarchical document structure[C]//2013 Network and Distributed System Security Symposium. San Diego, California: ISOC, 2013: 1-16.

[4] SRNDIC N, LASKOV P. Hidost: A static machine-learning-based detector of malicious files[J]. EURASIP Journal on Information Security, 2016, 2016(1): 22.

[5] SMUTZ C, STAVROU A. Malicious PDF detection using metadata and structural features[C]//Computer Security Applications Conference. [S.l.]: ACM, 2012: 239.

[6] MAIORCA D, ARIU D, CORONA I, et al. A structural and content-based approach for a precise and robust detection of malicious pdf files[C]//Proceedings of the 1st International Conference on Information Systems Security and Privacy. Loire Valley: [s.n.], 2015: 27-36.

[7] TORRES J, SANTOS S. Malicious PDF documents detection using machine learning techniques-a practical approach with cloud computing applications[C]// International Conference on Information Systems Security and Privacy. Funchal, Madeira-Portugal: SciTePress , 2018: 337-344.

[8] FETTAYA R, MANSOURY. Detecting malicious PDF using CNN[EB/OL]. [2021-10-21]. https://doi.org/10.48550/arXiv.2007.12729.

[9] KANG A R, JEONG Y S, KIM S L, et al. Malicious PDF detection model against adversarial attack built from benign PDF containing JavaScript[J]. Applied sciences, 2019, 9(22): 4764.

[10] CUAN B, DAMIEN A, DELAPLACE C, et al. Malware detection in PDF Files using machine learning[C]//The 15th Intrnational Joint Conference on e-Business and Telecommunications. Piscataway, NJ: IEEE, 2018: 412-419.

[11] 李坤明, 顾益军, 张培晶. 对抗环境下基于集成决策树的恶意 PDF 文件检测[J]. 计算机应用与软件, 2020, 10(37): 318-322.
LI K M, GU Y J, ZHANG P J. Detection of malicious PDF files based on integrated decision tree in adversarial environment[J]. Computer Applications and Software, 2020, 10(37): 318-322.

[12] 邢红梅, 陈欣, 王慧. 基于 LightGBM 模型的文本分类研究[J]. 内蒙古工业大学学报, 2020, 39(1): 52-59.
XING H M, CHEN X, WANG H. Research on Text classification based on lightGBM model[J]. Journal of Inner Mongolia University of Technology, 2020, 39(1): 52-59.

[13] AHMED F, LEI P, SHAMSUL H, et al. Improving malicious PDF classifier with feature engineering: A data-driven approach[J]. Future Generation Computer Systems, 2020, 115: 314-326.

编　辑　税　红