基于Ngram-TFIDF的深度恶意代码可视化分类方法

王金伟[1]，陈正嘉[1]，谢雪[2]，罗向阳[3]，马宾[4]

1.南京信息工程大学计算机学院，江苏 南京 210044；2.中国科学技术大学网络空间安全学院，安徽 合肥 230031；
3.信息工程大学网络空间安全学院，河南 郑州 450001；4.齐鲁工业大学网络空间安全学院，山东 济南 250353）

摘 要：随着恶意代码规模和种类的不断增加，传统恶意代码分析方法由于依赖于人工提取特征，变得耗时且易出错，因此不再适用。为了提高检测效率和准确性，提出了一种基于Ngram-TFIDF的深度恶意代码可视化分类方法。结合N-gram和TF-IDF技术对恶意代码数据集进行处理，并将其转化为灰度图。随后，引入CBAM并调整密集块数量，构建DenseNet88_CBAM网络模型用于灰度图分类。实验结果表明，所提方法在恶意代码家族分类和类型分类上分别提高了1.11%和9.28%的准确率，取得了优越的分类效果。

关键词：深度学习；数据可视化；恶意代码检测和分类

中图分类号：TP309

文献标志码：A

DOI：10.11959/j.issn.1000–436x.2024115

Deep visualization classification method for malicious code based on Ngram-TFIDF

WANG Jinwei[1], CHEN Zhengjia[1], XIE Xue[2], LUO Xiangyang[3], MA Bin[4]

1. School of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China
2. School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230031, China
3. School of Cyber Science and Technology, Information Engineering University, Zhengzhou 450001, China
4. School of Cyberspace Security, Qilu University of Technology, Jinan 250353, China

Abstract: With the continuous increase in the scale and variety of malware, traditional malware analysis methods, which relied on manual feature extraction, become time-consuming and error-prone, rendering them unsuitable. To improve detection efficiency and accuracy, a deep visualization classification method for malicious code based on Ngram-TFIDF was proposed. The malware dataset was processed by combining N-gram and TF-IDF techniques, transforming it into grayscale images. Subsequently, the CBAM was introduced and the number of dense blocks was adjusted to construct the DenseNet88_CBAM network model for grayscale image classification. Experimental results demonstrate that the proposed method achieves superior classification performance, with accuracy improvements of 1.11% and 9.28% in malware family classification and type classification, respectively.

Keywords: deep learning, data visualization, malicious code detection and classification

---

收稿日期：2023–11–28；**修回日期：**2024–05–27

通信作者：谢雪，xuexie2008@163.com

基金项目：国家自然科学基金资助项目（No.62072250, No.62172435, No.U20B2065）；中原科技创新领军人才基金资助项目（No.214200510019）；江苏自然科学基金资助项目（No.BK20200750）；河南省网络空间态势感知重点实验室开放基金资助项目（No.HNTS2022002）；山东省计算机网络重点实验室开放课题基金资助项目（No.SDKLCN-2022-05）

Foundation Items: The National Natural Science Foundation of China (No.62072250, No.62172435, No.U20B2065), The Leading Talents Program of Scientific and Technological Innovation in Henan Province (No.214200510019), The Jiangsu Natural Science Foundation (No.BK20200750), The Open Fund of the Key Laboratory of Network Space Situation Awareness (No.HNTS2022002), The Open Research Fund of Shandong Provincial Key Laboratory of Computer Networks (No.SDKLCN-2022-05)

0　引言

恶意代码是一种包含了恶意的代码片段或数据的计算机程序，其主要目的是在未经用户许可的情况下对操作系统内核进行破坏或实施其他危害行为。为了更好地理解恶意代码的特征、行为模式及攻击方式，研究人员常常对不同类型的恶意代码进行分类。该分类任务有助于有效地识别和分析新型恶意代码所属种类，开发出有效的防御策略和工具，从而更有效地预防和应对恶意代码的攻击。因此，恶意代码分类任务在保障信息安全方面具有重要的意义。针对不同的分类角度，恶意代码可以按照平台、类型和家族等维度进行划分。

近年来，为了应对不断增长的恶意代码，研究人员积极探索和应用各种检测和分类方法。这些方法包括机器学习算法，如随机森林、决策树、支持向量机等，以及深度学习算法，如卷积神经网络（CNN, convolutional neural network）、循环神经网络（RNN, recurrent neural network）等。这些方法广泛应用在恶意代码领域并取得了显著的成果，极大地提高了恶意代码的检测效率和准确率。特别是深度学习算法通过其强大的表征学习能力，有效地挖掘和捕捉恶意代码的隐含特征，提高了恶意代码的分类精确度。因此，这些方法已经成为恶意代码分析师的重要工具，对恶意代码的检测和分析提供了有力支持。

传统非可视化恶意代码检测和分类方法通常采用静态代码分析和动态代码分析2种技术。静态分析技术可快速获取恶意代码语法和语义信息，而不需要执行实际代码。通常，该技术使用多种静态特征进行分析和分类。例如，文献[1-3]采用应用程序接口（API, application program interface）调用序列，文献[4-7]提取字节序列的N-gram特征，文献[8]利用调用函数进行分析，文献[9-10]则使用可移植的执行体（PE, portable executable）文件头。此外，操作码频率分布、字符串签名以及控制流图等也是常用的静态特征。这些特征能够反映出代码的结构和行为，从而判断其是否为恶意代码。此外，静态检测技术还可以借助各种工具来实现，例如交互式反汇编器专业版（IDA Pro, interactive disassembler professional）等反汇编工具可以用于逆向分析恶意可执行文件，提供更有效的信息；LordPE内存转储工具可以在系统内存中获取受保护的代码，对分析有更大的帮助。相比于静态分析，动态分析技术通过在虚拟环境中执行代码来获取恶意代码的行为报告，包括函数调用监测、功能参数分析、信息流跟踪、指令跟踪和动态可视化分析等。该技术需要使用自动化工具来实现，例如，文献[11]使用Anubis，文献[12]使用CWSandbox。此外，TTAnalyzer、Ether和ThreatExpert等也是常见的自动化工具。2种技术都有优缺点，静态分析具有时间复杂度和空间复杂度较低、速度快、效率高的优势，并且可以全面地对恶意代码进行分析，捕获语法和语义信息，但在面对混淆和加壳代码时可能会发生漏检；动态分析技术更加准确和有效，但需要投入更多时间和空间成本。

近年来，可视化方法作为一种新兴的恶意代码检测和分类方法备受关注。恶意代码二进制文件中包含了大量人类难以理解的二进制代码，将它们转换成图像后，能够可视化其结构和特征，从而方便分析和研究。通过可视化可以发现，恶意代码图像中蕴含着丰富的信息。同一类别恶意代码通常具有相似的可视化图像，而不同类别的可视化图像则有明显的差异。此外，将二进制文件转换成图像的方法还为使用计算机视觉和深度学习方法进行恶意代码分类和检测提供了可能。相较于传统的特征提取方法，可视化方法减少了特征提取过程设计的复杂度，满足大数据计算、专家系统反馈和认知复杂性等方面的需求，从而可以更加高效地检测和分类恶意代码。

目前，关于可视化方法的研究主要集中在机器学习和深度学习。然而，这2个方面都存在一定的问题和挑战。在机器学习方面，手动提取和选择特征的过程较为复杂和耗时，同时分类效果受特征质量的影响较大。而在深度学习方面，部分分类模型存在特征提取能力偏弱的问题，这导致可视化方法或网络模型结构相对复杂。另外，目前针对恶意代码家族分类任务而设计的深度学习方法较多，针对恶意代码类型分类任务的方法则相对较少。

为了应对上述问题，本文采用可视化思想，并结合二进制程序静态文件结构，提出了一种基于Ngram-TFIDF的可视化方法。通过与相关文献实验结果对比可知，本文提出的基于Ngram-TFIDF的深度恶意代码可视化分类方法在恶意代码家族分类和恶意代码类型分类2个任务上均表现出优越的性

能。在后续的实验中，本文选择了包含加壳操作的PE数据集以及不包含加壳操作的Big2015数据集[13]进行实验验证。实验结果证明，无论恶意代码是否经过加壳操作，本文方法都能够提供可靠的分类结果，在实际应用中具有广泛的适用性和可靠性。通过在多个数据集上的实验验证，本文进一步确认了该方法的稳定性和有效性，为其在实际场景中的应用奠定了坚实的基础。本文主要贡献如下。

1) 本文提出了一种预处理方法。对于恶意代码二进制文件，本文首先将其转化为Ngram-TFIDF灰度图，随后利用像素值乘积以增强图像亮度，从而加强了恶意代码图像的辨别性，提升了恶意代码分类效果。实验结果表明，这种预处理方法可以大大提高恶意代码家族分类和恶意代码类型分类的准确率。

2) 本文通过引入卷积块注意力模块（CBAM, convolutional block attention module）和调整密集块的数量，设计了DenseNet88_CBAM网络模型用于恶意代码分类任务。

3) 实验结果表明，本文设计的基于Ngram-TFIDF的深度恶意代码可视化分类方法在恶意代码家族分类和恶意代码类型分类2个任务上均表现出优越的性能。特别是在恶意代码类型分类任务上，相较于先前的方法，本文方法准确率提高了9.28%。

1 相关工作

1.1 基于机器学习的恶意代码可视化检测

近年来，研究人员针对恶意代码可视化结合机器学习的方法已经展开了广泛而深入的研究，主要致力于提取出能够实现良好分类效果且不易受到干扰的恶意代码可视化特征。在特征提取的基础上，可以利用多种分类器对图像进行分类，从而更好地实现恶意代码的分类。Nataraj等[14]的研究开启了结合可视化技术的恶意软件检测和分类的新兴领域。该研究首先将恶意软件.text区块的二进制数据通过Nataraj矢量化技术转化为灰度图，然后基于全局特征信息（GIST）算法对转化后的灰度图特征进行提取，最后使用K最近邻（KNN, k-nearest neighbor）算法对提取的特征进行分类。此外，文献[15]使用图像处理的二进制纹理分析技术，可以更快速地对恶意软件进行分类。然而，由于纹理分析方法具有较大的计算开销，因此在处理大量的恶意软件时存在问题。Liu等[16]提出了一种基于机器学习的恶意软件分析系统，该系统主要由数据处理模块、决策模块和检测模块3个部分构成。数据处理模块利用操作码N-gram和导入函数对灰度图进行特征提取；决策模块负责对恶意软件进行分类和识别可疑恶意软件；检测模块使用共享最近邻（SNN）聚类算法来发现新的恶意软件家族。Fu等[17]提出了一种通过将熵、字节值和相对大小这3个特征分别映射到RGB三通道的方法，把恶意软件可视化为RGB彩色图。在特征提取方面，该方法采用了全局特征（如灰度共生矩阵（GLCM, Gray-level co-occurrence matrix）和颜色矩）和局部特征（如部分字节码序列），并使用随机森林、KNN和支持向量机等机器学习方法对恶意软件进行分类，不仅提高了模型的鲁棒性，而且还为研究者提供了一种新的可视化手段，能够更加直观地展示恶意软件的特征。刘亚姝等[18]解决了文献[14]在某些相似度比较高的恶意代码家族上分类精确度不高的问题。作者首先将恶意代码根据文献[14]中的Nataraj矢量化方法转化为灰度图，之后选取GIST作为全局特征，局部二值模式（LBP, local binary pattern）或者稠密SIFT（dense SIFT, dense scale-Invariant feature transform）作为局部特征进行融合，构造抗混淆、抗干扰的特征。实验结果表明，该方法更具稳定性和适用性，并且分类准确率得到了明显提高。郎大鹏等[19]提出了一种融合了3组特征的恶意软件检测和分类算法。该算法首先将恶意代码源文件和反汇编文件使用文献[14]中的Nataraj矢量化方法转化为灰度图；然后提取灰度图的GIST特征和GLCM特征，使用N-gram算法提取操作码序列，并采用改进型增益算法提取操作码特征；最后将这3组特征进行组合与分类，并利用随机森林算法进行学习。

1.2 基于深度学习的恶意代码可视化检测

近年来，基于深度学习的恶意代码检测得到了较快的发展。其中，常用的方法是将恶意代码转换为灰度图或彩色图，并将其输入深度学习网络中进行学习和分类。Kalash等[20]利用文献[14]的方法将恶意软件的二进制文件转换为灰度图，并使用卷积神经网络进行分类，最后在Malimg和Big2015这2个数据集上进行了验证，结果显示，其分类准确率分别高达98.52%和98.99%。由此可见，该方法

具有较高的分类精确度，能够有效地检测恶意软件。Vasan等[21]在恶意代码检测领域提出了一种将原始恶意代码二进制文件转换为彩色图的方法，首先使用文献[14]中的Nataraj矢量化方法将恶意软件转化为二维数组，然后添加彩色映射生成彩色图，最后利用调整后的VGG16网络模型对其进行检测和分类。该方法可以高效地识别混淆的恶意软件及其变种，具有很高的效率和实用性。王润正等[22]采用了反汇编工具提取恶意代码中的不同区块数据，并对代码段和数据段进行分离和可视化操作。由于每个区块代表的恶意代码信息不同，这种方法可以更直观地展现不同恶意家族之间的差异性。实验结果表明，采用这种方法可以进一步提高恶意代码分类的准确性。Pinhero等[23]提出了一种通过结合恶意软件可视化和深度学习分类的方法。该方法通过对不同维度的可执行文件进行视觉图像分析，评估了微调的卷积神经网络模型在恶意软件识别中的性能。结果表明，使用彩色视觉图像和马尔可夫图像可以取得最佳效果，同时应用Gabor滤波器提取纹理特征和熵图像进一步提高了分类准确性。Anandhi等[24]将恶意软件可视化为马尔可夫图像，并利用Gabor滤波器提取纹理特征。通过构建VGG3和微调DenseNet模型，实现了实时恶意软件检测和分类，在Malimg和BIG2015这2个数据集中分别取得99.94%和98.98%的准确率。Huang等[25]提出了一种结合恶意软件可视化技术和卷积神经网络的混合可视化方法，通过Cuckoo Sandbox生成行为分析报告，将静态特征和动态信息转换为可视化图像进行训练。实验结果表明，混合可视化方法在恶意软件检测方面具有较好的性能，优于仅使用静态代码分析的检测方法。Moussas等[26]提出了一种基于两级人工神经网络（ANN, artificial neural network）的新型恶意软件检测系统，旨在解决恶意软件图像变体的识别问题。通过从数据集中提取关键图像特征，并利用这些特征训练ANN模型。该系统不仅能够检测和分类数据集中的恶意软件样本，还通过第二级ANN进一步分类易混淆的恶意软件家族。该两级ANN模型在简洁性、准确性和速度方面表现出色，且易于实现和快速运行，适用于防病毒软件、智能防火墙和Web应用等领域。Darem等[27]结合深度学习、特征工程和图像转换和处理技术，提出了一种半监督方法用于检测混淆恶意软件，取得了99.12%的准确率，显著优于其他方法。Conti等[28]提出了一种以GEM图像为基础的深度学习方法，适用于较为简洁的卷积神经网络架构，以提升训练和分类效率。该方法将灰度矩阵图像、GLCM纹理特征、马尔可夫图像和熵图像结合，提出了一种新颖的恶意软件可视化方案，并与现有的卷积神经网络架构相容。Falana等[29]提出了一种基于深度卷积神经网络和深度生成对抗神经网络的集成方法Mal-Detect，用于恶意软件分析、检测与分类。Mal-Detect将恶意软件及良性文件转换为RGB图像，并通过深度生成对抗神经网络生成新的恶意软件图像，随后对生成的恶意软件图像、原始恶意软件图像及良性文件图像进行预处理，利用深度卷积神经网络训练数据集并提取关键特征，在MaleVis、Malimg和Virushare这3个基准数据集上的平均准确率为96.77%。Chaganti等[30]提出了EfficientNetB1框架用于恶意软件家族分类。该框架通过评估3种恶意软件图像表示，优化了模型参数，并全面评估了基于卷积神经网络的预训练模型性能和效率。研究结果表明，采用固定图像宽度的字节级表示的恶意软件图像在性能上优于基于文件大小和字节编码的图像宽度选择的表示。Mallik等[31]提出了一种基于卷积循环的恶意软件分类方法。该方法首先将恶意软件样本转换为灰度图，然后利用卷积神经网络捕捉其结构相似性，从而实现高效分类，最后通过堆叠双向长短期记忆（Bi-LSTM, bidirectional long short-term memory）网络层处理提取的特征，并将Bi-LSTM层与VGG16层的输出相结合，对恶意软件样本进行最终的家族分类。Chauhan等[32]利用不同颜色模式（包括RGB、HSV、灰度和BGR）将恶意软件文件转化为图像进行分类，结果表明，该方法在分类准确率、结果一致性、召回率、F1得分和精确度方面表现出色。

2　所提方法

本节详细介绍了本文提出的基于Ngram-TFIDF的深度恶意代码可视化分类方法。该方法主要包括3个部分，即恶意代码可视化、神经网络模型构建以及模型训练与评价，整体流程如图1所示。首先，阐述了N-gram和TF-IDF的相关知识。然后，详细介绍了提出的恶意代码可视化方法和构建的DenseNet88_CBAM网络模型。

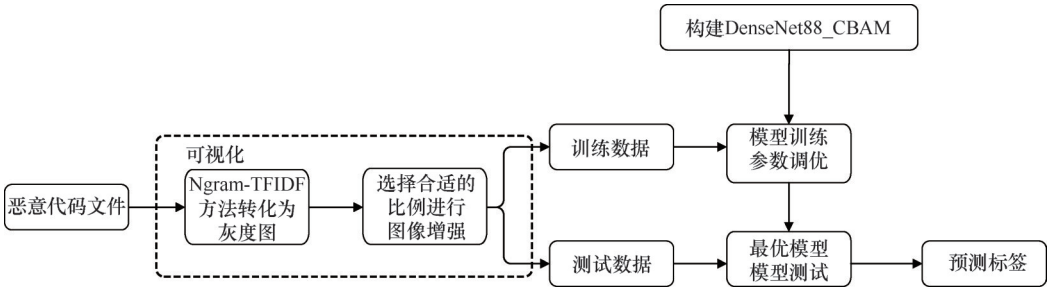

图 1 基于 Ngram-TFIDF 的深度恶意代码可视化分类方法流程

2.1 N-gram

N-gram 是一种基于统计语言模型的算法，其核心思想是将文本划分为长度为 N 的连续字节片段，这些字节片段被称为 gram，并且对 gram 出现频率进行统计。通过设定阈值进行过滤，形成关键 gram 列表，即文本的向量特征空间，其中每个 gram 代表一个特征向量维度。考虑到恶意代码十六进制文件的庞大性，本文选择将十六进制数字中长度为 n 的所有数字组合作为一个特定 n 值对应的 N-gram 总特征列表。对于给定的 n 值，总特征列表的数量为所有可能的子序列组合数，即 16^n 个特征。

2.2 TF-IDF

词频反文档频率（TF-IDF, term frequency inverse document frequency）[33] 结合了词频（TF, term frequency）计算式和逆向文件频率（IDF, inverse document frequency）计算式。

在一份给定的文件中，词频表示某个特定词语在该文件中出现的次数。词频的计算式为

$$\mathrm{tf}_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

其中，$n_{i,j}$ 为在给文件 d_j 中，该特定词语出现的次数，$\sum_k n_{k,j}$ 为在文件 d_j 中所有词语的个数。

逆向文件频率是一种衡量词语普遍重要性的度量。给定一个特定词语，其 IDF 可以通过文件总数除以包含该词语的文件数目再取对数计算，如式(2)所示。

$$\mathrm{idf}_i=\lg\frac{|D|}{|\{j:t_i\in d_j\}|} \tag{2}$$

其中，$|D|$ 为文件集或是语料库中的文件总数，$|\{j:t_i\in d_j\}|$ 为包含词语 t_i 的文件数目。

最终所求词语的 $\mathrm{tfidf}_{i,j}$ 是词频和逆向文件频率的乘积，即

$$\mathrm{tfidf}_{i,j}=\mathrm{tf}_{i,j}\mathrm{idf}_{i,j} \tag{3}$$

式(3)表明当一个词语在一篇文章中频繁出现而在其他文章中出现较少时，其 TF-IDF 值较大，这表明该词语对该篇文章的重要性较高。通过 TF-IDF 值的计算，能够更准确地衡量词语对文档的重要性，并实现文本数据的有效区分和分析。

2.3 恶意代码可视化

本文提出的可视化过程包括以下 2 个步骤：1) 利用 Ngram-TFIDF 方法将原始数据集转化为灰度图，以便更好地理解二进制恶意代码文件的特征；2) 通过像素值乘积实现图像增强，提升图像质量。

1) Ngram-TFIDF 方法转化为灰度图

采用 Ngram-TFIDF 方法将二进制恶意代码文件转化为灰度图。该方法结合了 N-gram 和 TF-IDF 特征提取技术，用于处理字节信息并生成灰度图。具体而言，将恶意代码文件转化为十六进制字节文件；在 N-gram 方法中选择适当的 n 值，构建由十六进制数字组成的 N-gram 特征集合；利用这些 N-gram 特征集合；选择一定数量的恶意代码文件，计算它们相应特征的 TF-IDF 值，并进行累加；从累加的值中选择 256 个 TF-IDF 值最大的特征，这些特征能够最佳地代表恶意软件的分类特征。

为了进一步说明，假设 N-gram 中的 n=4。在十六进制文件中，共有 65 536 个可能的特征。然后，随机选择 m 个文件，计算它们相对于这 65 536 个特征的 TF-IDF 值，并将这些值进行累加。最后，从累加值中选择 256 个 TF-IDF 值最大的特征，形成特征列表。特征选择示意如图 2 所示。

针对每个十六进制文件，将其按照每组 1 000 个十六进制位数进行分割，从而生成一个宽度为 1 000 的二维数组。接下来，基于挑选出的 256 个 N-gram 特征列表，对二维数组的每一行进行 TF-IDF 值的计算。通过这一步骤，每个恶意代码文件

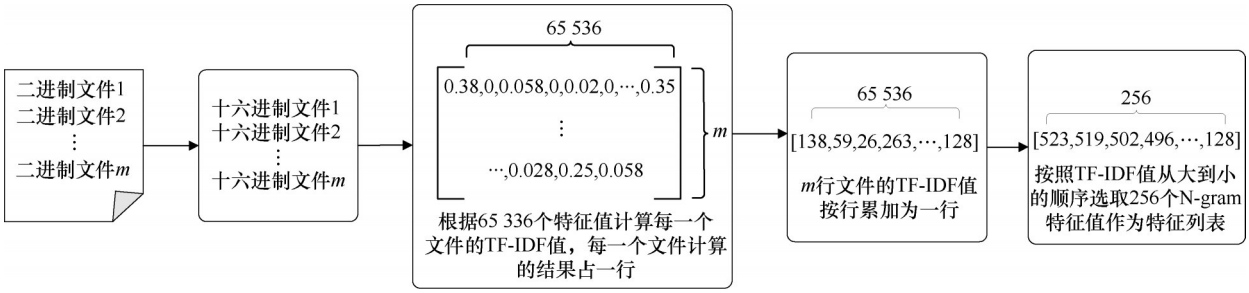

图2　特征选择示意

可以转化为一个宽度为256的二维数组。最后，对新的二维数组中的每个元素进行归一化处理，将其除以数组中的最大值，并乘以255，以将数组中的每个元素映射到[0, 255]的灰度值范围内。其中，像素值为0表示黑色，像素值为255表示白色。通过这一转换过程，可以成功地将二进制恶意代码文件转化为具有256宽度的长方形灰度图，这为后续的图像处理和分类任务提供了便利。具体的转换过程如图3所示。

将N-gram方法与TF-IDF方法结合应用于恶意代码分类任务可以带来多个好处。首先，恶意代码往往包含特定的关键词，这些关键词的组合和出现频率对准确分类具有重要作用。通过使用N-gram方法，可以考虑关键词之间的顺序和组合，捕捉到恶意代码中的特定模式和语法结构，从而提高模型对恶意代码的识别能力。其次，TF-IDF方法可以帮助区分恶意代码中常见和罕见的特征词语。恶意代码往往包含一些特定的关键词，这些词语在恶意代码集合中出现的频率较高，但在正常代码集合中较为罕见。通过使用TF-IDF方法，可以提取这些罕见但在恶意代码分类中具有重要作用的特征词语，使得模型能够更好地将恶意代码与正常代码区分开来。

2) 像素值乘积

在转化过程中，存在图像存在黑色像素点比例较高、白色像素点比例较低且暗淡的问题。这种情况可能会影响恶意代码分类的效果，因为白色像素点比例较低会掩盖可视化灰度图的细节特征。为了解决这一问题，本文采用了像素值增强的方法，即将使用Ngram-TFIDF转化的灰度图中每个像素值乘以一个增强比例系数（若像素值超过255，则截断为255），该方法可以有效增大白色像素点的比例，增加图像亮度，使恶意代码图像更加清晰。

研究过程中，本文对Big2015数据集与PE数据集中训练集的所有使用Ngram-TFIDF转化的灰度图进行了统计。在Big2015数据集和PE数据集中，增强前白色像素点占比平均值仅为0.000 13%和0.16%。然而，将矩阵中每个像素值按照最佳增强比例进行增强后，白色像素点的比例显著增大，分别增大到了10.99%和15.89%。由此可见，增强图像可以大幅度增大白色像素点所占比例，从而更加突出可视化灰度图的特征，提升分类效果，有助于分类模型更好地学习和识别恶意代码的特征。

上述二维数组中的每个元素除以数组中的最大值并乘以255以及图像增强操作过程如式(4)所示。

$$x' = \frac{x}{\max\_x} \times 255 \times a \tag{4}$$

其中，x表示二维数组中的每一个数值，$\max_x$表示数组中所有数值的最大值，a表示增强比例系数。为了进一步展示图像增强的效果，方便读者理解，图4给出了图像增强效果示意。

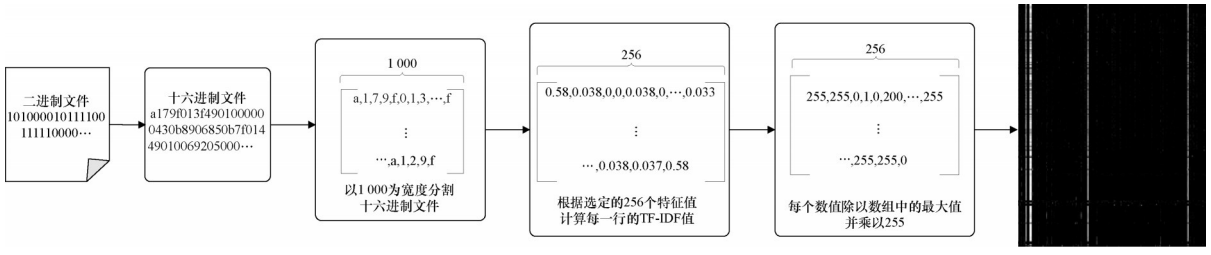

图3　具体的转换过程

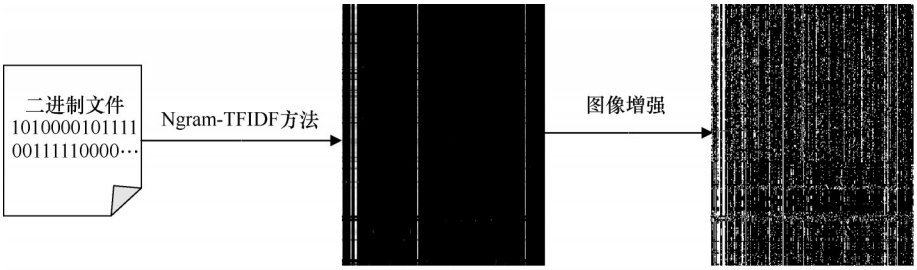

图4　图像增强效果示意

2.4　神经网络模型构建

本文旨在构建的模型需要在小型和大型数据集上均适用，具有较强的鲁棒性，且拥有较高的分类准确率和高效的特征提取能力。为此，本文构建了一种新型网络模型——DenseNet88_CBAM（由87个卷积层和1个全连接层组成）。该模型在DenseNet121[34]网络模型的基础上进行了改进，主要包括引入CBAM[35]和调整密集块的数量。

为了方便读者理解DenseNet88_CBAM模型，本文绘制了该模型的整体结构，如图5所示。

表1详细地描述了图5中DenseNet88_CBAM整体的网络结构参数，其中，num_classes表示多分类任务需要分类的种类数。

DenseNet88_CBAM网络模型与DenseNet121网络模型在密集块的数量上有所不同。具体而言，标准的DenseNet121网络模型包含4个密集块，而DenseNet88_CBAM网络模型的密集块数量为3个，这3个密集块由DenseNet121的前3个密集块组成，分别包含6、12、24个密集层。由于密集块的数量减少，DenseNet88_CBAM网络模型可能在特征提取的深度和复杂性方面相对于DenseNet121略有减少。然而，通过精心选择的3个密集块，DenseNet88_CBAM网络模型仍然能够充分利用密集连接的优势，实现特征的高效复用和信息的增加。这种结构调整在一定程度上平衡了模型的复杂性和计算资源的需求，同时保持了较高的特征表达能力和学习能力。

表1　DenseNet88_CBAM整体的网络结构参数

网络结构	输出大小/像素×像素	输出通道数/个	具体结构
Conv	112×112	64	7×7,/2
Maxpool	56×56	64	3×3,/2
Dense Block1（6个密集层）	56×56	256	[1×1 Conv]×6 [3×3 Conv]×6
Transition1	56×56	128	1×1 Conv
	28×28	128	2×2 Avgpool,/2
Dense Block2（12个密集层）	28×28	512	[1×1 Conv]×12 [3×3 Conv]×12
Transition2	28×28	256	1×1 Conv
	14×14	256	2×2 Avgpool,/2
Dense Block3（24个密集层）	14×14	1 024	[1×1 Conv]×24 [3×3 Conv]×24
Avgpool	1×1	1 024	7×7
FC	—	num_classes	—

更重要的是，在小型数据集中，由于数据样本数量有限，模型往往面临过拟合和学习不充分的挑战。然而，DenseNet88_CBAM通过精心设计的模型简化策略，有效降低了模型的复杂度，避免了在小型数据集上拟合数据过多的问题。同时，由于模型参数量的减少，DenseNet88_CBAM在数据集上的训练速度也更快，能够更快地收敛和学习数据的特征。

此外，值得注意的是，为了进一步提升DenseNet88模型的性能，本文在网络结构中添加了CBAM。具体而言，在DenseNet88_CBAM模型的起始部分之前和所有密集块处理完成后的关键位置添加CBAM。这样的设计考虑了特征提取的不同阶段和层次，充分利用了CBAM的自适应特性和信息整合能力。在起始部分之前的添加可以在输入图像经过初始卷积和池化操作之前，就对其进行全局的通道和空间关注，从而引导网络更有针对性地提取重要的特征。而在所有密集块处理完成后的添

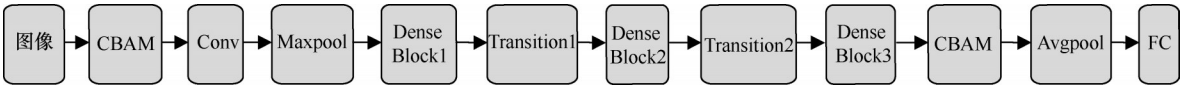

图5　DenseNet88_CBAM整体结构

加，则在特征已经逐层丰富和复杂的情况下，对最终特征图进行全局的通道和空间关注，进一步提升特征表达和判别能力。通过这种巧妙的安排，DenseNet88_CBAM模型能够更好地挖掘和利用图像中的重要信息，并在特征提取和分类任务中取得更为优异的性能。

综上所述，DenseNet88_CBAM模型通过综合运用注意力机制、密集连接和过渡层的策略，充分利用了逐层连接、特征复用和信息增强等优势，实现了高效而准确的特征提取与分类，为图像识别领域的研究和应用提供了有益的参考和启示。

3　实验与结果分析

3.1　数据集

本文研究的主要目标是针对恶意代码家族分类和恶意代码类型分类2个任务进行分类方法的设计与优化。为了实现这一目标，本文在实验中采用了2个不同的数据集。

3.1.1　恶意代码家族分类数据集

在恶意代码家族分类任务中，本文在恶意代码数据集Big2015上进行了实验。该数据集包含来自9个不同恶意代码家族的10 868个样本，每个样本都包含一个20个字符的哈希ID和一个表示家族的整数标签，如表2所示。

表2　Big2015数据集9个恶意代码家族及其数量

恶意代码家族	数量/个
Ramnit	1 541
Lollipop	2 478
Kelihos_ver3	2 942
Vundo	475
Simda	42
Tracur	751
Kelihos_ver1	398
Obfuscator.ACY	1 228
Gatak	1 013
总计	10 868

Big2015数据集中的每个恶意样本包含2个文件，分别是.byte文件和.asm文件。其中，.byte文件是十六进制表示的、去除PE头的二进制文件，.asm文件则是通过IDA Pro反编译后生成的元数据文件，其中包含了恶意样本的机器码、汇编指令等信息。在本文的Ngram-TFIDF方法中，选择使用.byte文件进行分析和处理。这是因为.byte文件能够提供恶意样本的二进制表示，这些表示对特征提取和分析具有重要意义。

3.1.2　恶意代码类型分类数据集

在恶意代码类型分类任务中，由于当前公开的恶意代码类型数据集较为有限，因此本文构建了一个非公开的数据集，包含了5 094个恶意代码PE文件，这些文件均来自企业内部。该数据集扩展了现有公开数据集的规模和多样性，为恶意代码类型分类任务提供了更加完整的数据支持。

本文的数据集覆盖了6个恶意代码类型，分别为后门（Backdoor）、通用（Generic）、木马（Trojan）、变形（Variant）、病毒（Virus）和蠕虫（Worm）。由于该数据集中的所有文件类型均为PE文件，为了方便描述，本文将该数据集称为PE数据集。表3列出了PE数据集6个恶意代码类型及其数量。

表3　PE数据集6个恶意代码类型及其数量

恶意代码类型	数量/个
Backdoor	898
Generic	898
Trojan	808
Variant	898
Virus	709
Worm	883
总计	5 094

3.2　实验结果与分析

本文实验分为两部分：第一部分为消融实验，其中包括Ngram-TFIDF方法细节消融实验、图像增强比例消融实验和网络结构消融实验；第二部分为对比实验，旨在将本文方法与其他效果较好的方法进行比较分析。

在消融实验和对比实验中，本文将批处理样本数设置为128，初始权重设置为随机，优化器设置为SGD优化器，损失函数设置为交叉熵损失函数。本文采用经验学习率值0.01，并使用余弦退火调度器将optimizer的学习率从初始值降低到最小值，再将其逐渐恢复到初始值，这一过程每200个epoch执行一次。结果显示，该调度器能够提高模型的训练效果，使其更快地收敛到最优解。

本文对Big2015数据集和PE数据集进行了随机抽样，并将其中80%的样本作为训练集，另外

10%的样本用于测试集，剩余10%的样本用于验证集。每个模型每次训练采用200个epoch，并将每次训练在验证集上的最高分类准确率作为本次训练的分类准确率。由于实验的随机性，本文对每个模型进行了10次独立的训练，并将这10次实验结果的分类准确率取平均值作为该模型的表现评估结果，从而降低了实验误差的影响。

3.2.1 消融实验

本节展示了3个消融实验：Ngram-TFIDF方法细节消融实验、图像增强比例消融实验和网络结构消融实验。

1) Ngram-TFIDF方法细节消融实验

本文提出了一种预处理方法，旨在采用Ngram-TFIDF方法将二进制恶意代码文件转化为灰度图。为了评估预处理方法中N-gram不同n值以及用于特征选择的文件数量对图像分类任务的影响，本文进行了多种组合实验。具体而言，在Big2015数据集和PE数据集上，本文尝试了3种n值（3、4、5）和2种特征选择文件数量（每个类随机选取200个文件和每个类选取全部文件）用于组合实验，其中图像均未进行增强操作。在实验中，本文采用DenseNet88_CBAM作为网络模型进行实验。表4显示了N-gram中不同n值以及用于特征选择的文件数量对分类准确率的影响。

表4 N-gram中不同n值以及用于特征选择的文件数量对分类准确率的影响

数据集	文件数量	n=3	n=4	n=5
Big2015数据集	每个类选取200个	98.59%	98.50%	98.12%
	每个类选取全部	98.87%	98.68%	98.40%
PE数据集	每个类选取200个	76.37%	74.25%	74.16%
	每个类选取全部	77.82%	76.63%	74.87%

通过分析表4，可以得出以下结论：在横向观察中，不论是恶意代码家族分类任务还是恶意代码类型分类任务，也不论用于特征选择的文件数量如何，当N-gram中n=3时，相比其他n值而言，都能进一步提高分类准确率；在纵向观察中，不论是恶意代码家族分类任务还是恶意代码类型分类任务，也不论N-gram中n的取值如何，每个类选取全部文件用于特征选择相较于每个类随机选取200个文件用于特征选择，都能进一步提高分类准确率。因此，在本文所采用的数据集实验中，使用n=3的N-gram，并将所有文件用于特征选择，可获得最佳的分类效果。

2) 图像增强比例消融实验

由于Ngram-TFIDF灰度图本身呈现较暗的特点，因此需要进行图像增强以凸显不同恶意代码家族与恶意代码类型的特征。通过增强图像的明暗交错特征，可以为分类模型提供更具辨识度的特征。

本文采用遍历的方式进行图像增强比例的选取。具体地，将数据集转化为Ngram-TFIDF灰度图（N-gram中n=3，将所有文件用于特征选择）后，对每个像素乘以增强比例a进行图像增强。对于Big2015数据集和PE数据集，$a=10^b$，其中b选取[0,20]的偶数。随后对每一个a值所对应的增强灰度图使用DenseNet88_CBAM进行训练，记录对应分类准确率，如表5所示。

表5 图像增强比例对分类准确率的影响

图像增强比例	Big2015数据集	PE数据集
10^0	98.87%	77.82%
10^2	99.06%	78.87%
10^4	98.97%	79.93%
10^6	99.06%	80.28%
10^8	98.50%	77.99%
10^{10}	99.15%	79.05%
10^{12}	98.96%	81.51%
10^{14}	99.34%	80.63%
10^{16}	99.06%	83.45%
10^{18}	99.15%	80.81%
10^{20}	98.87%	82.75%

由表5可知，对于Big2015数据集，当b=14时，分类准确率均达到最高值，因此本文建议在对Big2015数据集进行图像增强时，选择10^{14}作为增强比例，以达到最佳的分类效果；对于PE数据集，当b=16时，分类准确率均达到最高值，因此本文建议在对PE数据集进行图像增强时，选择10^{16}作为增强比例，以获得最佳的分类效果。这一结论具有一定的稳定性和普适性。

直接转化的Ngram-TFIDF灰度图过于黑暗，不易区分，因此，在应用像素值图像增强技术时，将图像的明度值调整到合适的范围可以使得各个代码类别之间的特点更加明显，从而能够提供更多有用的特征以支持模型进行准确分类。

3) 网络结构消融实验

本节的实验展示了不同网络结构对恶意代码家族分类任务和恶意代码类型分类任务的影响。经过实验分析发现，本文提出的DenseNet88_CBAM模型相比于其他模型在恶意代码分类准确率和计算效率上有一定的提升。

实验采用先前建立的灰度图（N-gram中n=3，将所有文件用于特征选择并进行图像增强）进行训练和分类实验验证，探讨不同网络模型对恶意代码家族分类和恶意代码类型分类任务的影响。在初步筛选中，本文发现ResNet18[36]、DenseNet121、DenseNet169、VGG16[37]以及Res2Net50[38]等网络结构的分类效果较好。因此，本文选取这5个网络结构与本文提出的DenseNet88_CBAM进行对比。

实验过程中，本文引入每秒浮点运算次数（FLOPS, floating point operation per second）作为衡量网络结构复杂度的指标，同时考虑参数量作为衡量网络结构大小的指标。

接下来，本文将综合考虑分类准确率、训练一轮所需时间、网络结构复杂度以及网络结构大小等多个因素，对各个网络结构的性能进行全面评估，实验结果如表6所示。分析表6可知，DenseNet88_CBAM模型在恶意代码家族分类和恶意代码类型分类2个任务中均表现出最高的分类准确率、最少的参数量、第二短的训练时间以及第二小的模型复杂度，具有精确度高、训练简单、模型复杂度低等优点，在恶意代码分类领域具有广泛的实际应用前景。

4) DenseNet_CBAM模型细节消融实验

为了对DenseNet_CBAM模型中注意力机制添加细节以及模型深度进行深入研究，本文设计了多组实验进行对比和验证。

首先，为了探究在模型中添加注意力机制的不同位置对性能的影响，本文设计了3种位置的CBAM用于实验，具体如下所示。

①在DenseNet121模型的初始化处理之前和所有密集块处理完成后添加CBAM，为了便于叙述，称其为DenseNet121_CBAM1。

②在DenseNet121模型的初始化处理后和所有密集块处理完成后添加CBAM，为了便于叙述，称其为DenseNet121_CBAM2。

③在DenseNet121_CBAM1的基础上，对每个密集块中的每个密集层都添加一个CBAM，为了便于叙述，称其为DenseNet121_CBAM3。

其次，为了探究注意力机制的有无以及不同类型的注意力机制在模型中的效果，本文在实验中添加了以下模型。

①DenseNet121：不添加任何注意力机制的DenseNet121模型。

②DenseNet121_SE：添加了SE（squeeze-and-excitation）注意力机制的DenseNet121模型（SE注意力机制的添加位置与DenseNet121_CBAM1模型中添加注意力机制的位置保持一致）。

此外，为了探究模型深度对性能的影响，将本文提出的DenseNet88_CBAM用于实验，该模型与DenseNet121_CBAM1模型的唯一区别在于网络结构的深度不同。具体而言，DenseNet88_CBAM模型由87个卷积层和1个全连接层组成，DenseNet121_CBAM1模型则由120个卷积层和1个全连接层组成。通过对比这2个模型的性能表现，能够进一步了解网络结构深度在模型性能和准确率方面的重要性。

表6　　网络结构对分类准确率的影响

网络结构	Big2015数据集				PE数据集			
	分类准确率	训练一轮所需时间/s	FLOPS/GB	参数量/MB	分类准确率	训练一轮所需时间/s	FLOPS/GB	参数量/MB
ResNet18	98.78%	640	1.818 6	11.181 1	78.02%	132	1.818 6	11.179 6
DenseNet121	99.09%	682	2.864 7	6.963 1	80.63%	137	2.864 7	6.960 0
DenseNet169	98.97%	738	3.396 4	12.499 5	80.28%	162	3.396 4	12.494 5
Vgg16	98.87%	708	15.506 8	134.305 9	80.28%	150	15.506 8	134.293 6
Res2Net50	98.97%	700	4.203 5	23.029 3	78.40%	145	4.203 5	23.023 1
DenseNet88_CBAM	99.34%	679	2.661 6	4.409 9	83.45%	135	2.661 6	4.406 9

在实验中，本文将DenseNet88_CBAM作为实验组，并将DenseNet121（不添加注意力机制）、DenseNet121_CBAM1、DenseNet121_CBAM2、DenseNet121_CBAM3和DenseNet121_SE作为对照组。本文使用先前建立的灰度图（N-gram中n=3，将所有文件用于特征选择并进行图像增强）对设计的模型进行训练和验证，结果如表7所示。

通过对表7结果的分析可以发现，在探究添加注意力机制位置的相关实验中，不同的添加位置对模型性能产生了不同的影响。在这3种位置中，DenseNet121_CBAM1在恶意代码家族分类和恶意代码类型分类2个任务中均展现出最高的准确率。通过比较DenseNet121_CBAM1和DenseNet121_CBAM2模型可以发现，在DenseNet121模型的初始化处理之前添加CBAM能够取得更好的效果。这是因为通过在初始化处理之前引入CBAM，模型能够在恶意代码的初始特征提取阶段就更加注重关键特征，并有效地抑制或忽略恶意代码中的无关或噪声特征，从而减少对这些特征的误认，提高了模型对恶意代码的敏感性和区分能力，进而提高了分类的准确性和稳定性。

其次，通过比较DenseNet121_CBAM1和DenseNet121_CBAM3模型可以发现，并非在DenseNet121模型的每个位置都添加CBAM能够获得更好的效果。这是因为恶意代码的特征具有多样性和复杂性，某些位置的特征对分类任务的重要性可能较低，而在这些位置添加CBAM并不能产生明显的改进效果。因此，仅在关键位置添加CBAM可能更为有效，有助于提高模型对恶意代码中关键特征的识别和利用能力。在设计和优化深度学习模型时，需要仔细考虑CBAM的添加位置和数量。对于不同的任务和数据集，关键特征的分布和影响程度可能存在差异，因此需要结合实际情况选择合适的CBAM位置和数量。这样能够充分利用模型的学习能力和特征提取能力，最大限度地提升模型在恶意代码分类任务中的性能和效果。

接着，对恶意代码家族分类和类型分类2个任务分析发现，相较于添加了CBAM的DenseNet121_CBAM1模型，未添加CBAM的DenseNet121模型准确率分别下降了0.06%和1.18%；添加了SE注意力机制的DenseNet121_SE模型准确率分别下降了0.18%和1.18%。这些实验结果强调了CBAM的存在对提升模型性能的关键作用，并进一步说明了选择合适的注意力机制类型在优化模型性能方面的重要性。

此外，在探究模型深度相关实验中，在恶意代码家族分类和类型分类2个任务上，DenseNet121_CBAM1相比DenseNet88_CBAM的准确率分别低了0.19%和1.64%。此外，DenseNet121_CBAM1的训练时间相对于DenseNet88_CBAM来说更长，在恶意代码家族和类型分类任务上，一轮训练时间分别增加了9 s和4 s。因此，本文认为网络层数较浅的DenseNet88_CBAM更适合用于恶意代码分类任务，因为其具有更高的分类准确率和更快的训练速度。

基于以上分析，本文认为DenseNet88_CBAM在恶意代码分类任务中更加适用，因为它具有更高的分类准确率，尤其是在恶意代码类型分类方面，它能表现出更出色的分类效果。该模型不仅拥有较浅的网络层数，同时选择了合适的注意力机制位置和类型，通过注意力机制、密集连接和过渡层的巧妙设计，实现了高效的特征提取和准确分类。这些优化策略在实验中得到了验证，并显著提升了深度

表7 注意力机制添加细节对分类准确率的影响

模型	Big2015数据集		PE数据集	
	分类准确率	训练一轮所需时间/s	分类准确率	训练一轮所需时间/s
DenseNet121	99.09%	682	80.63%	137
DenseNet121_CBAM1	99.15%	688	81.81%	139
DenseNet121_CBAM2	99.06%	682	79.89%	138
DenseNet121_CBAM3	98.78%	698	79.52%	145
DenseNet121_SE	98.97%	685	80.63%	135
DenseNet88_CBAM	99.34%	679	83.45%	135

学习网络的性能。

3.2.2　对比实验

为了验证本文提出的恶意代码分类方法在效果提升方面的有效性，本节将其与基线方法进行比较。

1) 基线方法

Kalash 等[20]利用 Nataraj 矢量化方法将恶意代码的二进制文件转换为灰度图，并使用基于 VGG16 改进的卷积神经网络模型进行分类。王博等[39]提出了一种创新的方法，该方法将每个二进制比特串切割成长度为 8 bit 的子串，并将每连续 3 个子串分别对应 RGB 通道。通过基于 VGG16 改进的卷积神经网络模型提取特征，实现对恶意代码的分类。针对文献[39]中存在的模型数量过多的问题，蒋考林等[40]提出了一种与其类似的恶意代码可视化为彩色图的方法。然而，其独特之处在于在末尾不足的情况下使用 0 进行填充，并使用 AlexNet 进行训练和分类。此外，为了比较深度学习与机器学习的性能，本文提取灰度共生矩阵[41]，并将其角二阶矩、对比度、熵以及反差分矩阵作为机器学习特征，随后应用 KNN 分类器进行分类。以上这 4 个特征为恶意代码提供了一种量化描述方式，有效地揭示了恶意代码在纹理、结构和行为特性方面的内在属性。值得注意的是，这 4 个特征在灰度共生矩阵特征提取的研究领域中具有广泛的应用和认可，如文献[17]也使用了这 4 个特征。具体来说，角二阶矩可以帮助评估图像的整体复杂度和纹理结构。对比度能提供关于代码纹理和模式变化的信息，从而有助于区分不同类型或变种的恶意代码。熵是描述系统随机性或不确定性的度量，能揭示代码的复杂性和随机性。反差分矩阵衡量了图像中不同像素值对之间的平均差异，可以捕获代码的局部结构和纹理变化，为恶意代码的分类和识别提供重要线索。借助这些特征，研究者能够更为深入和全面地探索恶意代码的独特属性和行为模式。上述 4 种方法都被视为本文的基线方法。

2) 实验结果与分析

通过对比实验，对本文提出的恶意代码分类方法的有效性进行了验证，实验结果如表 8 所示，其中，—表示没有此项内容。

对表 8 的实验结果进行分析可以发现，本文方法在恶意代码家族分类任务和恶意代码类型分类任务上的表现均优于其他对比方法。

与文献[20,39-40]这 3 种深度学习方法相比，本文方法在恶意代码分类方面表现优异，具有最高的分类准确率，这使得其具有广泛的应用价值和推广意义。相较于对比方法中使用的 VGGNet 和 AlexNet 网络，本文所采用的 DenseNet88_CBAM 神经网络模型结构简单，在实现准确分类的同时也减少了计算量和参数量的大小，在实际应用中具有更好的可扩展性和实用性。此外，本文设计的可视化方法结合了 N-gram 方法和 TF-IDF 方法，能够充分考虑恶意代码中关键词的组合及其出现频率，从而提高了恶意代码分类模型的性能。

相较于 GLCM 结合 KNN 的机器学习方法，本文方法在恶意代码家族和类型分类任务中均实现了显著的分类准确率提升，在恶意代码家族分类方面提升达到了 11.27%，在恶意代码类型分类方面提升高达 29.14%。尽管机器学习方法具有简单易懂、容易实现的优点，但因为需要人工选择特征和分类器，可能无法发现数据的潜在特征，而且会消耗大量的时间。而深度学习方法不需要手动进行特征提取和选择，而是通过模型自动提取特征，更加便捷，从而可以更好地适应不同的数据分布和任务。因此，本文方法能够更加准确地学习到恶意代码图

表 8　　**相关研究工作比较**

方法	Big2015 数据集				PE 数据集			
	分类准确率	训练一轮所需时间/s	FLOPS/GB	参数量/MB	分类准确率	训练一轮所需时间/s	FLOPS/GB	参数量/MB
文献[20]	97.84%	1 042	15.506 8	134.305 9	73.99%	171	15.506 8	134.293 6
文献[39]	98.03%	691	15.506 8	134.305 9	74.17%	130	15.506 8	134.293 6
文献[40]	98.23%	594	4.790 2	169.002 7	71.70%	111	4.790 2	168.996 6
GLCM+KNN	88.07%	—	—	—	54.31%	—	—	—
本文方法	99.34%	679	2.661 6	4.409 9	83.45%	135	2.661 6	4.406 9

像中的特征，具有更好的泛化能力和分类准确率。为了进一步评估本文方法的优越性，收集了12个最新的恶意代码样本用于实验，其中，收集的样本名称及其对应的sha256哈希值如表9所示。在实验中，将本文方法与已有对比方法对样本的检测类型进行了比较，实验结果如表10所示。通过对这些实验结果进行对比和分析，能够更全面地评估本文方法在恶意代码分类任务中的性能和效果。同时，这些实验样本的收集也增强了本文对恶意代码分类问题的理解，并为后续的研究提供了更准确和可靠的基准。通过在实验中充分考虑对比方法的性能和本文方法的优势，证明了本文方法的有效性和适用性，为恶意代码分类领域的研究和实践提供了有力的支持。

观察表10中的实验结果不难发现，在本文收集的12个最新的恶意代码样本中，只有本文方法能够较为准确地检测出这些恶意代码样本的类型（12个样本可以准确检测出9个），而对比方法均无法达到相应的检测效果，这一发现进一步验证了本文方法在恶意代码分类任务中的优越性和独特性。产生这一结果的原因是对比方法所采用的可视化方法提取的特征不够明显和有力。尽管对比方法在一定程度上能够捕获一些恶意代码样本的特征信息，然而，这些特征在可视化过程中无法充分表达其恶意属性，导致对比方法在恶意代码检测中存在明显的局限性。相反，本文方法通过充分融合N-gram和TF-IDF的特征表示能力，能够更好地捕捉恶意代码样本中的关键特征组合和频率信息，从而显著提升了检测的准确性和可靠性。

表9 收集的样本名称及其对应sha256哈希值

样本名称	对应sha256哈希值
JmNeK	076cb1ac8e46bc1226a8bb42d83afac656d525cb7e6dc9a4d79475ab9b286440
AlBy.exe	6b64730d26c6e0c4ec3db4e89ac886fbaf8cd4decf695c44201a8dc45b3d9f5a
WEXTRACT.EXE.MUI	6dd1a8408e598604e40099415b555bb490cef19c9c096944f693bc0bea46a099
7a598f85-f7a0-42aa-97e5-bf517a969646.exe	900dfd325ef667b5ac55768bbc18db18e9dcfd309fc07b37c8c83796f7fd9ac5
JlIH.exe	d0dee99d6879a777938604421ce10c42a0fba9420f8fe7a77f8a4875a869208e
hvNP.exe	d6ffff9bb266b05c6dfd2de91bcd38df25bb27e21e2c3626d03f682aa15ed3df
JmNeK	d8c9255982a5932dbaf224d475d2161d814de36784b797d576e41c263587e20a
CZhU.exe	e165bef062d93e5a4fd31d8f369fd8144d2158b5c8dcb85be3e06826cb5f81e6
ubCB.exe	e58e843ff95f4ca52f2e4da56fdb3edb62fe89415d53b30c0b187f12e9e644aa
Mpgzljfv.exe	ebc89503b1729887f5ea5423b334ef1b3ec215386b8c9c656a177f338158d4d1
Assad Luminescing.exe	1c809aae258aaa9f029a80ed7b754eead202037eaa84b95c1ee9df2e49faf927
PoRZQ	12dfb5124ecd3035e6263de472ca980bc47bd9e5574a6c6677da68a662dfb957

表10 针对新收集样本相关研究工作比较

方法	JmNeK	AlBy.exe	WEXTRACT.EXE.MUI	7a598f85-f7a0-42aa-97e5-bf517a969646.exe	JlIH.exe	hvNP.exe	JmNeK	CZhU.exe	ubCB.exe	Mpgzljfv.exe	Assad Luminescing.exe	PoRZQ	正确检测数量/个
VT检测	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	12
文献[20]	Trojan	Trojan	Variant	Backdoor	Trojan	Trojan	Trojan	Trojan	Variant	Trojan	Variant	Backdoor	7
文献[39]	Trojan	Backdoor	Variant	Backdoor	Variant	Trojan	Trojan	Variant	Backdoor	Trojan	Variant	Trojan	5
文献[40]	Variant	Trojan	Variant	Generic	Trojan	Trojan	Trojan	Backdoor	Backdoor	Backdoor	Variant	Variant	4
GLCM+KNN	Variant	Worm	Trojan	Trojan	Worm	Worm	Variant	Variant	Trojan	Trojan	Trojan	Trojan	6
本文方法	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Trojan	Backdoor	Backdoor	Backdoor	9

综上所述，本文方法在一个广泛使用的恶意代码家族数据集、一个非公开的恶意代码类型数据集以及新收集的十余种恶意代码样本上进行了全面而严格的实验评估，旨在验证在不同样本集上的性能表现。这些数据集涵盖了多个恶意代码家族和类型，并包含大量的恶意样本。通过实验分析可知，本文方法取得了优异的效果，在恶意代码家族分类和恶意代码类型分类2个任务的研究上，本文方法在分类准确率方面均显著优于机器学习方法以及深度学习方法，证明了其在恶意代码分类任务中的可行性。这些实验结果为本文方法提供了有力的支持，并为进一步研究和应用提供了有益的参考。

4　结束语

本文提出了一种基于Ngram-TFIDF的深度恶意代码可视化分类方法。该方法通过Ngram-TFIDF将恶意代码文件转化为灰度图，并利用像素值乘积进行图像增强。最终，利用DenseNet88_CBAM模型进行训练，实现了恶意代码的分类。实验结果表明，本文方法在恶意代码家族分类任务和恶意代码类型分类任务上均表现优越，分类准确率高于对比方法，具有广泛的适用性和可靠性。

尽管本文方法取得了较好的效果，但仍存在一些不足。首先，目前的研究集中在灰度图的可视化，未来可以进一步考虑彩色图的应用，以更全面地反映恶意代码特性。其次，非PE文件的特征提取和可视化方法需要进一步深入研究。最后，结合对抗生成网络和图卷积网络的应用在恶意代码检测领域还相对较少，未来可以探索这些方法与可视化技术的结合，以提高分类和检测效果。

参考文献：

[1] IWAMOTO K, WASAKI K. Malware classification based on extracted API sequences using static analysis[C]//Proceedings of the 8th Asian Internet Engineering Conference. New York: ACM Press, 2012: 31-38.

[2] IMRAN M, AFZAL M T, QADIR M A. Similarity-based malware classification using hidden Markov model[C]//Proceedings of the 2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec). Piscataway: IEEE Press, 2015: 129-134.

[3] HARDY W, CHEN L W, HOU S F, et al. DL4MD: A deep learning framework for intelligent malware detection[C]//Proceedings of the International Conference on Data Mining (ICDATA). Piscataway: IEEE Press, 2016: 61-67.

[4] SCHULTZ M G, ESKIN E, ZADOK F, et al. Data mining methods for detection of new malicious executables[C]//Proceedings of the 2001 IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2001: 38-49.

[5] KOLTER J Z, MALOOF M A. Learning to detect malicious executables in the wild[C]//Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM Press, 2004: 470-478.

[6] KOLTER J Z, MALOOF M A. Learning to detect and classify malicious executables in the wild[J]. Journal of Machine Learning Research, 2006, 6: 2721-2744.

[7] KANG B, YERIMA S Y, MCLAUGHLIN K, et al. N-opcode analysis for android malware classification and categorization[C]//Proceedings of the 2016 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). Piscataway: IEEE Press, 2016: 1-7.

[8] KONG D G, YAN G H. Discriminant malware distance learning on structural information for automated malware classification[C]//Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM Press, 2013: 1357-1365.

[9] LI B, ROUNDY K, GATES C, et al. Large-scale identification of malicious singleton files[C]//Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. New York: ACM Press, 2017: 227-238.

[10] KUMAR A, KUPPUSAMY K S, AGHILA G. A learning model to detect maliciousness of portable executable using integrated feature set[J]. Journal of King Saud University - Computer and Information Sciences, 2019, 31(2): 252-265.

[11] FIRDAUSI I, LIM C, ERWIN A, et al. Analysis of machine learning techniques used in behavior-based malware detection[C]//Proceedings of the 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies. Piscataway: IEEE Press, 2010: 201-203.

[12] ZOLKIPLI M F, JANTAN A. An approach for malware behavior identification and classification[C]//Proceedings of the 2011 3rd International Conference on Computer Research and Development. Piscataway: IEEE Press, 2011: 191-194.

[13] Microsoft malware classification challenge (big 2015)[R]. 2017.

[14] NATARAJ L, KARTHIKEYAN S, JACOB G, et al. Malware images: visualization and automatic classification[C]//Proceedings of the 8th International Symposium on Visualization for Cyber Security. New York: ACM Press, 2011: 1-7.

[15] NATARAJ L, YEGNESWARAN V, PORRAS P, et al. A comparative assessment of malware classification using binary texture analysis and dynamic analysis[C]//Proceedings of the 4th ACM workshop on Security and artificial intelligence. New York: ACM Press, 2011: 21-30.

[16] LIU L, WANG B S, YU B, et al. Automatic malware classification and new malware detection using machine learning[J]. Frontiers of Information Technology & Electronic Engineering, 2017, 18(9): 1336-1347.

[17] FU J W, XUE J F, WANG Y, et al. Malware visualization for fine-grained classification[J]. IEEE Access, 2018, 6: 14510-14523.

[18] 刘亚姝, 王志海, 严寒冰, 等. 抗混淆的恶意代码图像纹理特征描述方法[J]. 通信学报, 2018, 39(11): 44-53.
LIU Y S, WANG Z H, YAN H B, et al. Method of anti-confusion texture feature descriptor for malware images[J]. Journal on Communications, 2018, 39(11): 44-53.

[19] 郎大鹏, 丁巍, 姜昊辰, 等. 基于多特征融合的恶意代码分类算法[J]. 计算机应用, 2019, 39(8): 2333-2338.
LANG D P, DING W, JIANG H C, et al. Malicious code classification algorithm based on multi-feature fusion[J]. Journal of Computer Applications, 2019, 39(8): 2333-2338.

[20] KALASH M, ROCHAN M, MOHAMMED N, et al. Malware classification with deep convolutional neural networks[C]//Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Piscataway: IEEE Press, 2018: 1-5.

[21] VASAN D, ALAZAB M, WASSAN S, et al. IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture[J]. Computer Networks, 2020, 171: 107138.

[22] 王润正, 高见, 仝鑫, 等. 融合注意力机制的恶意代码家族分类研究[J]. 计算机科学与探索, 2021, 15(5): 881-892.
WANG R Z, GAO J, TONG X, et al. Research on malicious code family classification combining attention mechanism[J]. Journal of Frontiers of Computer Science and Technology, 2021, 15(5): 881-892.

[23] PINHERO A, M L A, VINOD P, et al. Malware detection employed by visualization and deep neural network[J]. Computers & Security, 2021, 105: 102247.

[24] ANANDHI V, VINOD P, MENON V G. Malware visualization and detection using DenseNets[J]. Personal and Ubiquitous Computing, 2024, 28(1): 153-169.

[25] HUANG X, MA L, YANG W Y, et al. A method for windows malware detection based on deep learning[J]. Journal of Signal Processing Systems, 2021, 93(2): 265-273.

[26] MOUSSAS V, ANDREATOS A. Malware detection based on code visualization and two-level classification[J]. Information, 2021, 12(3): 118.

[27] DAREM A, ABAWAJY J, MAKKAR A, et al. Visualization and deep-learning-based malware variant detection using OpCode-level features[J]. Future Generation Computer Systems, 2021, 125: 314-323.

[28] CONTI M, KHANDHAR S, VINOD P. A few-shot malware classification approach for unknown family recognition using malware feature visualization[J]. Computers & Security, 2022, 122: 102887.

[29] FALANA O J, SODIYA A S, ONASHOGA S A, et al. Mal-Detect: an intelligent visualization approach for malware detection[J]. Journal of King Saud University - Computer and Information Sciences, 2022, 34(5): 1968-1983.

[30] CHAGANTI R, RAVI V, PHAM T D. Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification[J]. Journal of Information Security and Applications, 2022, 69: 103306.

[31] MALLIK A, KHETARPAL A, KUMAR S. ConRec: malware classification using convolutional recurrence[J]. Journal of Computer Virology and Hacking Techniques, 2022, 18(4): 297-313.

[32] CHAUHAN D, SINGH H, HOODA H, et al. Classification of malware using visualization techniques[C]//International Conference on Innovative Computing and Communications. Berlin: Springer, 2022: 739-750.

[33] SPÄRCK JONES K. A statistical interpretation of term specificity and its application in retrieval[J]. Journal of Documentation, 2004, 60(5): 493-502.

[34] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2017: 2261-2269.

[35] WOO S, PARK J, LEE J Y, et al. CBAM: convolutional block attention module[C]//European Conference on Computer Vision. Berlin: Springer, 2018: 3-19.

[36] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.

[37] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//International Conference on Learning Representations. Berlin: Springer, 2015: 1-14.

[38] GAO S H, CHENG M M, ZHAO K, et al. Res2Net: a new multi-scale backbone architecture[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(2): 652-662.

[39] 王博, 蔡弘昊, 苏旸. 基于VGGNet的恶意代码变种分类[J]. 计算机应用, 2020, 40(1): 162-167.
WANG B, CAI H H, SU Y. Classification of malicious code variants based on VGGNet[J]. Journal of Computer Applications, 2020, 40(1): 162-167.

[40] 蒋考林, 白玮, 张磊, 等. 基于多通道图像深度学习的恶意代码检测[J]. 计算机应用, 2021, 41(4): 1142-1147.
JIANG K L, BAI W, ZHANG L, et al. Malicious code detection based

on multi-channel image deep learning[J]. Journal of Computer Applications, 2021, 41(4): 1142-1147.

[41] HARALICK R M, SHANMUGAM K, DINSTEIN I. Textural features for image classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1973, 3(6): 610-621.

[作者简介]

王金伟（1978– ），男，内蒙古呼伦贝尔人，博士，南京信息工程大学教授，主要研究方向为多媒体版权保护、多媒体取证、多媒体加密和数据认证。

陈正嘉（1999– ），男，江苏徐州人，南京信息工程大学硕士生，主要研究方向为网络安全、信息安全、恶意软件检测。

谢雪（1989– ），男，吉林长春人，中国科学技术大学博士生，主要研究方向为网络安全、多媒体取证。

罗向阳（1978– ），男，湖北荆门人，信息工程大学教授，主要研究方向为图像隐写和隐写分析技术。

马宾（1973– ），男，山东济宁人，齐鲁工业大学教授，主要研究方向为可逆信息隐藏、多媒体取证、隐写与隐写分析。