文章编号：2096-1618(2024)05-0546-07

基于改进的 MobileNetV2 模型的安卓恶意家族分类方法研究

李久玲， 甘 刚
(成都信息工程大学网络空间安全学院，四川 成都 610225)

摘要：针对人类视觉系统对颜色的高度敏感特点，提出一种基于改进的 MobileNetV2 模型的安卓恶意家族分类方法。该方法通过引入注意力机制，对 RGB 图像的 3 个通道进行特征融合，提高模型对图像颜色信息的敏感度。同时，针对小样本数据集的问题，提出一种改进的模块结构，减少模型的深度和宽度，以提高模型对小样本数据集的特征提取能力。将 SE(squeeze-and-excitation network)注意力机制与 CBAM(convolution block attention module)注意力机制融入模型进行对比，实验结果表明：CBAM 注意力机制在该图像分类任务中表现出显著的优越性，准确率达到94.18%，比原有模型提高了3.16%，验证了该方法的有效性和实用性。该研究对于小样本数据集的图像分类任务的准确性和实际应用中的性能有重要意义。

关 键 词：MobileNetV2；RGB 图像；注意力机制；安卓恶意家族分类
中图分类号：TP309.2 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2024.05.005

0 引言

作为全球最流行的移动操作系统之一，安卓系统已经成为恶意软件攻击的主要目标之一。近年来，由于安卓系统上的恶意应用数量和种类不断增长，恶意软件攻击已成为安卓用户面临的主要威胁之一。随着恶意软件攻击手段的不断升级，传统的恶意软件检测方法已经不能满足需求。为应对这一挑战，研究人员不断开发更有效的恶意软件家族分类和检测方法。

在安卓系统上，恶意软件的检测方法通常可以分为动态分析和静态分析两种。动态分析方法通过模拟恶意软件在真实系统中的行为，以检测其是否具有恶意行为。静态分析方法则主要通过对恶意软件的代码进行分析，以检测其是否包含恶意代码。动态分析方法可以检测到恶意软件的行为，但是需要在真实系统上进行模拟，可能会对系统造成安全隐患。静态分析方法可以在不运行恶意软件的情况下进行检测，但是可能会受到代码混淆等技术的影响。相较于传统的动态和静态分析方法，图像分类方法具有更高的准确性和可靠性。

通过将恶意软件样本转换为图像，然后使用深度学习模型对这些图像进行分类，可以在一定程度上避免代码混淆等技术的干扰，并提高检测准确率。同时，图像分类方法还可以提高模型对图像颜色信息的敏感度，从而更加有效地检测恶意软件。因此，图像分类技术在 Android 恶意软件检测领域具有广泛的应用前景。

目前，大部分 Android 恶意软件图像化[1-2]，是将 APK 文件转换成灰度图或 RGB 图像，使用的数据集大部分是 Maling 数据集[3] 和 Drebin[4] 数据集。Maling 数据集和 Drebin 数据集是较早时期构建的，数据集中包含的恶意软件和恶意行为种类比较有限，难以覆盖当前恶意软件的全部类型和变种。此外，恶意代码图像对神经网络模型提出了更高的要求。

随着基于图像的二进制表现的兴起，计算机视觉在自动恶意软件检测中发挥着越来越重要的作用。二进制图像生成速度快，不需要特征工程，并且对流行的模糊处理方法具有弹性。

基于以上论述，本文提出一种基于改进的 MobileNetV2 模型的恶意家族分类方法，主要工作如下：

(1)使用最大公共安全图像数据库 MALNET-IMAGE[5]，比现有数据库多提供 24 倍的图像和 70 倍的类，由 AndroZoo[6] 提供 1262024 个 Android 的 APK 文件。MALNET-IMAGE 包含 47 种类型和 696 个家族的 120 多万个恶意软件图像。本文使用 MALNET-IMAGE-TINY 数据集。

(2)将注意力机制融合进 MobileNetV2 模型中，提出结合注意力机制的 MobileNetV2 模型对恶意家族进行分类检测。实验表明，改进后的 MobileNetV2 模型在恶意家族分类中表现更优。

收稿日期：2023-05-29
基金项目：四川省科技计划资助项目(2023YFG0292，2021ZDY0011)；四川省社科基金资助项目(SC21B034)

(3)改进后的模型对恶意软件家族小样本数据集检测结果有明显的提升。

1　相关工作

在恶意软件可视化之前,静态分析是常用的方法。Pan 等[7]根据应用程序的特点,将 Android 恶意程序检测中的静态分析分为 4 类:基于 Android 特征的方法、基于操作码的方法、基于程序图的方法和基于符号执行的方法。Mat 等[8]提出一种基于权限特征的 Android 恶意软件检测系统,该系统使用贝叶斯分类。权限特征采用静态分析技术提取,使用两种特征选择算法信息增益和卡方进行实验。权限特征检测的最佳准确率为91.1%。Raghav 等[9]提出利用静态分析和自然语言处理技术的文件嵌入生成特征向量,然后这些嵌入的文件被用来训练二进制分类器,这些分类器可以有效地区分良性和恶意的安卓应用。静态分析检测快速、高效,但由于不能真实地模拟动态运行的程序,所以存在误报的可能。对存在代码混淆和多态变形的恶意软件也难以检测,会导致整个检测的效果差或者失效。

为解决代码混淆对恶意软件检测的影响,采用动态分析的方法也很多。Alzaylaee 等[10]提出一个深度学习系统 DL-Droid,通过使用有状态输入生成的动态分析检测 Android 恶意软件。Thangavelooa 等[11]提出一种用于 Android 恶意软件检测的动态分析技术——DATDroid。该方法分 3 个阶段,即特征提取、特征选择和特征分类,通过提取 5 个特征:系统调用、系统调用过程的错误和事件、CPU 使用率、内存和网络数据包来分析恶意软件。Sihag 等[12]提出一种基于深度学习的动态特征的 Android 恶意软件检测方法,利用仿真环境中执行的应用程序来动态分析行为特征。首先输入 APK 生成动态分析日志,然后进行特征处理。通过特征处理生成特征向量,将特征向量输入深度学习模型进行训练。YANG 等[13]提出一种基于权限补充和 API(application programming interface)调用的 Android 恶意软件检测方法。首先,定义权限补码来描述使用动态代码加载隐藏恶意行为的恶意软件的行为,并定义 3 种类型的攻击模式。然后,提出一种新的特征选择方法,通过基于恶意应用中的特征数量与良性应用中特征数量之间的差异来计算每个特征的权重方面与其他方法不同。最后,使用特征选择方法提出了一种基于权限补充和 API 调用的 Android 恶意软件检测方法,以应对通过动态代码加载隐藏恶意行为所带来的挑战。Cai 等[14]介绍了一种新的动态应用分类技术 DroidCat。通过使用基于方法调用和 ICC(inter-component communication)意图的多种动态特性,不涉及权限、应用程序资源或系统调用,同时完全处理反射。Zhang 等[15]提出一种新的特征工程方法和一种用于恶意软件检测的新的深度学习框架。特征工程方法利用哈希方法分别提取异构特征,从 API 名称、类别和参数中提取的特征进一步连接并反馈到深度学习模型中。使用多个门控 CNN(convolutional neural network)模型从每个 API 调用的高维哈希特征中学习抽象的低维特征。门控 CNN 模型的输出由双向 LSTM(long short term memory networks)处理,以提取所有 API 调用的顺序相关性。

基于动静态混合特征的恶意软件检测方法则是将静态特征和动态特征混合来检测恶意软件,比单一使用一种特征进行检测的效率更高,还能检测出使用单一分析方法检测不出的恶意软件,在时间和效率上都有较大改善。Surendran 等[16]将静态和动态机制的优势结合,提出一种新的基于 TAN(树状增强的朴素贝叶斯)的混合恶意软件检测方法。采用相关的静态和动态特征(API 调用、权限和系统调用)之间的条件依赖,训练 3 个正则化逻辑回归分类器,分别对应应用程序的 API 调用、权限和系统调用,并将其输出关系建模为 TAN,用于识别应用程序是否为恶意的。混合分析对特征提取的过程比较全面,但是特征提取过程复杂,静态特征和动态特征都要提取,检测难度加大,也将耗费更多的时间和资源。

Rong 等[17]提出一个基于深度迁移学习的框架 TransNet,用于检测未见过的恶意软件变体。首先,通过数据预处理将包含 OSI 模型各层数据的会话所代表的原始流量转换成固定大小的 RGB 图像。之后,基于在 ImageNet 上预训练的 ResNet-50 模型,用可转移归一化取代批量归一化作为归一化层,构建深度转移学习模型。Nahmias 等[18]介绍了 TrustSign,这是一种新颖的、可信赖的自动恶意软件签名生成方法,它基于在 ImageNet 数据集上预训练的 VGG-19 神经网络模型转移的高级深度特征,通过基于易失性内存中恶意进程的存在来产生签名。Prima 等[19]提出一种基于 CNN 的恶意软件分类架构。恶意二进制文件表示为灰度图像,通过冻结 ImageNet 数据集上预先训练的 VGG16 层并使最后一个完全连接的层适应恶意软件家族分类来训练深度神经网络。Khetarpal 等[20]将恶意软件的可执行文件转换为其可视化表示,并获得恶意文件的灰度图像。然后,灰度图像通过深度卷积神经网络将恶意文件分类到各自的恶意软件系列中。Jiang 等[21]

针对现有系统不能有效地检测出混淆的 Android 恶意软件,提出一种基于 TFIDF(term frequency-inverse document frequency)算法的有效方法来识别独特的操作码序列,将识别被混淆的恶意软件的问题减少为将两个图像相互转换的问题。陈小寒[22]针对恶意软件家族分类问题,将恶意软件操作码转换为灰度图像,利用 SimHash 对原始编码与递归神经网络的预测编码融合,生成特征图像,使用卷积神经网络对特征图像进行分类。于兴崭等[23]提出将安卓的 3 个特征文件转换成 RGB 图,将 Xception 模型中融合注意力机制,实现安卓恶意家族分类。

基于图像的恶意软件检测模型不需要执行软件也不需要反汇编文件,能够节省大量的时间和成本,比静态动态分析劳动密集度和耗时更少,且能获得较高的检测精度。

RGB 图比灰度图包含的信息量更多,将特征文件转换成 RGB 图并通过 CNN 提取特征并分类,检测效果更优。使用 CNN 进行特征提取和分类,通过对模型的设计和调参,以达到更好的检测效果。

2 数据集及分类模型

本文使用最大的图像数据集,并提出一种基于改进的 MobileNetV2 模型的安卓恶意家族分类方法。同时,在 MobileNetV2 模型中加入 CBAM 注意力机制,以提高模型的分类准确率。

2.1 图像数据集

MALNET-IMAGE 是一个层次化的图像和图形数据库,包括 1262024 张图片和图表,横跨 696 个家族和 47 种恶意软件。该图像数据集是从每个 Android APK 中提取 DEX 字节码,然后将提取的 DEX 文件转换为 8 位无符号整数的 1D 数组。阵列中的每个条目都在[0-255],其中 0 对应于黑色像素,255 对应于白色像素。然后,使用标准线性绘图将每个 1D 字节数组转换为 2D 数据,其中图像的宽度是固定的。并使用 Pillow 库中的标准 Lanczos 滤波器将每个图像缩放到 256×256。最后,根据每个字节的用途对其进行着色,在原始字节码的顶部添加一层语义信息。根据其在 DEX 文件结构中的位置,将每个字节分配给特定的 RGB 通道、标题、标识符和类定义以及数据。MALNET-IMAGE-TINY 数据集,包含 61201 个训练图像,8743 个验证图像和 17486 个测试图像。本文数据集使用 Malnet-IMAGE-TINY 数据集,选取 10 个类别进行实验,malnet-image-tiny-1 的类别和数量如图 1 所示。

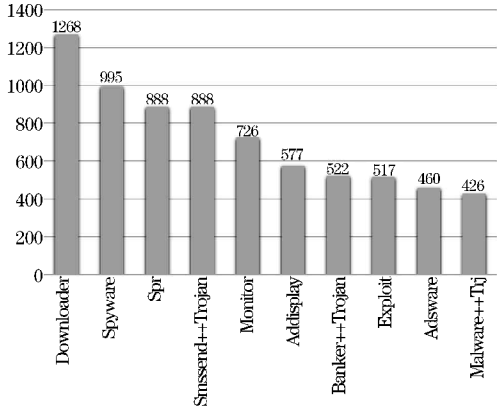

图 1 malnet-image-tiny-1 的类别和数量

2.2 分类模型

2.2.1 MobileNetV2 模型

MobileNetV2[24]是谷歌 2018 年提出的一种轻量级卷积神经网络模型,它是 MobileNet 系列的一个新版本。MobileNetV2 旨在提供一个高效的模型,可以在计算资源有限的移动设备上快速运行,同时还具有较高的精度。相比于 MobileNetV1,MobileNetV2 在保持同样精度的情况下,参数量减少一半,计算量也减少约 30%。MobileNetV2 主要由两个部分组成:特征提取层和分类器。特征提取层由一系列的深度可分离卷积层和倒残差块(inverted residual block)组成。在这个网络中,深度可分离卷积层代替了传统卷积层,以减少模型的参数量和计算量。倒残差块则是 MobileNetV2 的核心设计,它使用一种新的结构来加深网络,同时保持计算量不变。分类器部分由全局平均池化层、全连接层和 softmax 层组成,用于输出分类结果。MobileNetV2 具有高效、轻量级和高精度的特点,因此被广泛应用于移动端的视觉任务。MobileNetV2 网络模型结构如表 1 所示。

表 1 MobileNetV2 网络模型结构

Input	Operator	t	c	n	s
$224^2\times3$	Conv2d	–	32	1	2
$112^2\times32$	bottleneck	1	16	1	1
$112^2\times16$	bottleneck	6	24	2	2
$56^2\times24$	bottleneck	6	32	3	2
$28^2\times32$	bottleneck	6	64	4	2
$14^2\times64$	bottleneck	6	96	3	1
$14^2\times96$	bottleneck	6	160	3	2
$7^2\times160$	bottleneck	6	320	1	1
$7^2\times320$	conv2d 1×1	–	1 280	1	1
$7^2\times1280$	avgpool 7×7	–	–	1	–
$1^2\times1280$	conv2d7×7	–	k	–	–

其中 t 为瓶颈层升维的倍数,c 为特征的维数,n 为该瓶颈层重复的次数,s 为瓶颈层第一个 conv 的步幅。

2.2.2　注意力机制模块

SE 是一种用于深度卷积神经网络中的注意力机制模块,由 Hu Jie 等[25]2018 年提出,旨在解决现有卷积神经网络中特征图中的信息冗余和不必要的噪声等问题。SE 模块的核心思想是通过学习特征通道之间的关系来增强网络的表征能力。具体地,SE 模块在每个特征图中进行两个操作:压缩和激励。通过压缩操作,SE 模块将每个通道的特征图压缩成一个标量,然后通过激励操作,将这个标量作为权重对该通道进行加权,使网络能够更加关注重要的特征通道。压缩操作使用全局平均池化层来计算每个通道的特征图的平均值,然后使用两个全连接层来将平均值映射到一个标量。激励操作使用 sigmoid 函数将该标量作为权重对该通道进行加权,从而生成一个新的特征图。通过引入 SE 模块,网络能够更加关注重要的特征通道,从而提高网络的表现力和泛化能力。激励和压缩模块如图 2 所示。

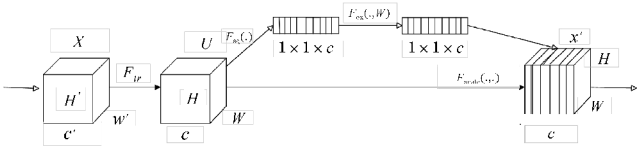

图 2　SE 的激励和压缩模块

CBAM 是一种用于增强卷积神经网络特征表示能力的注意力机制,由 Woo Sanghyun 等[26]2018 年提出。CBAM 方法通过引入一种“通道注意力”和“空间注意力”的方式,自适应地调整不同通道和不同空间位置的特征图的权重,从而提高模型的精度。CBAM 方法的核心思想是:在卷积神经网络中,不同通道和不同空间位置的特征图所包含的信息是不同的,有些通道或位置可能比其他通道或位置更重要。因此,可以通过自适应地调整不同通道和不同空间位置的特征图的权重,来增强模型的特征表示能力。具体来说,CBAM 方法通过引入一个称为“convolutional block attention module”的结构,对每个通道和空间位置进行加权。该模块包括两个步骤:通道注意力和空间注意力。在通道注意力步骤中,CBAM 方法使用一个全局平均池化层和两个全连接层来学习每个通道的权重。在空间注意力步骤中,CBAM 方法使用一个反卷积层和一个全连接层来学习每个空间位置的权重。通过这种方式,CBAM 方法可以自适应地调整每个通道和不同空间位置的重要性,从而提高模型的特征表示能力。CBAM 结构如图 3 所示。

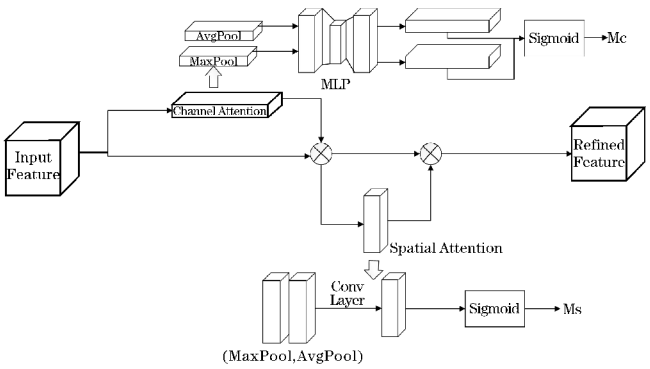

图 3　CBAM 通道注意力和空间注意力结构

2.2.3　融合注意力机制的 MobileNetV2 模型

将 MobileNetV2 模型与注意力机制融合,生成新的模型,对恶意软件家族进行分类。新的模型结构如图 4 所示。

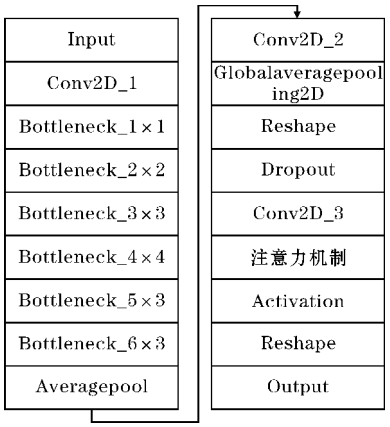

图 4　改进后的 MobileNetV2 模型

恶意软件家族分类整体流程如图 5 所示。

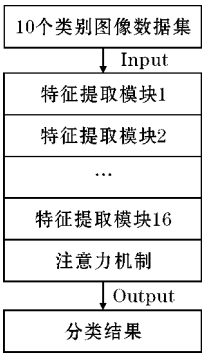

图 5　恶意软件家族分类整体流程

3　实验结果

3.1　实验环境及参数

实验设备配置如下:cuda12.1,PyCharm,python3.10,PTX2080,实验中模型使用以改进的 MobileNetV2 为骨

干。数据集使用 MALNET-IMAGE-TINY 数据集。实验参数设置如下:50 个 epoch,batch_size=16,dropout=0.3。

3.2 损失函数

本文使用交叉熵损失函数。交叉熵损失函数在训练时更加稳定,避免了梯度消失或爆炸的问题。在反向传播时的计算量较小,可以很快地进行模型训练,对噪声和异常值的鲁棒性较好。损失函数公式如下:

$$L=\frac{1}{N}\sum_i L_i=-\frac{1}{N}\sum_i\sum_{c=1}^{M}y_{ic}\log(p_{ic})$$

其中,M 为类别的数量,y_{ic} 为符号函数等于 0 或 1,如果样本 i 的真实类别等于 c 取 1,否则取 0。p_{ic} 为观测样本 i 属于类别 c 的预测概率。使用交叉熵损失函数主要是用来解决数据集不均衡的问题。

3.3 评价指标

本文选择准确率(Accuracy)、精确率(Precision)、召回率(Recall)和 F1 值(F1-score)作为评价指标来评价基本模型,改进后的模型则使用准确率进行评价,计算公式如下:

$$\text{Accuracy}=\text{TP+TN/TP+TN+FN+FP}$$

$$\text{Precision}=\text{TP/TP+FP}$$

$$\text{Recall}=\text{TP/TP+FN}$$

$$\text{F1}=2\text{TP/2TP+FP+FN}$$

TP、FP、TN、FN 为分别为所有类别的真阳率、假阳率、真阴率和假阴率。

3.4 实验分析

3.4.1 基本模型对比

使用 malnet-image-tiny-1 数据集在以下 4 种不同模型中进行测试,不同模型的分类准确率、精确率、召回率和 F1 分数如表 2 所示。从表 2 中可以看出,MobileNetV2 模型的准确率、精确率、召回率和 F1 分数最高。

表 2 不同模型性能对比 单位:%

Model	Accuracy	Precision	Recall	F1
EfficientNetB0	80.55	79.36	80.55	79.23
EfficientNetB2	81.93	81.75	81.93	81.52
VGG19	87.96	88.05	87.96	87.86
MobileNetV2	91.51	92.07	91.51	91.60

3.4.2 改进模型

使用 MobileNetV2 基本模型对 malnet-image-tiny-1 数据集进行检测,添加 dropout 层并选择不同优化器的结果如表 3 所示。

表 3 MobileNetV2 模型不同优化器的准确率对比 单位:%

Adam	SGD
91.02	89.14

将 SE 注意力机制和 CBAM 注意力机制插入不同层进行对比,以及选择不同的优化器之间的对比,结果如表 4、表 5 所示。

表 4 SE 注意力机制插入不同位置不同优化器的准确率对比 单位:%

插入位置	Adam	SGD
输入层	91.02	88.45
中间层	92.20	88.85
输出层	93.29	89.63

表 5 CBAM 注意力机制插入不同位置不同优化器的准确率对比 单位:%

插入位置	Adam	SGD
输入层	91.71	87.36
中间层	90.72	88.85
输出层	94.18	89.93

可以看出,使用 CBAM 注意力机制插入输出层,选择 Adam 优化器,检测的准确率更高,达到94.18%,相比原本的 MobileNetV2 模型,提高了3.16%。使用 SE 注意力机制,准确率提高了2.27%。

改进后的模型提高了对恶意软件家族分类的准确率,特别是针对小样本数据集,准确率有明显的提升。

4 结论与讨论

本文通过将 CBAM 注意力机制融入 MobileNetV2 模型,并与 SE 注意力机制进行对比,针对最大的图像数据库 MALNET-IMAGE 下的 MALNET-IMAGE-TINY 数据集进行实验。结论如下:

(1)改进后的模型能够显著提高恶意软件小样本家族的分类准确率。实验中恶意软件家族样本数量不多,但分类准确率有明显的提升。

(2)实验中,SE 注意力机制和 CBAM 注意力机制加入到模型中,都能提升模型的分类准确率,最佳的方式是将 CBAM 注意力机制插入到 MobileNetV2 模型的输出层,同时选用 Adam 分类器,此时对恶意家族分类的准确率最高,达到了94.18%,比原有模型提高了3.16%。

(3)原数据集较大,无法一次性进行实验,因此本文只能抽取一些恶意家族类别进行一组实验。为进一步提高模型的检测能力,未来将继续优化模型,降低复杂度,并使用更多的数据集进行实验。

致谢:感谢成都信息工程大学创新创业训练计划项目(202110621224、202110621225)对本文的资助

参考文献:

[1] Bakour K, Ünver H M. VisDroid: Android malware classification based on local and global image features, bag of visual words and machine learning techniques[J]. Neural Computing and Applications, 2021, 33: 3133-3153.

[2] Rahali A, Lashkari A H, Kaur G, et al. DIDroid: Android malware classification and characterization using deep image learning[C]. The 10th International Conference on Communication and Network Security. 2020: 70-82.

[3] 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.

[4] 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.

[5] Freitas S, Duggal R, Chau D H. MalNet: A large-scale image database of malicious software [J]. arXiv preprint arXiv: 2102. 01072, 2021.

[6] Allix K, Bissyandé T F, Klein J, et al. Androzoo: Collecting millions of android apps for the research community[C]. Proceedings of the 13th International Conference on Mining Software Repositories. 2016: 468-471.

[7] Pan Y, Ge X, Fang C, et al. A systematic literature review of android malware detection using static analysis[J]. IEEE Access, 2020, 8: 116363-116379.

[8] Mat S R T, Ab Razak M F, Kahar M N M, et al. A Bayesian probability model for Android malware detection[J]. ICT Express, 2022, 8(3): 424-431.

[9] Raghav U, Martinez-Marroquin E, Ma W. Static analysis for Android malware detection with document vectors [C]. 2021 International Conference on Data Mining Workshops (ICDMW). IEEE, 2021: 805-812.

[10] Alzaylaee M K, Yerima S Y, Sezer S. DL-Droid: Deep learning based android malware detection using real devices [J]. Computers & Security, 2020, 89: 101663.

[11] Thangavelooa R, Jinga W W, Lenga C K, et al. Datdroid: Dynamic analysis technique in android malware detection [J]. International Journal on Advanced Science, Engineering and Information Technology, 2020, 10(2): 536-541.

[12] Sihag V, Vardhan M, Singh P, et al. De-LADY: Deep learning based Android malware detection using dynamic features[J]. J. Internet Serv. Inf. Secur., 2021, 11(2): 34-45.

[13] YANG Jiyun, TANG Jiang, YAN Ran, et al. Android malware detection method based on permission complement and API calls[J]. Chinese Journal of Electronics, 2022, 31(4): 773-785.

[14] Cai H, Meng N, Ryder B, et al. Droidcat: Effective android malware detection and categorization via app-level profiling[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(6): 1455-1470.

[15] Zhang Z, Qi P, Wang W. Dynamic malware analysis with feature engineering and feature learning [C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(1): 1210-1217.

[16] Surendran R, Thomas T, Emmanuel S. A TAN based hybrid model for android malware detection [J]. Journal of Information Security and Applications, 2020, 54: 102483.

[17] Rong C, Gou G, Cui M, et al. TransNet: Unseen malware variants detection using deep transfer learning[C]. International Conference on Security and Privacy in Communication Systems. Springer, Cham, 2020: 84-101.

[18] Nahmias D, Cohen A, Nissim N, et al. Deep feature transfer learning for trusted and automated malware signature generation in private cloud environments [J]. Neural Networks, 2020, 124: 243-257.

[19] Prima B, Bouhorma M. Using transfer learning for malware classification [J]. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020, 44: 343-349.

[20] Khetarpal A, Mallik A. Visual malware classification using transfer learning [C]. 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2021: 1-5.

[21] Jiang Y, Li R, Tang J, et al. AOMDroid: Detecting obfuscation variants of Android malware using transfer learning[C]. International Conference on Security and Privacy in Communication Systems. Springer, Cham, 2020: 242-253.
[22] 陈小寒. 基于深度学习的恶意软件可视化分类技术研究[D]. 长沙: 湖南师范大学, 2021.
[23] 于兴崭, 芦天亮, 杜彦辉, 等. 基于合成图像和 Xception 改进模型的安卓恶意家族分类方法[J]. 计算机科学, 2023, 50(4): 351-358.
[24] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4510-4520.
[25] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 7132-7141.
[26] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European Conference on Computer Vision (ECCV). 2018: 3-19.

Research on Classification Method of Android Malware Family based on Improved MobileNetV2 Model

LI Jiuling, GAN Gang
(1. College of Cyber Space Security Academy, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Aiming at the highly sensitive characteristics of the human visual system to color, a classification method of the Android malicious family based on the improved MobileNetV2 model is proposed. By introducing the attention mechanism, this method performs feature fusion on the three channels of RGB image to improve the sensitivity of the model to the color information of the image. At the same time, aiming at the problem of small-sample datasets, an improved module structure is proposed, which reduces the depth and width of the model and improves the feature extraction ability of the model for small-sample datasets. The experimental results show that the Sequeeze-and-Excitation Network(SE) and the Convolution Block Attention Module(CBAM) are both located in the model. The experimental results show, the CBAM attention mechanism shows significant superiority in this image classification task, with an accuracy rate of 94.18%, which is 3.16% higher than the original model, which verifies the effectiveness and practicality of the proposed method. This study has important implications for the accuracy of image classification tasks and the performance in practical applications of small-sample datasets.

Keywords: MobileNetV2; RGB image; attention mechanism; Android malware family classification