TriCh-LKRepNet:融合三通道映射与结构重参数化的大核卷积恶意代码分类网络

李思聪[1],王 坚[1],宋亚飞[1*],王 硕[2]
(1. 空军工程大学防空反导学院,陕西西安 710051;2. 中国人民解放军95285部队,广西桂林 541000)

摘 要: 随着网络威胁的日益严峻,恶意代码的检测与分类变得尤为关键. 传统分析方法依赖手动特征提取,不仅耗时且难以跟上恶意代码的快速变异. 相比之下,深度学习技术在恶意代码分类方面展现出巨大潜力. 然而,模型复杂度和资源消耗仍是实际部署的难题. 本研究提出了TriCh-LKRepNet(Triple-Channel Large Kernel Reparameterisation Network),该网络专注于轻量化设计,旨在确保检测性能的同时降低计算和内存需求. 通过提出的三通道映射技术,将恶意代码的多维信息有效转换为图像通道,增强了特征的区分性. 结合卷积神经网络(Convolutional Neural Networks,CNN)和Transformer的优势,设计了一个高效的深度学习架构,并通过重参数化技术优化了连接路径,以降低内存消耗并提升运行效率. 此外,引入的线性训练时间过参数化和大卷积核技术进一步降低了模型的参数量和计算负担. 通过实验证明,TriCh-LKRepNet在提升恶意代码分类精度的同时实现了模型的轻量化,与现有技术相比,展现出更佳的性能和更广泛的应用潜力,特别是在资源受限和需要实时检测的环境中,提供了一种有效的解决方案.

关键词: 恶意代码分类;恶意代码可视化;结构重参数化;大卷积核;汇编信息;语义关系
基金项目: 国家自然科学基金(No.61806219,No.61703426,No.61876189);陕西省自然科学基金(No.2021JM226);陕西省高校科协青年人才托举计划(No.20190108,No.20220106);陕西省创新能力支撑计划(No.2020KJXX-065)
中图分类号: TP309.5 **文献标识码:** A **文章编号:** 0372-2112(2024)07-2331-10
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20240162

TriCh-LKRepNet: A Large Kernel Convolutional Malicious Code Classification Network for Structure Reparameterisation and Triple-Channel Mapping

LI Si-cong[1], WANG Jian[1], SONG Ya-fei[1*], WANG Shuo[2]
(1. *Air and Missile Defense College, Air Force Engineering University, Xi'an, Shaanxi* 710051, *China*;
2. *Unit of* 95285 *of the PLA, Guilin, Guangxi* 541000, *China*)

Abstract: With the increasing severity of cyber threats, the detection and classification of malicious code has become particularly critical. Traditional analysis methods rely on manual feature extraction, which is time-consuming and difficult to keep up with the rapid mutation of malicious code. In contrast, deep learning techniques show great potential for malicious code classification. However, model complexity and resource consumption are still challenges for practical deployment. In this study, we propose the TriCh-LKRepNet (Triple-Channel Large Kernel Reparameterisation Network), which focuses on lightweight design and aims to ensure detection performance while reducing computation and memory requirements. Through the proposed three-channel mapping technique, the multi-dimensional information of malicious code is effectively converted into image channels, which enhances the differentiation of features. An efficient deep learning architecture is designed by combining the advantages of convolutional neural networks (CNN) and Transformer, and the connection paths are optimized by a reparameterization technique to reduce the memory consumption and enhance the operation efficiency. In addition, the introduced linear training time over-parameterization and large convolutional kernel techniques further reduce the number of parameters and computational burden of the model. It is experimentally demonstrated that TriCh-LKRepNet improves the malicious code classification accuracy while realizing the model's lightweight, which shows better

收稿日期:2024-02-21;修回日期:2024-04-16;责任编辑:覃怀银
*通讯作者:宋亚飞

performance and wider application potential than existing techniques, especially in resource-constrained environments where real-time detection is required, providing an effective solution.

Key words: malicious code classification; malicious code visualization; structural reparameterisation; large convolutional kernel; assembly information; semantic relations

Foundation Item(s): National Natural Science Foundation of China (No.61806219, No.61703426, No.61876189); Natural Science Foundation of Shaanxi Province (No.2021JM226); Young Talent Fund of Association for Science and Technology in Shaanxi (No.20190108, No.20220106); Innovation Capability Support Program of Shaanxi (No.2020KJXX-065)

1 引言

随着网络技术的迅猛发展和数字化进程的加速,网络安全威胁日益升级,恶意代码作为其中的一大难题,已经成为全球范围内广泛关注的焦点. 根据2023年7月14日AV-TEST[1]发布的最新统计报告,全球范围内每天新检测到的恶意软件数量高达560 000个. 它揭示了恶意软件传播的惊人速度和广泛范围. 截至目前,已知的恶意程序总数已超过1亿个,其中新发现的恶意程序实例超过17万个. 这些数字不仅反映了恶意软件问题的严重性,也凸显了传统安全防护手段在应对这一挑战时的力不从心. 恶意代码不仅种类繁多,变种层出不穷,而且传播速度快,破坏力巨大,对个人隐私、企业数据乃至国家安全都构成了严重威胁. 因此,恶意代码的检测和分类成为了网络安全领域的重要研究课题.

传统的恶意代码分析方法多依赖于手动提取的特征,这些特征取自恶意代码的静态或动态行为,例如文件构造、代码段落或函数调用等. 但这种方法既耗时又耗力,要求操作者具有丰富的专业知识和实践经验. 此外,随着恶意代码日趋复杂,手动提取的特征难以全面描述其特性,进而影响检测效果. 虽然静态分析技术在识别已知恶意代码方面有一定成效,但对于未知或轻微变异的恶意代码,其检测能力则大打折扣. 多项研究[2~7]虽然提出了不同的检测框架和方法,但均受限于静态分析的固有弱点. 动态分析技术则是通过在安全环境中实际执行代码来观察其行为,从而能够捕捉到静态分析可能忽略的运行时行为,如网络活动、文件操作和注册表更改等. 这为检测和分类恶意代码提供了更多线索. 在这一领域中,文件熵作为一个关键特征被广泛使用[8~10]. 然而,动态分析可能带来额外的性能开销和安全风险.

近年来,深度学习技术以其强大的自动特征提取和分类能力,在图像识别、语音识别、自然语言处理等领域取得了显著成果. 在恶意代码检测方面,研究人员尝试将恶意代码特征转化为图像数据,然后应用传统的机器学习算法进行分类. 尽管如此,由于恶意代码图像与自然图像在语义上的显著差异,这种方法在实际应用中的表现往往不尽如人意[11~16]. 深度学习作为机器学习的一个子领域,通过构建多层神经网络来自动学习和提取数据特征. 在恶意代码检测中,一些研究者将恶意代码特征转化为图像,并利用卷积神经网络(Convolutional Neural Networks,CNN)进行分类和识别. 例如,Nataraj等人[17]提出了一种将恶意软件可视化为图像并自动分类的方法. 其他研究[18~21]也采用了类似的图像转换和CNN分类方法. 此外,还有研究探索了多通道可视化与CNN结合[22]以及利用控制流图(Control Flow Graph,CFG)进行深度学习模型嵌入和分类的方法[23~26]. 这些方法共同推动了恶意代码检测领域的发展.

尽管上述方法在恶意代码检测领域取得了一定的进展,但仍然面临着许多挑战和限制. 例如,手动特征提取的耗时耗力、动态分析可能带来的性能开销和安全风险、深度学习模型的复杂性和训练数据的需求等. 为应对以上问题,本文提出了一种名为LKRepNet(Large Kernel Reparameterisation Network)的恶意代码分类网络,通过三通道映射技术将恶意代码的三种表示形式转为RGB图像,并结合CNNs与Transformer设计了一个深度神经网络框架,引入网络结构重参数化、线性训练时间过参数化和大内核卷积技术,进一步提升检测的准确性和效率,同时降低计算复杂度和内存需求. 该方法在Kaggle和DataCon数据集上验证有效,展现出高准确率、强泛化能力和抗代码混淆能力.

2 模型概述

本章研究了基于深度学习的恶意代码检测方法,并提出了一个包含两个核心部分的新模型架构:恶意代码RGB映射可视化和LKRepNet模型.

在预处理阶段,通过结合二进制文件特征、汇编指令和API聚类特征,生成信息丰富的RGB图像,为模型训练提供高质量数据. 在LKRepNet模型构建方面,将CNN与Transformer的融合,采用新的Token混合算子进行结构优化,减少内存访问成本,并利用过参数训练和大核卷积技术提升精度. 模型通过调整优化器和损失函数快速收敛,实现高效检测. 整体模型结构如图1所示.

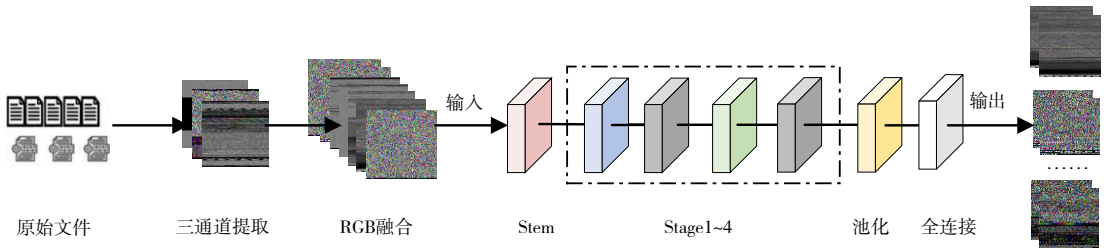

图1　TriCh-LKRepNet模型总体结构

2.1　数据预处理

在恶意代码分析中,特征预处理是关键步骤,涉及从原始数据提取有用信息并转换为分析可用形式. 本文侧重于处理Kaggle和DataCon数据集中的API调用和操作码操作数等关键信息. 对于Kaggle数据集,采取从反汇编文件中提取关键信息,而DataCon数据集则使用IDA Pro工具进行深入分析. 为确保特征完整性,本文采用三通道映射技术,如图2所示,本文将恶意代码二进制文件特征、汇编指令和API信息整合,构建全面的特征矩阵. 该技术按虚拟地址顺序填充特征矩阵,将恶意代码特征转换为便于理解和机器处理的RGB图像[27].

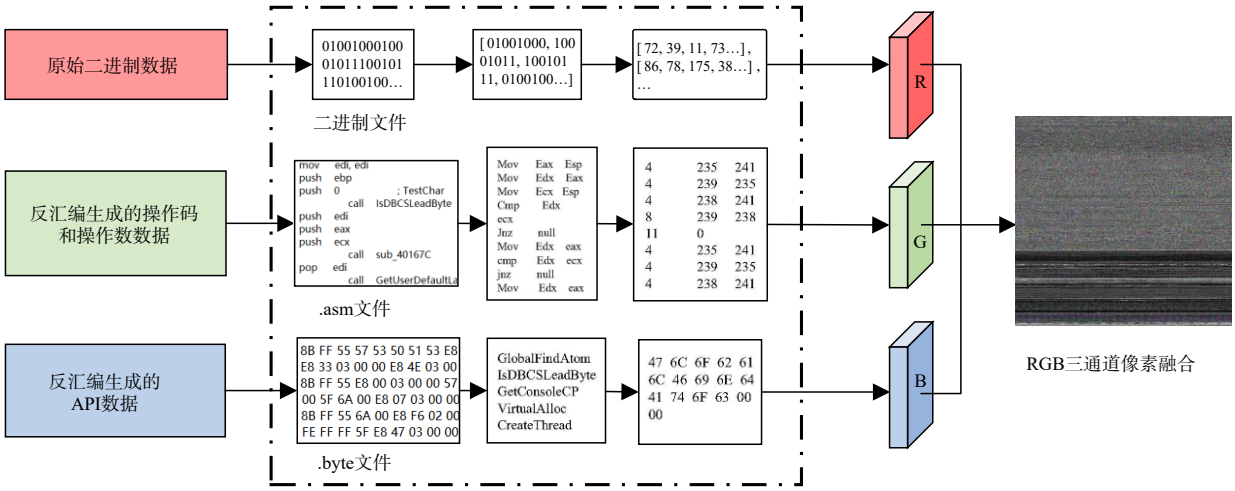

图2　恶意代码转换为RGB图像的流程

2.1.1　二进制恶意代码文件可视化

R通道采用直接保留二进制代码转换的方式. 首先,将给定的恶意代码二进制文件以每8位无符号整数为一组进行读取. 然后,将每组二进制数转化为10进制整形. 其次,根据PE文件大小确定行宽,并将其转换为二维数组,生成的灰度图像的大小,取决于虚拟地址数量,利用虚拟地址决定文件的长度,宽度利用文件大小进行计算,保证RGB三部分通过虚拟地址进行一一对应. 其行宽与文件大小对应关系如表1所示. 最后,以二维数组中每一个元素作为图像的灰度值,将二维数组映射为灰度图像.

表1　行宽与恶意文件大小对应关系

文件大小/KB	宽度	文件大小/KB	宽度
<10	32	100~200	384
10~30	64	200~500	512
30~60	128	500~1 000	768
60~100	256	>1 000	1 024

2.1.2　汇编指令和数据可视化

汇编指令和数据在程序开发和编译过程中受到多种因素影响,导致同族恶意代码的基本块可能不具有完全相同的操作码序列. 在处理汇编信息时,我们重点关注操作码和操作数两部分. 操作码反映程序行为,通常在同家族恶意软件中表现出相似性,操作数则根据寄存器类型标记,并在相应位置提取后用0填充剩余位置.

针对Kaggle数据集,考虑到235种常用操作码占据了绝大多数比例,本文选择仅使用这些操作码进行分析,并将剩余少见操作码归类到统一类型. 通过合并具有相同功能的操作码,如MOV、MOVQ和MOVD,我们有效地将操作码数量缩减至235个,并为它们分配了编码. DataCon数据集也经过类似的处理流程. 最终,由于编码范围(1~235)适合灰度级像素值(0~255),操作码和操作数可以直接转换为单通道图像G的像素值,从而将汇编信息转化为图像形式,便于后续分析. 表2展示了

提取的操作符、操作数、虚拟地址及其编码结果.

表 2 Opcode 编码实例

虚拟地址	10001000	10001001	10001002	10001003
操作码操作数	Mov	Eax	Esp	Null
编码	4	235	0	0
虚拟地址	10001004	10001005	10001006	10001007
操作码操作数	Mov	Eix	Eax	Mov
编码	4	0	235	4
虚拟地址	10001008	10001009	1000100A	1000100B
操作码操作数	Edx	Eax	Cmp	Eax
编码	239	235	8	235
虚拟地址	1000100C	1000100D	1000100E	1000100F
操作码操作数	Jnz	Edx	Mov	Edx
编码	11	239	4	239

2.1.3 API 可视化技术

基于上下文相似性,本文将相关的 API 聚类并分组到统一的聚类名称下. 通过将多个 API 有效地分组到有限数量的动态库中,我们能够描述恶意代码的行为属性. 然而,分类技术往往依赖于 API 序列模式,这使得插入干扰信息成为一种可能误导静态分析的手段.

从 Kaggle 和 DataCon 数据集中提取 API 和动态库信息,生成完整的 API 调用序列. API 信息通常由 ASCII 字符表示,其值范围在 32~126 之间. 为避免噪声干扰,API 信息被编码为单通道图像 B 的像素值,编码范围设计得小于灰度级像素值的上限. 这种编码方法简化了数据处理并提高了分析的准确性,具体的编码对应关系在表 3 中展示.

表 3 API 编码图解

虚拟地址	API 接口	十六进制编码
00402000	GlobalFindAtom	47 6C 6F 62 61 6C 46 69 6E 64 41 74 6F 63 00 00
00402004	IsDBCSLeadByte	6C 74 44 42 43 53 4C 65 61 64 42 79 74 65 00 00
00402008	GetConsoleCP	47 65 74 43 6F 42 74 6C 65 43 50 00 00 00 00 00
0040200C	VirtualAlloc	56 69 72 74 75 61 6C 41 6C 6C 6F 63 00 00 00 00
00402010	CreateThread	43 72 65 61 74 65 54 68 72 65 61 64 00 00 00 00

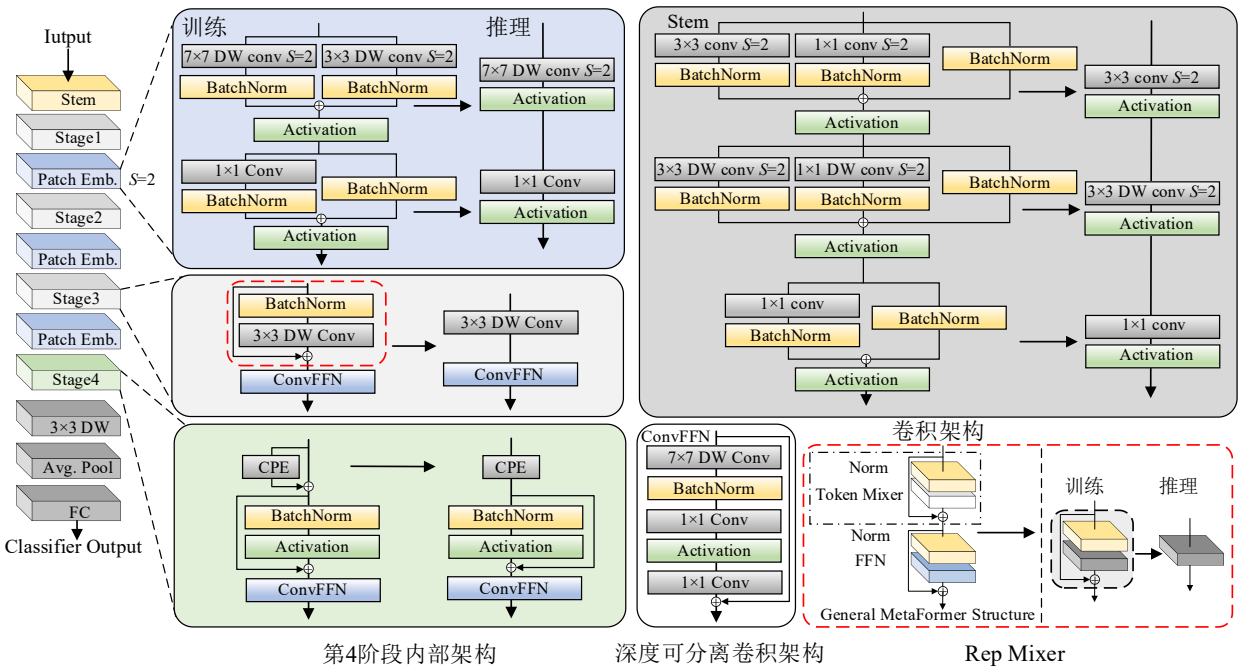

图 3 LKRepNet 特征提取架构

2.2 LKRepNet 模型构建

LKRepNet 模型融合了 CNN 和 Transformer 的优势,旨在提高恶意代码检测的性能. CNN 负责提取局部特征,而 Transformer 通过自注意力机制捕捉全局依赖,两者结合使得模型能够更全面地理解图像内容[28]. 模型的设计考虑了特征提取的互补性和计算效率的平衡,以优化特定任务的性能.

模型整体架构如图 3 所示,采用多阶段架构,每个阶段特征图分辨率减半通道数加倍,通过 Rep Mixer 捕获全局上下文信息. 第四阶段引入 Attention 机制作为

token mixer,提高分类精度. 此外,模型使用了改进的ConvFFN架构,通过大核7×7深度可分离卷积层捕获空间上下文信息,同时提高运行效率.

2.2.1　结构重参数化

多分支网络结构通过捕获输入数据的多样化特征来增强深度学习模型的表达能力[29,30]. 尽管增加分支可以提升性能,但也会导致训练时内存消耗和推理时计算速度的增加. 为此,本文采用了结构重参数化策略,允许网络在训练阶段利用多分支的优势,而在推理阶段转换为单分支结构,以提高运行效率并降低内存需求. 训练时的多分支结构通过一系列融合步骤转化为推理时的单分支结构. 这些步骤包括合并批量归一化(Batch Normalization,BN)层与卷积层,以及整合不同尺寸的卷积层,最终实现在推理阶段的高效性能.

BN层与卷积层的融合. 设卷积变化、BN操作分别表示为

$$y_{\text{conv}}=\omega\cdot x+b \tag{1}$$

$$\text{BN}_{\gamma,\beta}(y_{\text{conv}})=\gamma\frac{y_{\text{conv}}-\mu}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{2}$$

其中,x为输入.

则输入x经过卷积层和BN层可表示为

$$\text{BN}_{\gamma,\beta}(x)=\frac{\gamma\omega}{\sqrt{\sigma^{2}+\varepsilon}}x+\frac{\gamma}{\sqrt{\sigma^{2}+\varepsilon}}(b-\mu)+\beta \tag{3}$$

其中,ω和b为融合前的权重和偏置;γ和β为经过训练得到的平移和缩放参数;μ和σ分别为全部训练数据的均值和方差;ε为一个非常小的常数,以避免除0错误. 令

$$\hat{\omega}=\frac{\gamma\omega}{\sqrt{\sigma^2+\varepsilon}} \tag{4}$$

$$\hat{b}=\frac{\gamma}{\sqrt{\sigma^2+\varepsilon}}(b-\mu)+\beta \tag{5}$$

由式(4)和式(5)得

$$\text{BN}_{\gamma,\beta}(x)=\hat{\omega}\cdot x+\hat{b} \tag{6}$$

其中,$\hat{\omega}$、$\hat{b}$分别为融合后卷积核的权重和偏置. 通过上述变换可以将卷积层与BN层进行融合,以减少推理时的计算量. 具体融合方法如下:

(1)1×1卷积与3×3卷积的融合. 将1×1卷积使用0填充至3×3大小,利用卷积的可加性,将填充得到的卷积与原3×3卷积相加,得到融合后的3×3卷积.

(2)3×3卷积与7×7卷积的融合. 与1×1卷积和3×3卷积的融合类似,将3×3卷积用0填充至大小为7×7的卷积,然后与原7×7卷积相加,得到融合后的7×7卷积.

(3)残差分支与BN层的融合. 残差分支可以视为原输入数据与有特殊参数的卷积核进行卷积操作. 使用1×1卷积,并且使当前通道数对应的卷积核参数为1,其余通道对应的参数为0,即可得到与输入一样的输出. 将此1×1卷积与BN层进行融合,然后使用0填充至5×5卷积.

至此,可以将训练时的多分支基本块等价转化为推理时的一个卷积.

2.2.2　线性时间过拟合

为了提升模型效率并降低运算量,研究者们常采用将$k\times k$卷积替换为其分解形式的策略,即先使用$k\times k$深度卷积再接1×1点卷积[31,32]. 这虽然减少了参数和计算复杂度,但也可能影响模型学习复杂模式的能力. 为了解决因卷积层分解导致的能力下降问题,LKRepNet模型采用了线性训练时间过参数化的技术[33]. 通过在训练过程中增加额外的参数,增强了模型的表示能力,使模型能够更好地拟合训练数据,从而提高其性能. 尽管这会增加计算开销,但由于参数的增加是线性的,训练时间的增长仍然控制在合理范围内.

此外,结构重参数化技术允许在训练时将复杂网络转换为更易优化的结构,并在推理时恢复为原始结构,以此保留性能优势[34]. LKRepNet利用这种技术优化卷积层,确保了计算效率和模型性能的平衡.

2.2.3　大核卷积

在对比Vision Transformer的Self-Attention与Rep Mixer时,考虑到Self-Attention提供全局感受野,能够同时处理所有位置的信息,但计算成本较高. 而Rep Mixer关注局部区域,计算效率更高,但可能限制长距离依赖的捕捉.

LKRepNet通过在FFN和Patch Embedding层引入深度大核卷积来扩大感受野,同时利用深度可分离卷积减少计算量和参数[35]. 这种卷积在不使用自注意力的情况下,增强了早期特征提取能力. 具体来说,模型在Patch Embedding后的patches上应用大核卷积,以及在FFN中用大核卷积替代传统卷积,以捕获更丰富的空间上下文信息.

这些改进让LKRepNet在低计算复杂度下,达到或超越基于自注意力的Vision Transformer的性能[36]. 表4展示了模型的参数设置,旨在平衡性能、计算效率和内存消耗. 通过这些优化,LKRepNet在保持效率的同时,实现了强大的性能表现.

LKRepNet的FFN块在设计上与传统ConvNet块类似,但关键的区别在于FFN块中使用了批标准化(BN)而非层标准化. 这种设计使得BN层可以在推理时与前一层融合,简化计算并提高效率. 此外,它避免了层标准化所需的额外重塑操作,减少了错误和性能瓶颈的风险.

相较于标准的vanilla-FFN块,采用大卷积核的Convolutional-FFN块展现出更强的鲁棒性. 大卷积核

表4 LKRepNet网络参数设置

Stage	#Tokens	Layer Spec.		LKRepNet
Stem	$H\times W$	Conv.		3×3,stride=2
				3×3,stride=2
				48
1	$\frac{H}{4}\times\frac{W}{4}$	Patch Embed.		7×7,stride=2
				48
		LKCN-Rep Block	Mixer	Rep Mixer
			Exp.	3
			Blocks	2
2	$\frac{H}{8}\times\frac{W}{8}$	Patch Embed.		7×7,stride=2
				96
		LKCN-Rep Block	Mixer	Rep Mixer
			Exp	3
			Blocks	2
3	$\frac{H}{16}\times\frac{W}{16}$	Patch Embed.		7×7,stride=2
				192
		LKCN-Rep Block	Mixer	Rep Mixer
			Exp.	3
			Blocks	4
4	$\frac{H}{32}\times\frac{W}{32}$	Patch Embed		7×7,stride=2
				384
		LKCN-Rep Block	Mixer	Rep Mixer
			Exp.	3
			Blocks	2

因其广阔的感受野,能够有效捕获输入数据的特征,增强了模型对输入变化的适应能力.

3 实验与分析

3.1 数据集与实验环境

实验使用了两个数据集:Kaggle和DataCon.Kaggle数据集[37]包含10 868个恶意样本,分为9个家族,数据包括.asm和.bytes格式的样本以及它们的哈希值和类别标签. 该数据集样本分布不均,如Simda家族仅42个样本,Kelihos ver3家族近3 000个样本. DataCon数据集[38]由奇安信公司提供,包含挖矿型和非挖矿型恶意代码样本,包含壳和资源混淆样本.

实验方法包括两种:一种是将Kaggle数据集的9个家族分别标记为1~9,记为9-class-data;另一种是将DataCon样本标记为0和1,记为2-class-Datacon. 实验采用五折交叉验证,将数据分为10部分,9部分作为训练集,1部分作为测试集,以充分评估和验证所提方法. 相关家族名称、类型号、样本数和类别编号详见表5.

3.2 性能评价指标

实验采用准确率(Accuracy,Acc)、精确率(Precision,Pr)、召回率(Recall,Rc)和F_1分数(F_1-score,F_1)四个指标对模型的性能进行评价,其公式如下:

表5 实验数据集类别

样本种类	样本名称	样本数量	样本类型
Kaggle样本	Ramnit	1 541	Worm
	Lollipop	2 478	Adware
	Kelihos_ver3	2 942	Backdoor
	Vundo	474	Trojan
	Simda	42	Backdoor
	Tracur	751	TrojanDownloader
	Kelihos_ver1	398	Backdoor
	Obfuscator.ACY	1 228	各种混淆代码
	Gatak	1 013	Backdoor
Datacon样本	Miner	5 869	Miner
	Not_Miner	10 868	Not_Miner

$$\mathrm{Acc}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{7}$$

$$\mathrm{Pr}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{8}$$

$$\mathrm{Re}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{9}$$

$$F_1\text{-score}=2\times\frac{\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{10}$$

其中,TP表示实际的阳性恶意代码样本被正确预测为阳性样本,FP表示实际的阴性恶意代码样本被错误地预测为阳性样本. 类似的,TN实际的阴性恶意代码样本被正确地预测为阴性样本,FN表示实际的阳性恶意代码样本被错误地预测为阴性样本. 多分类问题中,TP、FP、TN和FN的值如图4所示. 其中,$F_i(i=0,1,2,\cdots,n)$表示恶意代码家族类别.

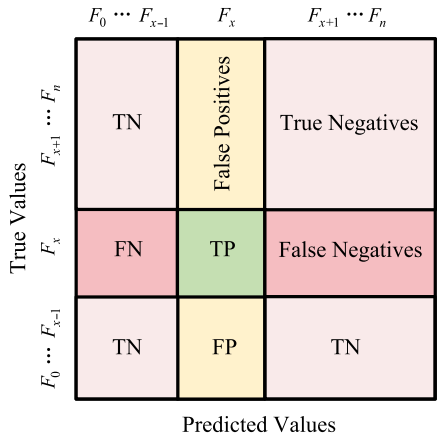

图4 多分类问题混淆矩阵

3.3 实验结果分析

3.3.1 模型与已有的恶意代码分类技术对比实验结果

为了充分证明LKRepNet模型在恶意代码检测方面的卓越性能,本节将利用Kaggle数据集和DataCon数据集进行全面对比验证. 我们选择了多种当前流行的

基于可视化分析的恶意代码检测方法作为参照,以确保评估的公正性和全面性.

对比实验的结果如表6所示,各项性能指标均清晰展示了LKRepNet模型的显著优势. 无论是准确率、召回率还是F_1分数,LKRepNet均表现优于国内外已有的其他恶意代码检测方法. 这些结果强有力地证明了LKRepNet模型在恶意代码检测领域的先进性和实用性.

由于基于现网捕获的DataCon数据集中包含了大量经过复杂加密和混淆处理的恶意代码样本,这无疑增加了检测的难度. 然而,LKRepNet模型凭借其强大的特征提取和分类能力,依然能够准确识别这些恶意代码,进一步凸显了其在实际应用中的价值.

表6　各恶意代码分类方法的性能指标　　单位:%

文献来源	数据集	模型概述	Acc	Pr	Rc	F_1
文献[20]	Kaggle	CNN+Gray	97.49	—	—	94.38
文献[39]	Kaggle	CNN+LSTM+Gray	98.20	—	—	95.77
文献[21]	Kaggle	Byte+ API+Opcode	99.24	—	—	98.72
文献[22]	Kaggle	GDMC+Gray	99.26	—	—	—
文献[23]	Kaggle	LeNet5+RGB+Word2Vec	98.76	—	—	—
文献[40]	Kaggle	RNN+Word2Vec+skip-gram	97.80	—	—	—
文献[25]	Kaggle	CFG+LSTM	87.80	—	—	84.20
文献[26]	Kaggle	GCN+CFG	99.25	99.23	99.34	99.31
文献[27]	Kaggle	CFG+Graph+transformer	92.70	—	—	90.10
文献[17]	Kaggle	One-class SVM	92.00	—	—	—
文献[41]	Kaggle	PCA＋KNN	96.60	—	—	—
文献[7]	Kaggle	Mcs-ResNet	97.21	—	—	—
文献[21]	Kaggle	Gray	97.49	—	—	—
本文	**Kaggle**	**RGB+ LKRepNet**	**99.47**	**99.45**	**99.47**	**99.49**
文献[38]	DataCon	Gray+CNN	96.80	96.42	96.26	97.38
文献[42]	DataCon	集成学习	96.99	94.05	—	92.19
本文	**DataCon**	**RGB+ LKRepNet**	**97.51**	**97.52**	**97.51**	**97.50**

3.3.2　超参数选取综合实验

在深度学习模型中,输入图像的尺寸以及优化器的选择都会显著影响模型的性能. 本节以Kaggle数据集为基准,对这些因素进行了详细的实验分析.

由于CNN中全连接层的限制,输入图像的尺寸必须是固定的. 为了确定最适合LKRepNet模型的输入尺寸,本文使用最近邻插值调整恶意代码图像至不同的尺寸(32×32、64×64、128×128、256×256和512×512),并在Kaggle数据集上进行了测试. 实验结果如表7所示,随着图像尺寸从32×32增加到256×256,模型的准确率从83.74%提升至99.47%. 然而,当图像尺寸进一步增加到512×512时,准确率反而下降至99.12%,这表明模型出现了过拟合现象. 此外,随着图像尺寸的增大,模型的参数量也不断增加,导致计算资源消耗增加和训练时间延长. 综合考虑分类精度、参数量的实验结果,我们最终选择256×256作为模型的输入图像尺寸.

在深度学习中,优化器是更新、寻找模型最优参数的算法. 出于优化模型参数的目的,本文对一些在分类任务中表现出色的优化器Adagrade、Adamax、Adam、Nadam、RAdam和Adam W进行对比实验. 表8表明,采用Adam W的模型在准确率、精确率、召回率和F_1分数方面都优于其他优化器. 因此,我们选择Adam W作为LKRepNet模型的优化器.

表7　模型在不同恶意代码图像尺寸下的性能

输入尺寸	Acc/%	Pr/%	Rc/%	F_1/%	参数量/M
32×32	83.74	83.75	83.74	83.76	0.28
64×64	95.36	95.39	95.36	95.34	0.31
128×128	98.52	98.52	98.51	98.51	0.42
256×256	99.47	99.45	99.47	99.49	0.57
512×512	99.12	99.13	99.12	99.12	1.10

表8　不同优化器对比实验结果　　单位:%

优化器	Acc	Pr	Rc	F_1
Adagrade	97.12	97.13	97.12	97.11
Adamax	97.28	97.29	97.28	97.28
Adam	98.51	98.52	98.51	98.51
NAdam	99.12	99.13	99.12	99.11
RAdam	99.38	99.39	99.38	99.38
Adam W	**99.47**	**99.45**	**99.47**	**99.49**

3.3.3　LKRepNet网络消融综合实验

LKRepNet网络在Patch Embedding层和FFN两个位置引入了大核卷积,通过结合深度大核卷积来提高不使用自注意力的早期阶段的感受野. 为验证大核卷

积对于整个网络的作用. 本节基于Kaggle数据集进行消融实验,分别使用大核卷积逐层替代自注意力模块,RM表示当前阶段使用RepMixer-FFN块,SA表示当前阶段使用Self Attention-FFN块,标准设置在块嵌入和主干层使用3×3因子化卷积,FFN使用1×1卷积. 在变体V_4和V_5中,补丁嵌入和FFN层使用了7×7大核卷积. 实验结果如表9所示.

表9 大核卷积消融实验

Variant	Stages				参数量/M	Acc/%	推理速度/ms
	1	2	3	4			
V_1	RM	RM	RM	RM	5.39	97.12	54
V_2	RM	RM	RM	SA	7.94	98.22	65
V_3	RM	RM	SA	SA	8.12	99.32	138
V_4	RM	RM	RM	RM	6.03	98.25	58
V_5	RM	RM	RM	SA	7.82	99.47	61

比较可得:V_5与V_3相较,模型大小减小了3%,且在准确率上有0.15%的增益,推理速度增加了5.6倍. V_2比V_4大32%,达到相近准确率的同时,推理速度比V_4高12%. 总体而言,大核卷积在LKRepNet上提供了1.25%的精度提升. 实验结果表明,使用深度大核卷积可以与使用自注意力机制高度竞争,同时会引起推理速度的小幅增加.

为进一步验证多分支结构以及引入大核卷积对网络表征能力的增强效果,通过网络消融方法移除LKRepNet若干个组件,分析组件对于整个网络的作用. 将LKRepNet的两个捷径分支和大核卷积块为组件,分别测试与主分支结合后模型的识别准确率. 保证其他参数不变,消融分析实验结果见表10.

表10 LKRepNet网络消融综合实验

1×1分支	BN分支	MDC分支	Acc/%	推理速度/ms
			88.14(11.33↓)	205
√			90.26(9.21↓)	184
	√		95.13(4.34↓)	141
√	√		95.29(4.18↓)	59
√	√	√	99.47	61

由表10可知,在不包含任一组件时,准确率相比LKRepNet下降了11.33%,但推理速度大幅提高. 在仅使用一个组件时,准确率相比LKRepNet也都不同程度的下降,推理速度也相应提高,结果表明多分支结构能够增加网络的表征能力,提高模型的识别准确率. 在引入大核卷积后,LKRepNet相比原始网络准确率提高了4.18%,且推理速度几乎相同,表明引入大核卷积对于推理阶段的计算资源没有显著影响.

4 总结与展望

面对新型恶意软件的快速变化和复杂性,传统检测手段常常难以适应. 本文提出了一种创新的解决方案,即基于RGB图像映射的恶意代码预处理和LKRepNet分类网络,该网络结合了CNNs和Transformer的优势,并引入了重参数化机制. 实验结果显示,本文方法有效提升了恶意软件的检测效率和模型的分类准确性. 尽管此方法在检测与分类任务上表现出色,但它受限于训练数据的质量和数量,以及深度学习模型的复杂性. 未来工作将探索数据增强和网络简化技术,研究将聚焦于探索恶意代码的新表示形式,开发跨平台、跨架构的检测方法,探索更加有效的恶意代码检测策略.

参考文献

[1] The Independent IT-Security Institute. Malware statistics[EB/OL]. (2022-02-06) [2023-07-14]. https://data-prot.net/statistics/malware-statistics.

[2] SHABTAI A, MOSKOVITCH R, ELOVICI Y, et al. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey[J]. Information Security Technical Report, 2009, 14(1): 16-29.

[3] MANAVI F, HAMZEH A. A novel approach for ransomware detection based on PE header using graph embedding[J]. Journal of Computer Virology and Hacking Techniques, 2022, 18(4): 285-296.

[4] VYAS R, LUO X, MCFARLAND N, et al. Investigation of malicious portable executable file detection on the network using supervised learning techniques[C]//2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Piscataway: IEEE, 2017: 941-946.

[5] SUN Z, RAO Z H, CHEN J F, et al. An opcode sequences analysis method for unknown malware detection[C]//Proceedings of the 2019 2nd International Conference on Geoinformatics and Data Analysis. New York: ACM, 2019: 15-19.

[6] KAN Z L, WANG H Y, XU G A, et al. Towards light-weight deep learning based malware detection[C]//2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). Piscataway: IEEE, 2018: 600-609.

[7] SHAO Y L, LU Y, WEI D, et al. Malicious code classification method based on deep residual network and hybrid attention mechanism for edge security[J]. Wireless Communications and Mobile Computing, 2022, 2022: 3301718.

[8] 李晓勇, 马威. 动态代码的实时可信传递研究[J]. 电子学报, 2012, 40(10): 2009-2014.
LI X Y, MA W. Research on real-time transitive trust for dynamic codes[J]. Acta Electronica Sinica, 2012, 40(10): 2009-2014. (in Chinese)

[9] JACOB G, DEBAR H, FILIOL E. Behavioral detection of malware: From a survey towards an established taxonomy[J]. Journal in Computer Virology, 2008, 4(3): 251-266.

[10] 刘豫, 王明华, 苏璞睿, 等. 基于动态污点分析的恶意代码通信协议逆向分析方法[J]. 电子学报, 2012, 40(4): 661-668.
LIU Y, WANG M H, SU P R, et al. Communication protocol reverse engineering of malware using dynamic taint analysis[J]. Acta Electronica Sinica, 2012, 40(4): 661-668. (in Chinese)

[11] ELOVICI Y, SHABTAI A, MOSKOVITCH R, et al. Applying machine learning techniques for detection of malicious code in network traffic[C]//Lecture Notes in Computer Science. Berlin: Springer Berlin Heidelberg, 2007: 44-50.

[12] SHABTAI A, MOSKOVITCH R, ELOVICI Y, et al. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey[J]. Information Security Technical Report, 2009, 14(1): 16-29.

[13] MOHAMMED K B. Ransomware detection using random forest technique[J]. ICT Express, 2020, 6(4): 325-331.

[14] LI X, QIU K F, QIAN C, et al. An adversarial machine learning method based on opcode n-grams feature in malware detection[C]//2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC). Piscataway: IEEE, 2020: 380-387.

[15] BARSHAN E, GHODSI A, AZIMIFAR Z, et al. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds[J]. Pattern Recognition, 2011, 44(7): 1357-1371.

[16] MOON H J, BU S J, CHO S B. Directional graph transformer-based control flow embedding for malware classification[M]//Intelligent Data Engineering and Automated Learning — IDEAL 2021. Cham: Springer International Publishing, 2021: 426-436.

[17] NATARAJ L, KARTHIKETAN 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, 2011: 1-7.

[18] CAI L R, LI Y, XIONG Z. JOWMDroid: Android malware detection based on feature weighting with joint optimization of weight-mapping and classifier parameters[J]. Computers & Security, 2021, 100: 102086.

[19] VU D L, NGUYEN T K, NGUYEN T V, et al. A convolutional transformation network for malware classification[C]//2019 6th NAFOSTED Conference on Information and Computer Science (NICS). Piscataway: IEEE, 2019: 234-239.

[20] GIBERT D, MATEU C, PLANES J, et al. Using convolutional neural networks for classification of malware represented as images[J]. Journal of Computer Virology and Hacking Techniques, 2019, 15(1): 15-28.

[21] YUAN B, WANG J, LIU D, et al. Byte-level malware classification based on Markov images and deep learning[J]. Computers & Security, 2020, 92: 101740.

[22] QIAO Y C, JIANG Q S, JIANG Z C, et al. A multi-channel visualization method for malware classification based on deep learning[C]//2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science And Engineering (TrustCom/BigDataSE). Piscataway: IEEE, 2019: 757-762.

[23] LEI T, XUE J, WANG Y, et al. An empirical study of problems and evaluation of IoT malware classification label sources[J]. Journal of King Saud University-Computer and Information Sciences, 2024, 36(1): 101898.

[24] HUO X, LI M, ZHOU Z H. Control flow graph embedding based on multi-instance decomposition for bug localization[C]//The Thirty-Fourth AAAI Conference on Artificial Intelligence. New York: AAAI, 2020: 4223-4230.

[25] YAN J Q, YAN G H, JIN D. Classifying malware represented as control flow graphs using deep graph convolutional neural network[C]//2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Piscataway: IEEE, 2019: 52-63.

[26] 王硕, 王坚, 王亚男, 等. 一种基于特征融合的恶意代码快速检测方法[J]. 电子学报, 2023, 51(1): 57-66.
WANG S, WANG J, WANG Y N, et al. A fast malicious code detection method based on feature fusion[J]. Acta Electronica Sinica, 2023, 51(1): 57-66. (in Chinese)

[27] 轩勃娜, 李进. 基于改进CNN的恶意软件分类方法[J]. 电子学报, 2023, 51(5): 1187-1197.
XUAN B N, LI J. A malware classification method based on improved CNN[J]. Acta Electronica Sinica, 2023, 51 (5): 1187-1197. (in Chinese)

[28] CUI Z H, XUE F, CAI X J, et al. Detection of malicious code variants based on deep learning[J]. IEEE Transactions on Industrial Informatics, 2018, 14(7): 3187-3196.

[29] YU W H, LUO M, ZHOU P, et al. Metaformer is actually what you need for vision[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2022: 10819-10829.

[30] JIAN Y F, KUANG H B, REN C L, et al. A novel framework for image-based malware detection with a deep neural network[J]. Computers & Security, 2021, 109: 102400.
[31] VENKATRAMAN S, ALAZAB M, VINAVAKUMAR R. A hybrid deep learning image-based analysis for effective malware detection[J]. Journal of Information Security and Applications, 2019, 47: 377-389.
[32] VASAN D, ALAZAB M, WASSAN S, et al. Image-Based malware classification using ensemble of CNN architectures (IMCEC)[J]. Computers & Security, 2020, 92: 101748.
[33] AZEEZ N A, ODUFUWA O E, MISRA S, et al. Windows PE malware detection using ensemble learning[J]. Informatics, 2021, 8(1): 10.
[34] LE Q, BOYDELL O, NAMEE B MAC, et al. Deep learning at the shallow end: Malware classification for non-domain experts[J]. Digital Investigation: the International Journal of Digital Forensics & Incident Response, 2018, 26(S): S118-S126.
[35] HAN K, KANG B, IM E G. Malware analysis using visualized image matrices[J]. The Scientific World Journal, 2014, 2014: 132713.
[36] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16×16 words: Transformers for image recognition at scale[EB/OL]. (2020-10-22) [2023-11-15]. https://arxiv.org/abs/2010.11929.
[37] Microsoft Malware Classification Challenge. Kaggle BIG 2015 dataset[DB/OL]. (2015-04-18) [2023-11-15] https://www.kaggle.c-om/c/malware-classification/data.
[38] 奇安信技术研究院. DataCon: 面向安全研究的多领域大规模竞赛开放数据[EB/OL]. (2021-11-11)[2023-11-15]. https://datacon.qi-anxin.com/opendata.
QinetiQ Institute of Technology. DataCon: Open data for multi-domain, large-scale competitions for security research[EB/OL]. (2021-11-11) [2023-11-15]. https://datacon.qi-anxin.com/opendata. (in Chinese)
[39] LE Q, BOYDELL O, NAMEE B MAC, et al. Deep learning at the shallow end: Malware classification for non-domain experts[J]. Digital Investigation, 2018, 26(S): S118-S126.
[40] CHEN J, GUO S Z, MA X, et al. SLAM: A malware detection method based on sliding local attention mechanism[J]. Security and Communication Networks, 2020, 2020: 6724513.
[41] NARAYANAN B N, DJANEYE-BOUNDJOU O, KEBEDE T M. Performance analysis of machine learning and pattern recognition algorithms for malware classification[C]//2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS). Piscataway: IEEE, 2016: 338-342.
[42] 杨望, 高明哲, 蒋婷. 一种基于多特征集成学习的恶意代码静态检测框架[J]. 计算机研究与发展, 2021, 58(5): 1021-1034.
YANG W, GAO M Z, JIANG T. A malicious code static detection framework based on multi-feature ensemble learning[J]. Journal of Computer Research and Development, 2021, 58(5): 1021-1034. (in Chinese)

作者简介

李思聪 女, 2000 年 8 月出生于陕西省西安市. 现为空军工程大学防空反导学院硕士研究生. 主要研究方向为智能信息处理和恶意代码检测.
E-mail: lisicong0813@163.com

王 坚 男, 1982 年 2 月出生于陕西省渭南市. 现为空军工程大学防空反导学院副教授. 主要研究方向为智能信息处理和恶意代码检测.
E-mail: 26471375@qq.com

宋亚飞 男, 1988 年出生于河南汝州. 现为空军工程大学防空反导学院副教授. 主要研究方向为机器学习及其在目标识别和入侵检测等领域中的应用.
E-mail: yafei_song@163.com

王 硕 男, 1998 年 11 月出生于重庆市合川区. 现为解放军 95285 部队助理工程师. 主要研究方向为智能信息处理和基于深度学习的图像检测.
E-mail: Luoan_W@163.com