Image-based malware classification using section distribution information

2024-09-28 18:40:52

# 摘要

传统的基于机器学习的恶意软件分类方法面临着如何高效、准确地检测大量恶意程序的严峻挑战。为了应对这一挑战，基于恶意软件图像和深度学习的恶意软件分类成为一种有效的解决方案。然而，从由PE文件二进制序列转换而来的灰度图像中很难识别出区段的数量、顺序和大小等区段分布信息。因此，该文提出了一种新颖的可视化方法，引入彩色标签框（ CoLab ）来标记PE文件的区段，从而强调恶意软件图像中的区段分布信息。此外，还构建了一种称为 MalCVS （使用 CoLab 图像， VGG16 和支持向量机进行恶意软件分类）的恶意软件分类方法。

# 引言

传统的恶意软件分析方法都存在一定的不足，基于特征的恶意软件分析方法随着恶意软件的迅速增长而变得低效，基于静态分析的恶意软件分析方法会因为混淆而变得过度复杂，基于动态分析的恶意软件分析方法大多都很费时，而且一旦恶意软件检测到自己在虚拟环境中运行并改变行为，动态检测方法会直接失效。

随后基于机器学习的的恶意软件检测和分类方法被提出，然而此类方法非常依赖提取的特征，但特征提取和特征工程非常耗时。最近几年，基于恶意软件图像的深度学习方法受到很大的关注，因为该方法能够自动分析数据之间的关系从而消除大量的特征提取工作。

由二进制文件的字节序列转化的灰度图是最广泛使用的恶意软件图像方法。该文将PE文件的节数量、顺序和大小称为区段分布信息。考虑到属于同一恶意软件家族的PE文件的相似性，区段分布信息可能对恶意软件分类有用。不过从恶意软件图像中识别区段分布信息很难。因此，该文提出了一种新颖的可视化方法，该方法能够强调恶意软件图像中的区段分布信息。

总而言之，该文主要贡献如下：

* 提出了一种恶意软件可视化方法，进一步强调转换后图像中PE文件的区段分布信息，有利于增强同一家族的恶意软件图像的相似性。
* 提出了使用 CoLab 图像、VGG16和支持向量机对恶意软件进行分类的恶意软件分类方法（ MalCVS ）。
* 在 VX-Heaven 和 Virusshare 以及 Microsoft Malware Classification Challenge 数据集上收集的恶意软件样本上的实验结果表明， MalCVS 在准确率、召回率、精确率和 F1 分数方面优于基于灰度图像和基于操作码的恶意软件分类方法。

# 相关工作

## 传统的基于机器学习的恶意软件分类方法

### 基于静态特征的方法

* **优点**：
  + 不需要执行恶意软件，减少了安全风险。
  + 可以从不完整的恶意软件样本中提取特征。
* **缺点**：
  + 依赖于专业知识手动设计特征，过程耗时。
  + 面对使用混淆技术的恶意软件，静态分析可能无效。

### 基于动态特征的方法

* **优点**：
  + 能够观察到恶意软件在运行时的真实行为。
  + 能够应对一些使用混淆技术的恶意软件。
* **缺点**：
  + 动态分析过程耗时，需要诱导恶意软件执行。
  + 若恶意软件检测到虚拟环境，可能会改变其行为。

## 基于图像和深度学习的恶意软件分类方法

### 基于单一模态数据的方法

* **优点**：
  + 减少了特征工程的工作量。
  + 能够处理大量的恶意软件变种。
* **缺点**：
  + 对于某些方法，需要对PE文件进行反汇编，增加了复杂性。
  + 仅依靠单一模态数据可能无法捕获恶意软件的全部特征。

### 基于多模态数据的方法

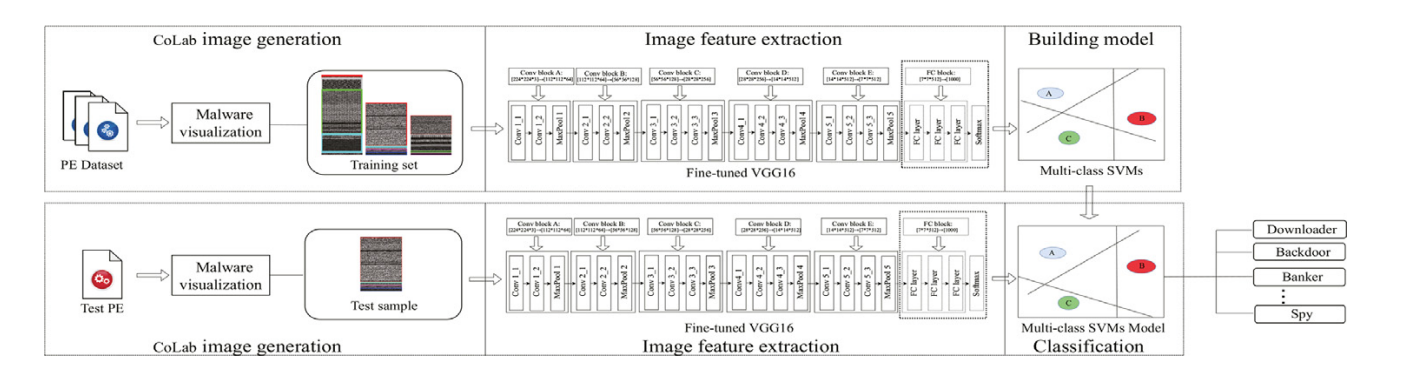
* **优点**：
  + 结合了不同类型的数据，提高了分类的准确性。
  + 能够更全面地描述恶意软件的特征。
* **缺点**：
  + 计算成本高，需要处理和分析大量的数据。
  + 需要复杂的模型来融合不同模态的特征。

# MalCVS

## 动机

相同恶意软件家族的恶意软件区段分布信息相似，但从灰度图中难以识别区段分布信息。传统的灰度图方法会依据 padding 将PE文件分为多个部分，但该方法存在将伪边界错误识别为真边界的问题。因此，该文提出一种新颖的可视化方法，该方法能够强调恶意软件图像中的区段分布信息。

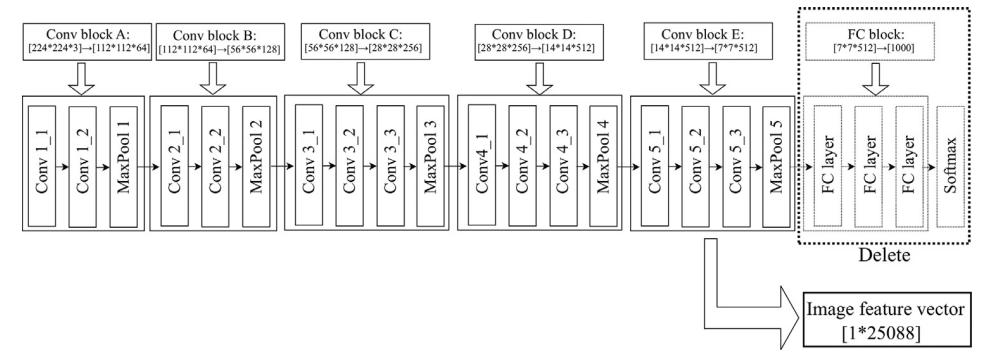
## MalCVS 框架

MalCVS 首先将PE文件转换为 CoLab 图像，然后使用微调的 VGG16 模型进行图像特征提取。随后，利用提取的特征构建多类 SVM 模型。框架如下： 

## CoLab 图像生成

将不同的区段用不同颜色的框框圈起来，已知区段名用固定的颜色，未知区段名根据区段位置按顺序用预定义的颜色集。

## 图片特征提取



微调后的VGG16

与原始VGG16相比，MalCVS中使用的微调VGG16删除了三个全连接层和一个softmax层，即只保留了之前五个用于提取图像特征的卷积模块。该结构的参数较少，因此在训练过程中的时间和空间消耗比原始VGG16要少得多。每个卷积模块包含若干个卷积层和一个最大池化层，其中卷积层用于从原始CoLab图像中提取特征。

## 分类模型构建

此训练阶段使用的每个特征向量的维数为25088。这些特征向量属于高维数据，因此处理它们可能会面临“维数灾难”和“数据稀疏性”的问题。因此，有必要采用能够处理高维数据的分类算法。

在目前流行的分类算法中， SVM 是一个不错的选择，因为它在处理高维数据方面非常有效。 SVM 的最终决策函数仅由少数支持向量决定，因此其计算复杂度取决于支持向量的数量而不是空间维数，从某种意义上避免了“维数灾难”。此外，先前的研究表明， SVM 在大型数据集上具有良好的性能，并且已被证明对恶意软件分类非常有效。 MalCVS 采用线性 SVM 构建恶意软件分类模型。经过训练阶段后，可以使用构建的分类模型确定哪种恶意软件属于哪个恶意软件家族。

# 实验评估

## 实验设置和数据集

MalCVS 的 VGG16 和 SVM 模型分别由python使用keras1和sklearn2构建。除了对 VGG16 模型进行微调外， VGG16 和 SVM 的其余参数均采用默认设置。为了评估 MalCVS 的泛化性能，我们使用10折交叉验证，即将数据集分成10个大小相等的子集，第i个子集依次作为测试数据，其余子集作为训练数据，重复10次。实验在具有16 GB内存的Intel i7-9750H机器上进行。

本文使用两个恶意软件数据集进行实验，分别是 VXV 和 BIG-2015 数据集。在 VXV 数据集中，有6398个PE文件属于16个恶意软件家族，每个文件在从VX-heaven或Virusshare下载时都会被标记为一个家族。 BIG-2015 数据集（Kaggle提供的训练数据集）包含10868个恶意软件，属于9个家族，每个恶意软件都有一个ASM文件和一个BYTE文件。请注意，由于 BIG-2015 数据集中的BYTE文件不包含文件头头信息，因此缺少节名称，因此它们的节根据每个节出现的顺序从预设的颜色集中分配颜色。此外，我们将 BIG-2015 的每个BYTE文件中的字节“??”替换为了“FF”。

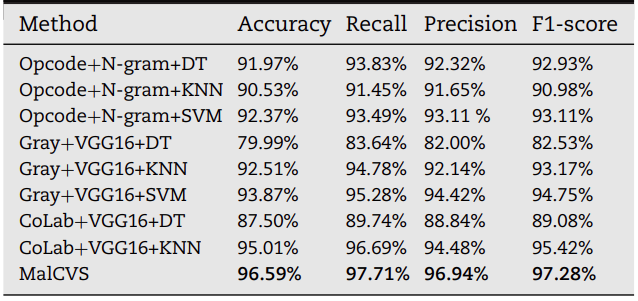
## 评估指标

四个广泛使用的评估指标，召回率、精确度、准确度和 F1 分数，用于评估分类性能，并使用混淆矩阵给出每个类别的详细分类性能。

## 实验结果

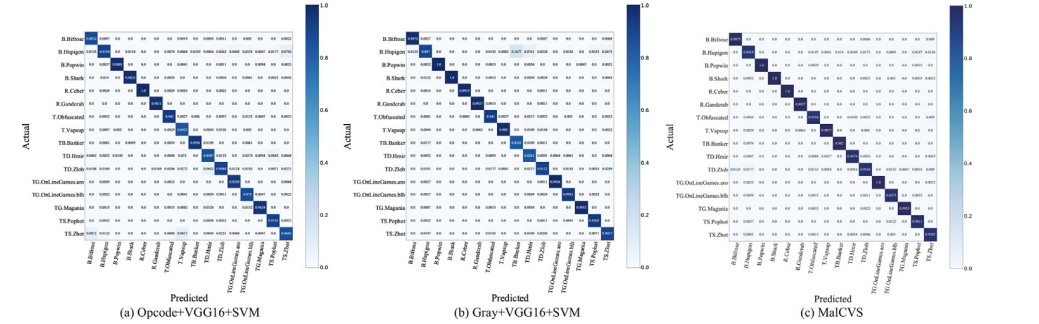
### VXV数据集实验

下图展示了 MalCVS 和基于操作码+n-gram、基于灰度图像和基于 CoLab 图像的方法在 VXV 数据集上进行 10 倍交叉验证的分类性能。 MalCVS 获得了最佳分类结果，各项指标都高于其他方法。



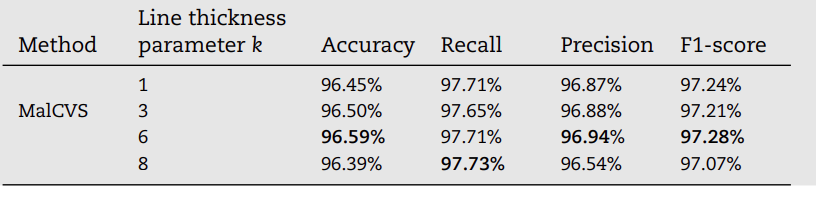
MalCVS和其他方法在VXV数据集的表现

下图展示了 MalCVS 、”Gray+VGG16+SVM” 和 ”Opcode+N-gram+SVM” 对每个恶意软件家族的混淆矩阵。从中可以看出 ”Opcode+Ngram+SVM” 对属于 ”B.Bifrose”、”T.Vapsup”、”TD.Hmir”、”TG.OnLineGames.blh” 和 ”TS.Zbot” 家族的样本分类效果不佳，而 ”Gray+VGG16+SVM” 对属于 ”B.Hupigon”、”TB.Banker” 和 ”TD.Hmir” 家族的样本识别效果不佳。至于 MalCVS ，它在所有 16 个家族中的表现都优于这两种方法，并且对 16 个家族中的每一个的准确率都高于 94%。



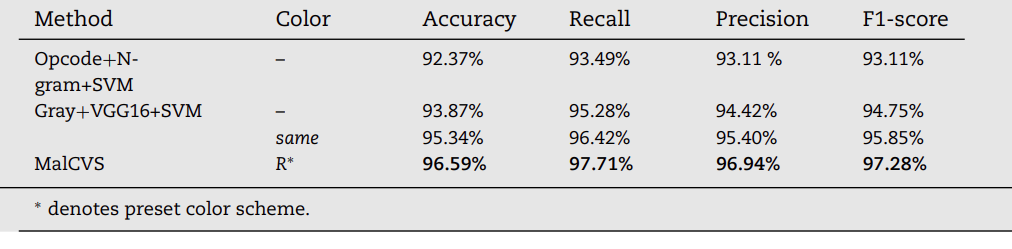
使用SVM和不同特征向量的混淆矩阵

下图展示了不同线段厚度对 MalCVS 的影响



不同线段厚度对 MalCVS 的影响

下图展示了不同颜色对 MalCVS 的影响



不同颜色对 MalCVS 的影响

# 评价

## 创新点

* 认识到相同恶意软件家族的样本在PE文件的节数目、顺序和大小上具有相似性，利用这一点来增强图像特征的表达能力。
* 提出了一种将PE文件区段分布信息融入恶意软件图像的方法。
* 结合了CoLab图像、微调后的VGG16模型和支持向量机（SVM）来构建一个高效的恶意软件分类方法（MalCVS）。