

基于多通道记忆融合的恶意代码检测方法

刘超¹, 黎秋玲², 赵学健²

(1. 南京数睿数据科技有限公司, 江苏南京 211100;
2. 南京邮电大学宽带无线通信与传感网技术教育部重点实验室, 江苏南京 210003)

摘要: 随着恶意代码规模和种类的不断增加, 传统恶意代码检测难以应对恶意软件变体和复杂的加密样本。为此, 提出一种基于多通道混合卷积网络的恶意代码检测模型, 旨在提升复杂环境下的恶意代码检测性能。该模型引入一种新的自适应去噪机制, 从多个角度获取恶意代码特征后形成多通道图像特征, 融合交叉熵与软阈值函数进行去噪处理, 抑制小于阈值的特征并保留大于阈值的特征, 解决了特征图像缩放引起的特征信息丢失问题。此外, 该模型提出一种基于 DenseNet121 与门控循环单元 (GRU) 的特征融合策略, 在 DenseNet121 的过渡层后引入 GRU 模块, 构建多通道共享记忆模块 (SMM), 捕捉多通道特征间的依赖关系, 避免模型过度依赖单一特征通道。实验结果表明, 与现有的恶意代码检测方法相比, 该模型在精确率、准确率、召回率和 F1 分数等指标上均有一定提升, 特别是在处理恶意软件变体和复杂加密样本时, 展现出更强的泛化能力和抗干扰能力。

关键词: 恶意代码检测; 多通道; 记忆融合; 门控循环单元; 深度学习

DOI: 10.11907/rjtk.251307

中图分类号: TP309

文献标识码: A

文章编号: 1672-7800(2025)012-0183-07

扫描二维码阅读全文:



Malware Detection Method Based on Multi-Channel Memory Fusion

LIU Chao¹, LI Qiuling², ZHAO Xuejian²

(1. Nanjing Shurui Data Technology Co., Ltd., Nanjing 211100, China; 2. Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: With the continuous increase in the scale and variety of malicious code, traditional detection methods struggle to address malware variants and complex encrypted samples. To tackle this challenge, a multi-channel hybrid convolutional network-based malicious code detection model aimed at enhancing detection performance in complex environments is proposed. The model introduces an innovative adaptive denoising mechanism. After extracting multi-angle features of malicious code to form multi-channel image representations, it combines cross-entropy with a soft threshold function for denoising. This approach suppresses features below the threshold while preserving those above it, effectively mitigating feature loss caused by image scaling. Furthermore, the model proposes a feature fusion strategy based on DenseNet121 and Gated Recurrent Units (GRU). By integrating a GRU module after the transition layers of DenseNet121, it constructs a Shared Memory Module (SMM) to capture dependencies among multi-channel features, thereby preventing over-reliance on any single feature channel. Experimental results demonstrate that compared to existing malware detection methods, the model achieves improvements in precision, accuracy, recall, and F1-score. Notably, it exhibits stronger generalization and anti-interference capabilities when handling malware variants and complex encrypted samples.

Key Words: malicious code detection; multi-channel; memory fusion; gated recurrent unit; deep learning

0 引言

在当今的数字化时代, 恶意代码已成为网络安全领域

的主要威胁之一。随着互联网和计算机技术的迅速发展, 恶意软件变得越来越复杂和多样化, 给计算机系统和网络带来了严重的安全风险。恶意代码的种类繁多, 包括病毒、蠕虫、特洛伊木马、勒索软件等, 其通过不同的途径感

收稿日期: 2025-04-27

基金项目: 国家自然科学基金项目 (62303239)

作者简介: 刘超 (1981-), 男, 硕士, 南京数睿数据科技有限公司工程师, 研究方向为机器学习、软件工程等; 黎秋玲 (2001-), 女, 南京邮电大学宽带无线通信与传感网技术教育部重点实验室硕士研究生, 研究方向为机器学习、路径规划等; 赵学健 (1982-), 男, 博士, CCF 会员, 南京邮电大学宽带无线通信与传感网技术教育部重点实验室副教授, 研究方向为人工智能、无线网络。本文通讯作者: 赵学健。

染计算机系统,窃取敏感信息、破坏数据或干扰系统正常运行。如何有效检测和防范这些恶意软件,已成为研究人员和安全从业者面临的重大挑战。

传统的恶意代码检测方法主要分为基于签名检测和基于行为分析两种。基于签名检测的方法通过匹配已知恶意代码的特征签名来识别恶意软件,具有检测速度快、误报率低的优势。然而,其核心依赖于已知恶意代码的特征库,一旦恶意软件采用代码混淆、指令重排、加密包装等变种技术,原有签名将失效,导致检测能力大幅下降。因此,该方法难以有效应对新型恶意软件及其变种。

相比之下,基于行为分析的方法通过监测程序的运行行为来判断其是否为恶意软件,能够一定程度上检测未知的恶意代码。然而,这类方法严重依赖行为特征的定义,一旦恶意代码采用规避检测技术(如环境感知、沙箱逃逸等),行为模式可能发生变化,导致检测准确率下降。此外,行为分析通常依赖于动态监测,计算开销较大,难以满足实时检测的需求。

近年来,机器学习和深度学习技术的快速发展为恶意代码检测提供了新的解决思路。这些方法通过数据驱动的方式自动学习恶意代码的特征,无需依赖人工定义,从而突破了传统方法的局限。深度学习方法尤其在特征提取和泛化能力上展现出显著优势。例如,卷积神经网络(Convolutional Neural Network, CNN)能够自动提取恶意代码的静态模式特征,而循环神经网络(Recurrent Neural Network, RNN)能够有效建模恶意代码的动态行为特征。此外,深度学习模型通过端到端的训练方式学习不同层次的特征表示,从而提升对恶意软件变种和加密样本的检测能力。

在具体应用中,研究人员已提出多种基于机器学习和深度学习的检测方法。例如,支持向量机(Support Vector Machine, SVM)、随机森林(Random Forest, RF)等传统机器学习算法在静态恶意代码检测中得到了广泛应用,而深度学习方法如 CNN、RNN 及其变种(如 Transformer)进一步增强了特征提取能力,使得恶意代码检测模型在复杂环境下的表现更加稳定,且具有更强的鲁棒性。实验结果表明,相较于传统方法,深度学习方法在检测精度、泛化能力和抗规避攻击能力方面均取得了显著提升。

尽管基于机器学习和深度学习的方法在恶意代码检测中已取得了一定进展,显著提高了检测的准确性和效率,但这些方法仍然存在不足之处。首先,传统的特征提取和去噪方法往往难以有效处理恶意代码中存在的冗余干扰信息和复杂特征,容易导致特征信息的丢失或过度压缩,从而影响检测性能。其次,现有模型通常过于依赖单一特征通道,难以充分挖掘和利用来自不同角度的多通道信息,导致模型的鲁棒性不足,特别是在面对恶意软件变体和加密样本时,泛化能力较弱。此外,现有方法在多通道特征融合方面还存在一定的局限性,未能有效解决多通

道特征之间的相互依赖问题,可能导致模型对某些特征的过度依赖,从而影响整体检测效果。

为解决上述问题,本文提出一种基于多通道混合卷积记忆网络的恶意代码检测模型。首先将恶意代码文件转化为三通道图像,结合 Shannon 交叉熵与软阈值函数进行降噪处理,随后引入多通道共享记忆模块构建 DenseGRU-Net 模型,融合输入图像的多通道特征。最后,将 DenseGRU-Net 提取和融合的高维多通道特征输入 ResNet 分类器进行分类检测。

本文主要工作及创新点如下:

(1)针对恶意代码图像中的噪声问题,提出一种基于 Shannon 交叉熵与软阈值函数的自适应去噪机制。将提取的恶意软件二进制文件信息转换为 3 种不同的灰度图像,并将其融合为三通道 RGB 图像后,利用 Shannon 交叉熵衡量特征信息的不确定性,并结合软阈值函数对低置信度特征进行抑制,有效实现了噪声的动态去除。同时保留关键特征信息,解决了特征图像缩放造成的特征信息丢失问题,提升了模型在复杂环境中的检测效果和抗噪能力。

(2)针对多通道特征间信息交互不足的问题,提出一种引入多通道共享记忆模块的 DenseGRU-Net 模型用于融合输入图像的多通道特征。该模块通过门控循环单元(Gated Recurrent Unit, GRU)在通道间实现动态特征共享与依赖关系更新,能够捕捉和整合不同通道间的潜在相关性,有效提升各通道特征的互补表达,避免特征孤立造成的信息丢失。

(3)在 BIG 2015 数据集与 Ember 2018 数据集上验证了所提出 DenseSMM-Net 模型的性能。

1 相关工作

为应对日益增长的恶意代码威胁,研究人员积极应用多种检测和分类方法,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方法,如卷积神经网络、循环神经网络等。Shabtai 等^[1]总结了基于静态特征的机器学习检测方法,采用不同分类器对新型恶意代码进行识别。该方法通过多种静态特征组合提升检测精度,但其特征选择依赖人工经验,且在面对混淆或加密样本时检测能力有限。Zhang 等^[2]提出一种仿生免疫系统启发的恶意代码检测方法,采用进程行为监测与家族基因分析,通过抗体生成与基因进化识别未知恶意代码。实验结果显示该方法具有高检测率和低误报率,但特征建模过程复杂,泛化能力受限于样本的多样性。Cheng 等^[3]构建了集成学习框架,融合签名检测、行为检测与知识图谱分析,提高了检测的可解释性。但该框架依赖多个子模块配合,计算资源消耗大,难以部署于轻量级终端。Lindorfer 等^[4]采用动态分析对恶意软件的功能模块进行语义赋值,并通过版本比较观察其变化。尽管该方法具有语义解释性,但其严重依

赖于沙箱环境,易被具备逃逸能力的恶意代码规避,且检测过程耗时较长。

近年来,恶意代码可视化结合机器学习技术的检测方法成为研究热点。通过将恶意代码转换为图像形式,再结合图像特征提取方法进行检测和分类。Yin等^[5]将恶意代码二进制转为调用图图像,提取图像指纹特征进行检测。该方法在稳定性和灵敏度方面表现良好,但图像特征层次较浅,难以捕捉深层语义特征。Ikram等^[6]提出一种恶意软件分类方法,通过将恶意软件的二进制文件转化为灰度图像,并使用GIST(Generalized Image Statistic)描述符提取图像特征。提取后的特征通过K最近邻(K-Nearest Neighbor, KNN)算法进行分类,该方法简单、高效,但对图像旋转、缩放等扰动敏感,鲁棒性不足。Manavi等^[7]提出一种基于操作码可视化的恶意软件检测方法,将操作码图像化后使用GIST方法提取特征,然后通过KNN、SVM等机器学习方法进行分类。该方法在降低复杂度的同时提高了检测精度。Darus等^[8]针对安卓平台恶意软件,将其二进制文件转化为灰度图像,并使用GIST描述符提取特征,随后通过KNN、随机森林(RF)和决策树(DT)算法进行检测。该方法能够有效检测新创建的恶意软件。Sheneamer等^[9]实现了一种基于图像可视化的恶意软件检测方法,通过将恶意软件二进制文件转化为图像,提取图像特征并应用机器学习方法进行分类。研究结果表明,使用卷积神经网络(CNN)算法在多种恶意软件数据集上的分类准确性较高,而KNN在传统分类方法中表现最好。

Mohammed等^[10]提出一种新颖的恶意软件检测方法,将恶意软件二进制文件通过离散余弦变换(Discrete Cosine Transform, DCT)转换为灰度图像,并使用浅层神经网络和深层网络架构进行分类。该方法结合GIST描述符等特征,能够有效分类和检测恶意软件。尽管计算效率高,但模型深度不足,易受特征干扰影响。Yang等^[11]提出一种新的多尺度卷积模型,结合了多重注意力机制。该模型将注意力机制引入到Res2块结构中,通过通道排序和分组实现多尺度卷积。实验结果表明,该模型在图像分类方面具有优越的性能,但未针对恶意代码图像中的噪声与变异特征进行专门的设计。Yang等^[12]提出卷积自注意力网络,通过多头注意力机制增强自注意力网络的性能,特别是加强邻近元素之间的依赖性和不同注意力头之间的特征交互。实验结果表明,该方法在多语言翻译任务上优于强基线Transformer模型。Wu等^[13]提出一种多粒度注意力网络(Multi-Grained Attention Network, MGAN),通过在多尺度下计算神经元的重要性来自适应地重新缩放特征响应。该方法在单图像超分辨率任务中显示出优越的性能,但不适用于复杂类别的分类任务。

恶意代码检测的可视化方法虽然在特征提取和分类任务中取得了一定进展,但在面对复杂环境和多样化的恶意代码样本时,其性能仍存在明显的提升空间。仅通过可

视化转换提取图像特征的方法,可能无法充分捕捉恶意代码深层次的特征模式,尤其是面对特征表现力较弱或易受噪声干扰的数据。为此,研究人员进一步探索基于深度学习的恶意代码检测方法,以期通过更强的特征表达能力和自动特征学习机制,弥补可视化方法的不足。Bello等^[14]提出一种增强卷积网络的注意力机制,通过引入相对自注意力机制,将卷积特征图与通过自注意力机制生成的特征图连接起来。实验结果表明,该方法在ImageNet图像分类和COCO目标检测任务上的性能均获得了显著提升,但该方法训练成本高,且在处理结构扰动时鲁棒性一般。Wang等^[15]提出一种基于多尺度特征提取和注意力机制的图像超分辨率方法,通过级联多个多尺度注意力块(Multi-scale Attention Blocks, MSAB)来平衡模型的复杂性和综合性能。实验结果表明,相较于传统超分辨率模型,该方法在多个数据集上表现更优,且在高频细节和纹理的精准恢复上具备明显优势。

尽管现有研究在图像转换、特征提取与分类建模等领域进行了广泛探索,但依旧存在不足之处。多数方法采用单通道图像建模,未能融合来自不同视角的特征信息,且缺乏有效的通道间依赖建模机制,导致模型对特征间的关系捕捉不足。上述文献中普遍未针对图像中存在的噪声或混淆特征进行自适应处理,易造成关键信息丢失,影响检测的准确性。因此,迫切需要一种融合多通道特征、具备动态记忆能力和自适应去噪机制的检测模型,以提升模型在复杂环境下的鲁棒性和泛化能力。

2 本文方法

本文提出的恶意代码检测方法主要包含3部分:多通道特征提取、自适应去噪机制和模型构建。在特征提取部分,首先将恶意代码转化为3种不同的灰度图以捕捉不同信息,通过自适应插值技术统一图像尺寸,再将3种灰度图融合为三通道RGB图像,完成多通道特征的初步提取。

2.1 多通道特征提取

2.1.1 灰度图生成

本文提出3种灰度图像转换方法,分别是ASCII图像转换、十六进制图像转换以及熵图像转换。其中,ASCII图像主要用于保留文件文本信息,如可见字符串、恶意域名、路径等;十六进制图像用于捕捉二进制数据的全局结构模式;熵图像能够衡量恶意代码的复杂性,辅助识别加密或混淆代码。

ASCII图像生成步骤如下:①将恶意代码解析得到的信息序列转换为连续的ASCII字符;②固定生成的ASCII图像宽度为256像素,将连续ASCII字符的前256字节作为第一行像素,将257~512个字节作为第二行像素,依次类推;③当最后一行字节序列小于256字节时,使用0字节进行填充。

十六进制图像生成步骤如下:①针对恶意软件二进制文件中的十六进制值,将每8位转换为一个灰度像素值(范围0~255);②根据数据集样本量的不同,将恶意代码文件划分为不同大小的区间;③当最后一行不足设置的宽度与长度时,则用0x00进行填充,以确保每行像素达到固定宽度。

熵图生成步骤如下:①将恶意代码解析得到二进制序列的每8位作为一个单位进行划分,计算每个窗口的熵值;②将生成的熵值映射为0~255之间的整数;③将生成的熵灰度值以256像素的宽度排列,第一个窗口的熵值作为第一行的一个像素,依次类推,形成二维灰度矩阵。当最后一行的像素不足时,使用零值进行填充,以达到固定宽度。

2.1.2 图像尺寸调整与融合

通过自适应插值,对图像的不同区域应用不同的插值策略:在边缘区域,由于像素变化较大,需要更精细的插值方法,以保留图像细节;在平坦或平滑区域,插值方法可以相对简单,以减少计算量,并防止过度平滑导致的细节丢失。设图像像素点 $I(x, y)$ 的插值结果为 $I_{\text{adaptive}}(x, y)$,则自适应插值的计算公式为:

$$I_{\text{adaptive}}(x, y) = \begin{cases} I_{\text{bicubic}}(x, y), & \text{Var} \geq \tau \\ I_{\text{bilinear}}(x, y), & \text{otherwise} \end{cases} \quad (1)$$

其中, $I_{\text{bicubic}}(x, y)$ 为三次插值的结果, $I_{\text{bilinear}}(x, y)$ 为双线性插值的结果。 Var 是N-gram特征的方差,用于判断该区域是否属于边缘区域,阈值 τ 用于区分边缘和平滑区域。

三次插值的计算公式如下:

$$I_{\text{bicubic}}(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 I(x_i, y_j) \cdot W(x - x_i) \cdot W(y - y_j) \quad (2)$$

其中, $W(x)$ 是三次插值的权重函数,定义为:

$$W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & 1 < |x| < 2 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

通过线性插值算法对3种不同方法生成的灰度图像进行放大和缩小,生成256×256大小的图像。采用ASCII图像作为第一通道,熵图像作为第二通道,十六进制图像作为第三通道,形成三维RGB图像。

使用CLAHE算法对生成的RGB图像进行增强。首先,将图像分割成大小相等的子区域,计算每个子区域的直方图;其次,定义阈值 β ,如果块的直方图超过定义的阈值,则直方图从顶部修剪,超过阈值的部分均匀分布到每个灰度级,直到实现均衡;最后,以各子区域的中心参考点获取灰度值,并用插值算法进行处理,得到增强图像。CLAHE算法是对自适应直方图均衡化算法的改进,在抑制噪声的同时可以增强图像的对比度。阈值 β 计算公式如下:

$$\beta = \frac{P}{Q} \left(1 + \frac{\alpha}{100} (y_{\max} - 1) \right) \quad (4)$$

2.2 自适应去噪机制

恶意代码检测中存在一些无关紧要的特征。在生成三维RGB图像之后引入一种基于交叉熵的去噪权重方法,可以有效降噪并帮助模型专注于关键特征,从而提高分类的准确性和模型的鲁棒性。同时减少计算复杂度,避免过拟合问题。将Shannon交叉熵引入软阈值函数,通过计算每个节点的特征分布与其邻域特征分布之间的交叉熵,能够有效识别出图像中具有显著差异的节点。交叉熵越大,表明节点与其邻域之间的特征差异越大,这可能是由于噪声干扰导致的。因此,对于这些差异较大的节点,需要施加更强的去噪力度,以确保有效抑制噪声。同时,交叉熵的引入能够动态调整软阈值函数的阈值,使得降噪过程更加自适应。

对于每一幅灰度图像,将其转化为一个图结构 $G = (V, E)$ 。其中, V 表示图的节点集合,每个节点对应图像中的一个像素; E 表示图的边集合,每条边连接相邻的像素节点。灰度图像被表示为一个图结构,其中像素之间的空间关系被转换为图的连接关系。对于每个节点 v_i ,定义其特征分布为 P_i ,该分布可以通过节点及其邻域的像素值直方图来表示。类似地,其邻域中的节点集合可以定义为 $N_{(v_i)}$,邻域节点的特征分布 Q_i 可以通过对邻域中所有节点特征进行归一化处理获得。具体定义如下:节点的特征分布 P_i 为基于节点 V_i 及其邻域的像素值计算的概率分布;邻域的特征分布 Q_i 为基于节点 V_i 邻域内所有节点的像素值计算的概率分布。

交叉熵用于衡量节点 V_i 与其邻域 $N_{(v_i)}$ 之间的特征分布差异。具体来说,节点 V_i 的特征分布 P_i 与邻域特征分布 Q_i 之间的Shannon交叉熵 $H(P_i, Q_i)$ 计算公式如下:

$$H(P_i, Q_i) = - \sum_{k=1}^n P_i(x_k) \log Q_i(x_k) \quad (5)$$

其中, $P_i(x_k)$ 表示节点 v_i 特征分布中像素值 x_k 的概率, $Q_i(x_k)$ 表示其邻域特征分布中对应像素值 x_k 的概率。交叉熵越大,表明节点 V_i 与其邻域之间的特征差异越大,这通常暗示该节点可能受到噪声影响较大,或位于图像的重要边缘或细节区域。

为了实现自适应降噪,本文将Shannon交叉熵引入软阈值函数,并根据交叉熵的大小动态调整阈值 λ_i 。具体而言,对于每个节点 V_i 的像素值 x_i ,其软阈值函数定义如下:

$$S_{\lambda_i}(x_i) = \text{sgn}(x_i) \cdot \max(|x_i| - \lambda_i, 0) \quad (6)$$

其中, $\text{sgn}(x_i)$ 为符号函数, λ_i 为基于交叉熵 $H(P_i, Q_i)$ 动态调整的阈值。阈值 λ_i 可以根据以下公式确定:

$$\lambda_i = \alpha \cdot H(P_i, Q_i) \quad (7)$$

其中, α 是一个可调节的超参数,用于控制去噪的强度。交叉熵较大的节点将被赋予更高的阈值 λ_i ,以便更好

地抑制噪声,同时保留图像的重要特征。

在应用软阈值函数进行降噪时,首先计算每个节点 V_i 与其邻域的交叉熵 $H(P_i, Q_i)$,然后根据该交叉熵动态调整软阈值函数的阈值 λ_i 。最后对每个节点的像素值应用软阈值函数,得到去噪后的像素值 x'_i :

$$z_i = \sigma(W_z \cdot [h_{i-1}, x_i]) \quad (8)$$

通过此方式,噪声较大的区域将得到更好的降噪处理,而图像的边缘和细节区域则可以较好地保留其特征信息,从而实现更好的图像去噪效果。降噪处理完成后,将所有节点的去噪像素值 x'_i 重新组合为新图像。

2.3 模型构建

基于 GRU 与深度卷积网络的思想,本文提出一种基于

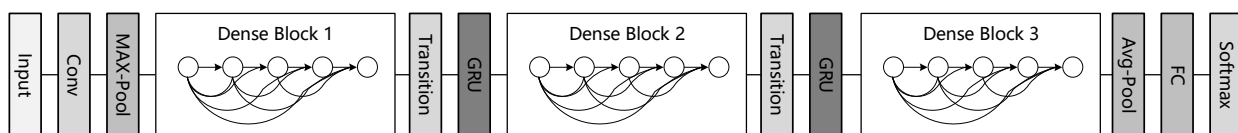


Fig. 1 The architecture of DenseGRU-Net

图 1 DenseGRU-Net 框架

该模型中引入的多通道共享记忆模块结合动态记忆机制和门控机制,实现了多通道特征的有效共享和增强。

假设输入为多通道特征 $X_i \in \mathbb{R}^{C \times H \times W}$,其中 C 是通道数。将特征图展平成时间序列的形式: $X_i \rightarrow [x_1, x_2, \dots, x_C]$,每个通道特征作为 GRU 的一个时间步。GRU 的更新公式如下:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (9)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (10)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \circ h_{t-1}, x_t]) \quad (11)$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \quad (12)$$

其中, z_t 为更新门,控制当前状态和前一状态的融合; r_t 为重置门,控制前一状态对当前候选状态的影响; $\tilde{h}_t$ 为当前隐状态; W_z, W_r, W_h 为 GRU 的权重矩阵。GRU 输出经过所有时间步后,生成一个融合的隐状态 H_T ,表示所有通道特征的全局表示。将 H_T 重新整形为特征图的形式 $H_T \in \mathbb{R}^{C \times H \times W}$,用于后续网络计算。

由于 DenseGRU-Net 模型在结构设计上引入了多个密集连接块 (Dense Block),其特征提取部分包含大量卷积层,这在一定程度上增强了模型对复杂特征的提取能力。然而,深层网络结构也可能带来过拟合风险,尤其是在训练样本有限或数据分布偏移的情况下。为在保持模型表达能力的同时提高泛化性能,本文对 DenseGRU-Net 的结构进行了调整,重点优化了各级密集块的层数配置。

具体而言,将第 1 密集块从 6 层减少到 4 层,第 2 密集块从 12 层减少到 8 层,第 3 密集块从 24 层减少到 16 层,第 4 密集块从 16 层减少到 12 层。该调整既保留了网络浅层对基础图像边缘、纹理等底层特征的敏感性,又削减了中高层网络的冗余计算,降低了参数规模与计算复杂度,进

DenseGRU-Net 的恶意代码检测模型。该模型在 DenseNet121 过渡层 (Transition Layer) 后引入 GRU 形成共享记忆模块 (Shared Memory Module, SMM),以动态捕捉通道间的特征依赖。

具体而言,过渡层输出的特征图被展开为序列输入 GRU,使 GRU 通过门控机制选择性地保留重要信息,并在多个通道之间共享上下文信息。这一机制使模型能够动态调整不同通道特征的影响力,从而克服传统 CNN 仅能独立处理各通道特征的局限性。最终,经过 GRU 处理的多通道特征被重新整形并输入后续 DenseBlock,以实现端到端的特征融合。

具体模型架构如图 1 所示。

而抑制过拟合现象的发生。

综合上述优化设计,DenseGRU-Net 模型在实现结构简化的同时,保持了较强的特征表示与融合能力。其整体架构通过密集连接策略实现特征复用,并增强了信息流动。GRU 模块引入时间序列建模思想以加强特征通道间的语义耦合,而自适应去噪机制则提升了模型在处理恶意代码变种与加密样本时的鲁棒性。表 1 详细描述了图 1 中 DenseGRU-Net 的各层结构参数配置。

3 实验与结果分析

3.1 实验环境及数据集

本文实验环境为 64 位 Windows 11 操作系统,搭载 Intel(R) Core(TM) i5-10210U 处理器,16 GB 内存,开发环境使用 Tensorflow 和 Keras 深度学习框架。实验数据集包括 BIG 2015 和 Ember 2018。其中,BIG 2015 数据集包含 9 类恶意代码家族,共计 10 868 个样本,广泛用于恶意代码检测研究,提供了可靠的基准测试环境。Ember 2018 数据集涵盖 1 000 000 个样本,包括恶意软件和良性软件。为了降低计算成本并保证实验的可行性,本文从中随机抽取 30 000 个样本(包括 15 000 个恶意样本、15 000 个良性样本)进行实验。该数据集包含每个样本的二进制文件和反汇编代码,可提取多种静态特征,如 ASCII 表示、熵特征和十六进制结构,以提供更丰富的恶意代码特征信息。本文按照约 6:4 的比例划分训练集和测试集。

3.2 评价指标

本文采用准确率 (Accuracy)、精确率 (Precision)、召回率 (Recall)、F1 分数 (F1-score) 评价指标进行模型评价。

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (13)$$

Table 1 DenseGRU-Net network structure parameters

表1 DenseGRU-Net 网络结构参数

网络结构	输出大小	参数	输出 通道数	层数	压缩因子
Conv	112×112	7×7, 步长 2	64	--	--
Maxpool	56×56	3×3, 步长 2	64	--	--
DenseBlock 1	56×56	1×1 Conv 3×3 Conv 1×1 Conv	128	4	--
Transition1	28×28	3×3 Avgpool 步长 2	64	--	0.5
DenseBlock 2	28×28	1×1 Conv 3×3 Conv	192	8	--
GRU 1	28×28	--	--	--	--
Transition 2	14×14	1×1 Conv 2×2 Avgpool 步长 2	96	--	0.5
DenseBlock 3	14×14	1×1 Conv 3×3 Conv	384	16	--
GRU 2	14×14	--	--	--	--
Transition 3	7×7	1×1 Conv 2×2 Avgpool 步长 2	192	--	0.5
DenseBlock 4	7×7	1×1 Conv 3×3 Conv	384	12	--
Avgpool	1×1	--	384	--	--
FC	1×1	--	输出类别数	--	--

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \quad (14)$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad (15)$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \quad (16)$$

3.3 实验结果分析

3.3.1 消融实验

为验证本文提出的 DenseGRU-Net 模型中多通道共享记忆模块和动态记忆机制的有效性,本文设计了消融实验。实验基于以下模型进行对比:SCFM 模型直接使用 DenseNet 架构处理单通道图像特征,不包含多通道融合和 GRU 模块;MCFC 模型为无动态共享机制的多通道特征拼接模型,该模型对多通道图像特征进行简单拼接后输入 DenseNet 架构进行处理,不包含 GRU 模块用于捕捉通道间的依赖关系。训练集和验证集准确率变化如图 2、图 3 所示。

由图 2、图 3 可以看出,DenseGRU-Net 模型在准确率上显著优于其他模型,充分验证了其多通道共享记忆模块和动态记忆机制的有效性。具体而言,SCFM 模型仅能处理单通道特征,缺乏对多通道信息的利用能力,因此性能较差;而 MCFC 模型虽然引入了多通道特征拼接,但由于

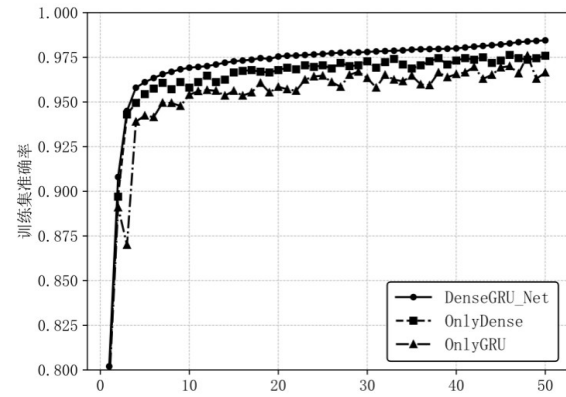


Fig. 2 Change in accuracy of the training set

图2 训练集准确率变化

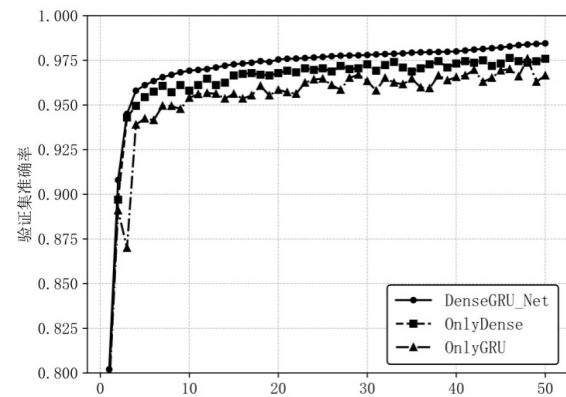


Fig. 3 Change in accuracy of the validation set

图3 验证集准确率变化

未能捕捉通道间的动态依赖关系,其性能提升有限。相比之下,DenseGRU-Net 利用 GRU 模块在多通道特征融合中动态捕捉依赖关系,有效提升了模型对恶意代码复杂特征的表达能力,同时多通道特征的动态共享与去噪方面表现优异。

3.3.2 对比实验

在 BIG 2015 和 Ember 2018 数据集上,本文将 DenseGRU-Net 模型与主流检测方法

DenseNet121-Ngram^[18]、DESMNet^[19]、J-ResNet^[20] 进行对比实验,以评估 DenseGRU-Net 在恶意代码检测任务中的有效性。由图 4 和表 2 可以看出,该模型在精确率、准确率、召回率以及 F1 分数等评价指标上均有小幅提升。特别是在处理恶意软件变体和复杂加密的样本时,DenseGRU-Net 模型表现出更强的鲁棒性和泛化能力。通过引入多通道特征融合和 GRU 动态记忆机制,模型能够动态调整各通道特征权重,更好地捕捉复杂的特征依赖关系,从而有效提升了恶意软件识别的准确性。模型在训练过程中展现了稳定的收敛性,损失率逐渐下降,最终模型在训练集和验证集上的准确率均达到了较高水平。结果

表明,DenseGRU-Net 模型在学习到数据特征后能够有效地进行分类,且通过精心设计的多通道共享记忆机制,成功避免了过拟合问题,提升了模型的泛化能力。此外,模型在处理类别不平衡的数据集时,依然能够保持较好的分类性能,这使得 DenseGRU-Net 模型在实际恶意软件检测场景中具有更广泛的应用潜力和适应性。

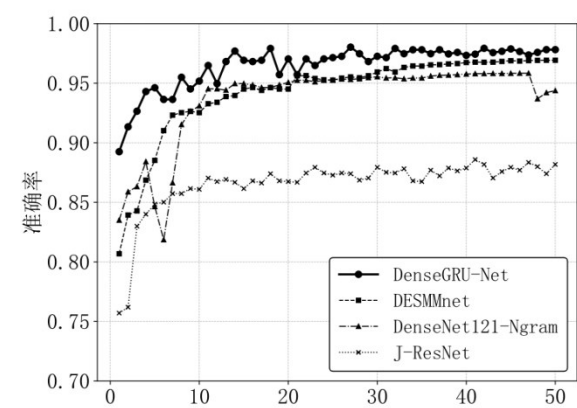


Fig. 4 Comparison of accuracy among four detection methods
图4 4种检测方法准确率比较

Table 2 Performance comparison of different networks
表2 不同网络性能比较

模型	Accuracy	Precision	Recall	F1-score
DenseGRU-Net	92.619	88.802	91.287	91.634
DenseNet121-Ngram	92.365	88.165	90.643	90.817
DESMMnet	90.719	87.701	90.813	90.230
J-ResNet	84.735	86.327	81.238	87.147

4 结语

本文提出一种基于多通道混合卷积记忆网络的恶意代码检测模型 DenseGRU-Net,旨在解决传统方法在面对恶意代码变体和复杂加密样本时鲁棒性不足的问题。通过引入 Shannon 交叉熵与软阈值函数的自适应去噪机制,有效地抑制了噪声干扰,同时保留了关键特征信息,提升了模型的抗干扰能力和特征提取精度。此外,模型在 DenseNet121 过渡层后引入 GRU 共享记忆模块,以增强多通道特征间的依赖建模能力。实验结果表明,DenseGRU-Net 模型在精确率、准确率、召回率和 F1-score 等指标上优于传统恶意代码检测模型和其他深度学习模型。特别是在处理复杂任务(如恶意软件变体和加密样本)时,模型表现出更强的鲁棒性和泛化能力,在复杂环境中的性能优势尤为显著。未来计划深入研究恶意代码特征的多样性,进一步探索多模态特征融合策略,使恶意代码检测更加稳定和高效。

参考文献:

[1] 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.

[2] ZHANG Y, SONG L, HE Y. A novel immune-inspired method for malicious code extraction and detection [C]//International Conference on Applied Informatics and Communication, 2011: 501-509.

[3] CHENG J, ZHENG J, YU X. An ensemble framework for interpretable malicious code detection [J]. International Journal of Intelligent Systems, 2022, 37(12): 10100-10117.

[4] LINDORFER M, DI F A, MAGGI F, et al. Lines of malicious code: insights into the malicious software industry [C]//Proceedings of the 28th Annual Computer Security Applications Conference, 2012: 349-358.

[5] YIN Z, YU X, NIU L. Malicious code detection based on software fingerprint [C]//The International Conference on Artificial Intelligence and Software Engineering, 2013: 212-216.

[6] IKRAM B A O, MOHAMMED B, ABDELHAKIM B A, et al. Machine learning application for malwares classification using visualization technique [C]//Proceedings of the 4th International Conference on Smart City Applications, 2019: 1-6.

[7] MANAVI F, HAMZEH A. A new method for malware detection using op-code visualization [C]//2017 Artificial Intelligence and Signal Processing Conference, 2017: 96-102.

[8] DARUS F M, SALLEH N A A, ARIFFIN A F M. Android malware detection using machine learning on image patterns [C]//2018 Cyber Resilience Conference, 2018: 1-2.

[9] SHENEAMER A, ALHAZMI E, HENRYDOSS J. Empirical analysis of learning-based malware detection methods using image visualization [J]. International Journal of Advanced Computer Science and Applications, 2022, 13(4): 925-936.

[10] MOHAMMED T M, NATARAJ L, CHIKKAGOUDAR S, et al. Malware detection using frequency domain-based image visualization and deep learning [DB/OL]. <https://arxiv.org/abs/2101.10578>.

[11] YANG Y, XU C, DONG F, et al. A new multi-scale convolutional model based on multiple attention for image classification [J]. Applied Sciences, 2019, 10(1): 101.

[12] YANG B, WANG L, WONG D, et al. Convolutional self-attention networks [DB/OL]. <https://arxiv.org/abs/1904.03107>.

[13] WU H, ZOU Z, GUI J, et al. Multi-grained attention networks for single image super-resolution [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 31(2): 512-522.

[14] BELLO I, ZOPH B, VASWANI A, et al. Attention augmented convolutional networks [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 3286-3295.

[15] WANG L, SHEN J, TANG E, et al. Multi-scale attention network for image super-resolution [J]. Journal of Visual Communication and Image Representation, 2021, 80: 103300.

[16] SHANNON C E. A mathematical theory of communication [J]. The Bell System Technical Journal, 1948, 27(3): 379-423.

[17] CHOI E, SCHUETZ A, STEWART W F, et al. Using recurrent neural network models for early detection of heart failure onset [J]. Journal of the American Medical Informatics Association, 2017, 24(2): 361-370.

[18] WANG J, CHEN Z, XIE X, et al. Deep visualization classification method for malicious code based on Ngram-TFIDF [J]. Journal on Communication/Tongxin Xuebao, 2024, 45(6): 160.

[19] ZHANG K, GUO Y, WANG X, et al. Multiple feature reweight densenet for image classification [J]. IEEE Access, 2019, 7: 9872-9880.

[20] ZHU Z, ZHAI W, LIU H, et al. Juggler-ResNet: a flexible and high-speed ResNet optimization method for intrusion detection system in software-defined industrial networks [J]. IEEE Transactions on Industrial Informatics, 2021, 18(6): 4224-4233.

(责任编辑:黄健)