一种基于特征融合的恶意代码快速检测方法

王 硕,王 坚,王亚男,宋亚飞
(空军工程大学防空反导学院,陕西西安 710051)

摘 要: 随着恶意代码对抗技术的发展,恶意攻击者通过加壳、代码混淆等技术繁衍大量恶意代码变种,而传统恶意代码检测方法难以对其进行有效检测. 基于恶意代码可视化的恶意代码检测方法被证明是一种能够有效识别恶意代码及其变种的新方法. 针对目前研究仅着眼于提升模型分类准确率而忽略了恶意代码检测的时效性,本文提出了一种基于特征融合的恶意代码快速检测方法. 该方法以深度神经网络为框架,采取模块化设计思想,将多尺度恶意代码特征融合与通道注意力机制结合,增强关键特征表达,并使用数据增强技术改善数据集类别不平衡问题. 通过实验证明本文方法分类准确率高且参数量小、检测时效性高,优于目前的恶意代码检测技术.

关键词: 恶意代码;深度神经网络;特征融合;通道注意力机制;数据增强技术;恶意代码可视化
基金项目: 国家自然科学基金(No.61703426)
中图分类号: TP309.5 **文献标识码:** A **文章编号:** 0372-2112(2023)01-0057-10
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20211701

A Fast Malicious Code Detection Method Based on Feature Fusion

WANG Shuo, WANG Jian, WANG Ya-nan, SONG Ya-fei
(*Air Defense and Antimissile School*, *Air Force Engineering University*, *Xi'an*, *Shaanxi* 710051, *China*)

Abstract: With the development of anti-malicious code technology, malicious attackers multiply a large number of malicious code variants by adding shell, code obfuscation and other technologies. However, traditional malicious code detection methods are difficult to detect them effectively. Malicious code detection based on malicious code visualization has been proved to be an effective method for identifying malicious code variants. The current research only focuses on improving the accuracy of model classification while ignoring the timeliness of malicious code detection. To solve the above problem, this paper proposes a fast malicious code detection method based on feature fusion. Based on the framework of deep neural network and the idea of modular design, our method combines multi-scale malicious code feature fusion with channel attention mechanism to enhance typical feature expression. In addition, data augmentation technology is utilized to deal with the problem of dataset category imbalance. The results of experiments indicate that the proposed method achieves high classification accuracy, small number of parameters and high detection timeliness, which is superior to the current malicious code detection technology.

Key words: malicious code; deep neural network; feature fusion; channel attention mechanism; data augmentation technology; malicious code visualization

Foundation Item(s): National Natural Science Foundation of China (No.61703426)

1 引言

恶意代码是指经过人为设计执行恶意行为或攻击的软件. 据2021年国家互联网应急中心发布的第2期周报统计[1],在1月4日至10日仅一周的时间,境内被感染网络病毒的主机数量约为67万个,境内计算机恶意程序传播次数高达4 009.3万. 大量的恶意代码不仅对用户日常生活产生严重影响,甚至影响了国家网络的安全,阻碍网络命运共同体的构建.

恶意代码分析技术按照是否执行文件分为动态分析技术和静态分析技术. 动态分析是指在沙箱、模拟器和虚拟机中运行可执行文件并通过系统调用监视、分析应用程序行为的实践. 静态分析方法提取恶意代码的静态特征来识别样本的不法行为. 静态分析方法在速度和有效性方面优于动态分析,因为它可以捕获与结构特性相关的信息[2,3]. 传统的恶意代码检测方法采取基于特征码的模板匹配思想,它需要研究员根据专家知识手工提取恶意代码的特征码,并将其与数据库

收稿日期:2021-12-24;修回日期:2022-07-31;责任编辑:李勇锋

中已知特征码进行逐一比对. 随着恶意代码混淆、加壳等技术的发展,恶意代码繁衍出大量变种. 而传统检测方法效率较低,并且难以有效地检测与识别恶意代码的变种. 因此,如何准确、高效地对恶意代码及其变种进行检测、分类成为了该领域的研究热点.

为了解决传统恶意代码检测方法面临的困境,更有效地检测经过加壳、混淆后的恶意代码变种,基于可视化的恶意代码检测方法应运而生[4~6]. 该方法先将恶意代码映射为图像,根据同一恶意家族中的图像纹理特征具有相似性,不同恶意家族中的图像纹理特征具有差异性的特点,提取恶意代码图像的纹理特征并进行分类. 该方法被证明能够有效地检测恶意代码变种,并且检测速度相较于动态检测技术快4 000倍[7]. 从该方法提出以来,大量专家学者对此展开了研究[8,9]. Nataraj等[10]融合图像和信号特征来描述恶意代码,并使用KNN(K-Nearest Neighbor)作为分类器来识别恶意代码. Kancherla等[11]为了增强特征的多样性将Gabor特征、小波特征和强度特征融合作为总特征,并训练SVM(Support Vector Machines)分类器实现恶意代码分类. 刘亚姝等[12]通过融合恶意图像的GIST特征与LBP(Local Binary Pattern)特征构建抗混淆特征,以解决模型在相似恶意图像中的分类性能下降的问题. Naeem等[13]为了减少计算时间,提出了一种融合恶意代码图像局部特征和全局特征的LGMP特征描述子. 卢喜东等[14]使用HOG特征作为恶意图像的分类依据,最后使用随机森林分类器对恶意代码及其变种进行检测与分类. 上述研究将机器学习应用于基于可视化的恶意代码检测方法,这些方法的特征提取与分类是分开进行的. 恶意图像纹理特征的提取需要依靠手工方式,而手工提取特征的方式需要消耗大量的计算资源,导致该方法效率较低,并且检测的精度仍有待提升.

深度学习在图像分类任务中取得了瞩目的成绩,将深度学习与恶意代码可视化相结合是一种有效提升恶意代码分类准确率的方式. Gibert等[15]分析了手工特征提取的特点与不足,设计了一种深度神经网络结构用于提取恶意图像的特征,在多个恶意代码数据集上均取得了良好的分类性能,具有较好的泛化能力. Danish等[16]将恶意代码映射为彩色图像,丰富了恶意图像中的信息,并将迁移学习应用于恶意代码检测任务,实验结果表明该方法在分类准确率方面表现卓越. Kabanga等[17]设计了一个由三个卷积层和两个全连接层组成的卷积神经网络(Convolutional Neural Network,CNN)框架用于识别恶意代码,取得了较好的性能. 崔志华等[18,19]针对恶意代码数据集中样本不平衡问题,提出了一种使用群智能算法优化深度神经网络模型的最优样本类别输入比例的解决方案.

上述基于恶意代码可视化的方法能够实现恶意代码变种的检测与分类,在一定程度上解决了代码混淆问题. 但是这些方法仅将注意力聚焦于提升分类准确率,而忽略恶意代码分类模型其他的性能指标,如检测时间、模型体积大小等. 针对这个问题,本文提出了一种基于特征融合的恶意代码快速检测方法. 该方法旨在提升分类准确率的同时缩短识别时间. 首先,该方法将恶意代码映射为灰度图像并通过双线性插值算法对恶意图像进行尺寸归一化. 然后,使用数据增强技术解决恶意代码数据集不平衡问题. 其次,融合在不同尺寸卷积核中提取的多尺度特征来增加特征的多样性,并结合通道注意力机制增强恶意图像关键特征表达. 最后,训练深度神经网路模型实现对恶意代码变种的分类.

本文的工作主要包括以下3点. (1)提出了一种基于特征融合的深度神经网络模型来检测和分类恶意代码及其变种. 该模型结合了多尺度特征融合与通道注意力机制,具有良好的纹理特征提取能力、参数量小,在提升分类准确率的同时能够快速检测未知的新样本. (2)提出了使用数据增强技术来解决数据不平衡问题. 通过对图像的变换实现少样本的过采样,均衡数据集,提升模型性能. (3)在DataCon数据集和Malimg数据集上通过大量实验证明本方法在恶意代码变种检测和分类中的卓越性能,并分析了性能提升的原因.

2 模型概述

本文提出的恶意代码检测方法包含两个部分:数据预处理和FFSE模型构建. 其中,数据预处理包括恶意代码可视化、图像尺寸归一化以及数据增强技术. 该方法的结构如图1所示.

2.1 数据预处理

2.1.1 恶意代码可视化

恶意代码可视化是将恶意代码二进制文件转化为灰度图像的过程,其流程如图2所示. 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,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,部分转化后的恶意家族样本如图3所示.

2.1.2 图像尺寸归一化

在卷积神经网络中,由于全连接层的权值矩阵大小是固定的,即输入到全连接层的特征尺寸必须保持一致. 如果输入图片的尺寸不同,那么经过卷积和池化操作后的特征尺寸也会产生差异,即输入全连接层的

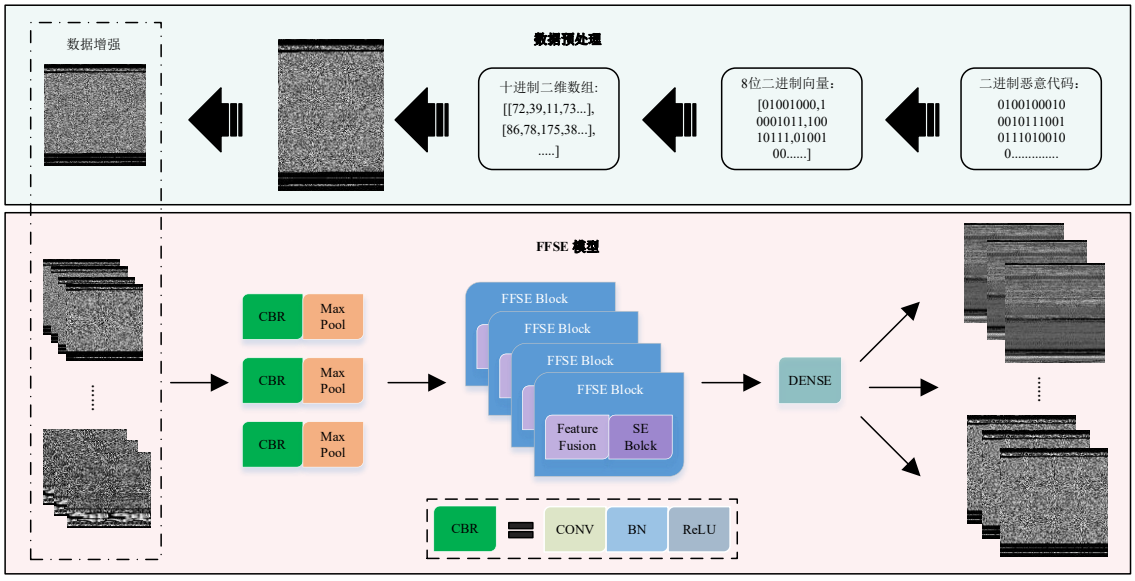


图 1　模型结构示意图

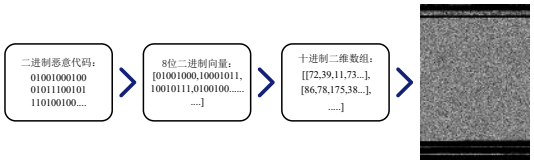


图 2　恶意代码可视化流程图

表 1　行宽与恶意文件大小对应关系

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

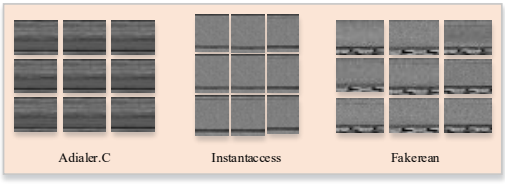


图 3　不同恶意家族可视化后的恶意图像

特征尺寸不同,这导致全连接层失效. 因此,输入卷积神经网络的图片必须为同一尺寸. 但是,可视化后的恶意图像尺寸均不相同. 因此,需要对可视化后的恶意图像进行尺寸归一化.

为了使经过归一化后的恶意图像尺可能保持原有的纹理特征不变,本文采用双线性插值算法对图像尺寸进行归一化. 该算法首先选取与恶意图像插值点直接相邻的4个像素点,然后先在X方向上进行两次线性插值运算;最后在Y方向上进行线性插值计算得到插值点的像素:

$$
\begin{aligned}
f(x,y_1)&=\frac{x_2-x}{x_2-x_1}f(x_1,y_1)+\frac{x-x_1}{x_2-x_1}f(x_2,y_1)\\
f(x,y_2)&=\frac{x_2-x}{x_2-x_1}f(x_1,y_2)+\frac{x-x_1}{x_2-x_1}f(x_2,y_2)\\
f(x,y)&=\frac{y_2-y}{y_2-y_1}f(x,y_1)+\frac{y-y_1}{y_2-y_1}f(x_2,y_2)
\end{aligned}
\tag{1}
$$

其中,$f(x,y)$是恶意图像中插值点的像素值,$(x_i,y_j)(i,j=1,2)$是恶意图像插值点附近的4个像素. 图4为Allaple.A家族中某样本经过归一化后的恶意图像,通过观察可以看出经过双线性插值算法后的恶意图像的基本纹理特征得到良好的保留.

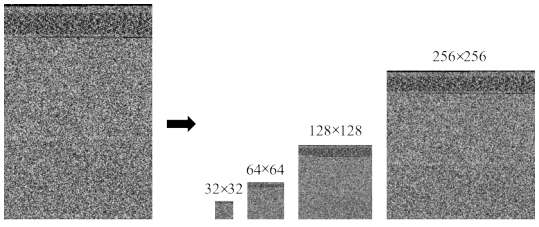


图 4　双线性插值法放缩恶意代码图像

2.1.3　数据增强技术

在深度学习模型中,分类的效果与数据集的质量有着密切的关系,充足且均衡的数据集不但能够提升模型的分类准确率而且还能在一定程度上避免过拟合现象的产生. 当数据集样本数量较小或者各类别样本数量不均衡时,使用数据增强技术可以增加少数类的样本数量,从而抑制数据集样本类别不均衡给模型造成的影响,提高模型的鲁棒性. 常见的图像数据增强是通过对原始图像数据的变换来生成新的数据,比如:缩

放、翻转、移位等. 为解决恶意代码数据集中各类样本数量不均衡的问题,本文使用python中的图像数据增强技术函数对数据集进行样本扩充,表2给出了实验中使用的数据增强技术的参数设置. 本文将Malimg数据集的70%划分为训练集,20%划分为验证集,10%划分为测试集. 本文模型使用数据增强技术将原训练集的6 604个样本增扩到51 608个样本.

表2 数据增强技术的参数设置

方法	设置	方法	设置
rescale	1/255	shear range	0.0
width shift	0.0	zoom range	0.0
height shift	0.0	horizontal flip	False
rotation range	0.0	fill mode	None

2.2 FFSE模型构建

卷积神经网络通过端到端的学习能够自动地提取样本的特征,并根据特征对样本进行分类. 很多学者通过构建卷积神经网络模型来识别和分类恶意代码,但是这些方法均使用单一尺度的卷积核对恶意图像进行特征提取,忽略了特征提取的多样性,导致提取的特征不具备鲁棒性并且影响了恶意代码的检测精度.

为了解决上述的问题,本文结合通道注意力机制(Squeeze and Excitation Networks,SE)[20]设计了一个多尺度特征融合的网络结构,如图1中FFSE模型所示. 模型的核心设计思想为增强模型的特征提取能力,使用少量的神经网络层数获得较深的神经网络的特征提取效果. 通过减少神经网络参数、降低浮点运算量来提升模型运算速度,在提高恶意代码分类准确率的同时具有较快的恶意代码检测速度. 模型主体由CBR层、最大池化层、FFSE模块、以及全连接层构成. 其中,CBR层是本文模型的基础单元,其包括卷积层、BN(Batch Normalization)层和Relu(Rectified linear unit)激活函数. 它是传统卷积层的一种改进,能够加速模型的收敛. 其流程为:首先,输入特征进入卷积层进行卷积操作,然后进入BN层进行批量归一化,最后经过Relu函数进行激活得到非线性特征输出.

FFSE模块是模型的核心结构,其包括特征融合模块和通道注意力机制模块,其结构如图5所示. 特征融合部分的核心思想是同时使用不同大小的卷积核提取图像的多尺度的特征,并将这些特征相融合以获得兼顾局部特征与全局特征的总特征. 在特征提取的过程中,特征图是由每一个通道提取的特征结合得到,但并非每一个通道都能有效的提取特征. 通道注意力机制能够根据各个通道的特征提取效果计算各个通道的权重,赋予特征提取效果好的通道相对较大的权重,赋予特征提取效果差的通道相对较小的权重,将通道注意力集中在图像的主要特征上,以增强恶意代码图像的关键特征表达,提升恶意代码检测和分类的精度.

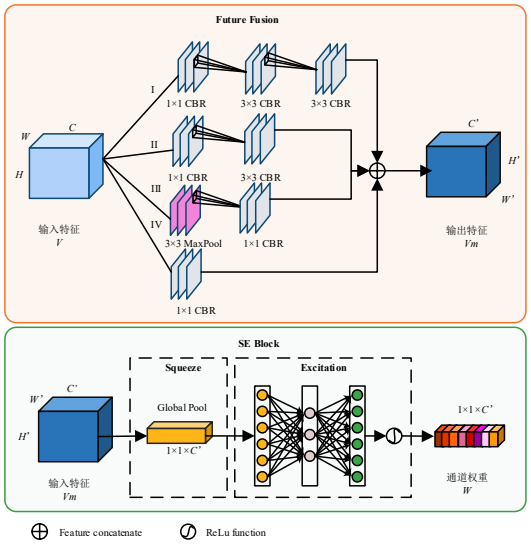


图5 FFSE模块结构示意图

在特征融合模块中,首先,输入特征$V\in\mathbb{R}^{C\times H\times W}$会同时通过四个分支I,II,III,IV进行运算,为了使提取的特征具有多样性、代表性,在每个分支中采用的是不同感受野的卷积核进行特征提取,每个分支会得到相应的分支输出特征$V_1\in\mathbb{R}^{C_1\times H'\times W'}$,$V_2\in\mathbb{R}^{C_2\times H'\times W'}$,$V_3\in\mathbb{R}^{C_3\times H'\times W'}$,$V_4\in\mathbb{R}^{C_4\times H'\times W'}$. 然后,将得到的分支输出特征$V_1,V_2,V_3,V_4$进行融合,得到既包含局部特征又包含全局特征的总特征,并将其作为输出特征$V_m\in\mathbb{R}^{C'\times H'\times W'}$,输出通道数$C'=C_1+C_2+C_3+C_4$. 通道注意力模块分为两个部分:压缩(Squeeze)和激活(Excitation). 首先,压缩操作F_s是对V_m进行全局池化得到$Z\in\mathbb{R}^{1\times 1\times C}$,其公式如下:

$$z_c=F_s(V_m)=\frac{1}{H'+W'}\sum_{i=1}^{H'}\sum_{j=1}^{W'}V_m(i,j) \tag{2}$$

然后,对压缩得到的Z进行激活操作F_e得到通道的权重值w:

$$w=F_e(Z,W')=\sigma\left(g(Z,W')\right)=\sigma\left(W_2'\delta(W_1'Z)\right) \tag{3}$$

其中,δ是Relu激活函数,$W_1'\in\mathbb{R}^{\frac{C}{r}\times C}$,$W_2'\in\mathbb{R}^{C'\times\frac{C'}{r}}$,$w\in\mathbb{R}^{1\times 1\times C}$. r为变换中的超参数一般取$r=16$. 最后,将计算得到的通道的权重值w与输入特征V_m进行F_{scale}操作,实现将通道的权重值赋予给输入特征的各个通道,得到通道加权后的输出$\tilde{V}_m$,其公式如下:

$$\tilde{V}_m=F_{\text{scale}}(v_c,w_c) \tag{4}$$

其中,$\tilde{V}_m=[\tilde{v}_1,\tilde{v}_2,\cdots,\tilde{v}_C]$,$\tilde{v}_C\in\mathbb{R}^{H'\times W'}$. 恶意图像通过FFSE模块后,得到一个局部特征与全局特征相融合的综合特征,通道注意力机制对特征图中每个通道特征赋予权重,抑制特征提取效果差的通道特征表达,增强

特征提取效果好的通道特征表达,进而提高图像纹理特征提取能力.

3　数据集及评价指标

3.1　实验环境及数据集

本文实验采用64位Windows10操作系统,Intel(R) Core(TM) i7-7700HQ CPU,16 GB RAM,Nvidia GeForce GTX 1050 GPU,Python 3.6编译环境和Tensorflow 2.1深度学习框架.

本文在两个恶意代码数据集上来评估FFSE模型的性能,其具体信息如下:

数据集一由奇安信息技术研究院2020年"DataCon开放数据计划"[21]提供,记作DataCon. 数据集中共计23 655个PE样本,包含15 759个正常样本和7 896个恶意挖矿样本. DataCon中的样本均源于从现网中捕捉的真实数据,包含大量的经过加壳、混淆后的样本. 在实验中,数据集的70%划分为训练集,20%划分为验证集,10%划分为测试集.

数据集二为Malimg数据集[5],它包含25个恶意家族共9 435个恶意代码. Malimg数据集中各个恶意家族的样本数量相差较大,是一个不均衡数据集. 在实验中,将Malimg数据集的70%划分为训练集,20%划分为验证集,10%划分为测试集.

3.2　实验环境及数据集

本文采用准确率(Accuracy)、精确率(Precision)、召回率(Recall)和F1分数(F1-score)四个指标对模型的性能进行评价,这些评价指标已经广泛的应用于相关研究[22~24],其公式如下:

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{5}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{6}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{7}$$

$$\text{F1-score}=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{8}$$

其中,TP表示被正确预测为正类的正样本,FP表示被错误预测为正类的负样本,FN表示被错误预测为负类的正样本,TN表示被正确预测为负类的负样本.

4　实验结果与分析

为了验证本文模型检测恶意代码的效果和效率,本节设计了如下实验:(1)输入图像尺寸选择实验;(2)数据增强技术有效性验证实验;(3)模型恶意代码检测能力验证实验(4)与经典神经网络模型的对比实验;(5)模型消融实验;(6)与近期恶意代码分类模型的对比实验.

4.1　输入图像尺寸选择实验

由于CNN中全连接层的限制,输入到模型中的恶意代码图像大小必须是固定的. 另外,输入CNN的图像尺寸不仅会影响模型的大小,也会影响模型性能. 为了得到更适合模型的输入图像尺寸的大小,使用双线性插值法将恶意代码图像归一化至32×32、64×64、128×128、256×256和512×512. 然后,将Malimg数据集中的恶意图像输入到模型中测试模型的性能,实验结果如表3所示. 从表3中可分析得出恶意代码图像尺寸从32×32增加到256×256的过程中,准确率从84.28%提升至99.05%;图像尺寸从256×256增加到512×512的过程中,准确率从99.04%下降至98.61%,这说明模型出现了过拟合现象. 另外,参数量随着图像尺寸的增大不断增大,这是因为图像尺寸越大卷积运算越多参数量越大,并且参数量越大消耗的计算机资源越大,也会导致模型的训练时间增加. 综合恶意代码分类精度与参数量,选择256×256的恶意代码图像作为模型的输入.

表3　输入图像尺寸对模型影响的实验结果

尺寸	准确率/%	精确率/%	召回率/%	F1/%	参数/M
32×32	84.28	83.44	84.27	82.82	0.31
64×64	94.22	93.08	94.22	93.26	0.35
128×128	98.07	98.06	98.07	98.07	0.50
256×256	99.04	99.18	99.04	99.06	1.11
512×512	98.61	98.61	98.60	98.60	3.58

4.2　数据增强技术有效性验证

为了验证数据增强技术对模型性能提升的有效性,我们在Malimg数据集上把使用数据增强技术的模型与未使用数据增强技术的模型进行实验对比,结果如表4所示. 通过观察可知,使用数据增强技术的模型的准确率为99.04%,未使用数据增强的数据增强技术的模型准确率为98.35%. 使用数据增强技术后的模型不论是在准确率、精确率、召回率或是F1分数方面表现均优于未使用数据增强技术的模型. 这证明数据增强技术能够消除数据集不平衡带来的影响,有效地提升恶意代码检测模型的性能.

表4　使用数据增强技术前后模型性能比较

模型	准确率/%	精确率/%	召回率/%	F1/%
数据增强前	98.35	98.42	98.35	98.38
数据增强后	99.04	99.18	99.04	99.05

4.3　模型的恶意代码检测能力验证实验

为了测试本文模型检测恶意代码的能力,本节在DataCon上与近期恶意代码检测方法进行实验对比. 杨望等[25]在DataCon上提取其函数调用图,并将归一化后

的函数调用图输入到深度神经网络中进行训练得到检测结果. 刘亚姝等[26]先使用LDA(Latent Dirichlet Allocation)算法对预处理后的恶意代码进行降维,并将降维后的特征作为输入训练随机森林分类器得到检测结果. Guo等[27]将数据集映射为恶意图像并使用GIST算法提取恶意图像特征,并训练KNN分类器和随机森林分类器进行投票实现恶意代码的检测. Saadat等[28]使用卷积神经网络作为特征提取器,训练XGBoost分类器来检测恶意代码. 图6为本文模型与近期恶意代码检测模型的对比实验结果,从图中可清晰观察到本文方法在DataCon上的检测准确率为96.35%,高于上述文献中的检测准确率. 这说明本文模型具有良好的恶意代码检测能力,能够有效地检测目前现网中的恶意代码变种.

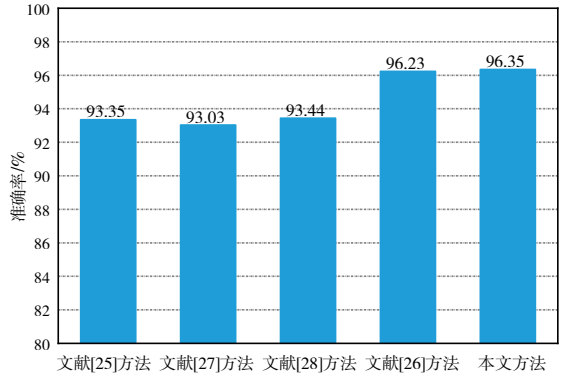


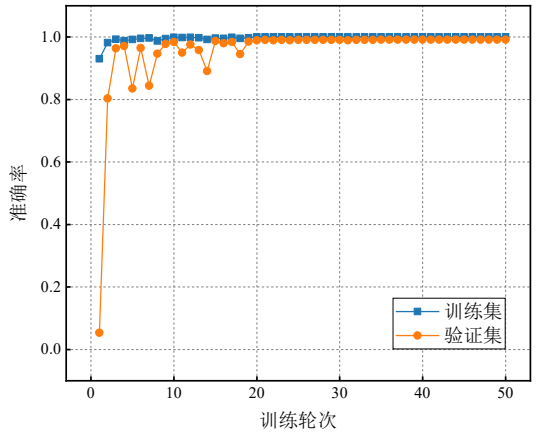
图6 各恶意代码检测方法的准确率对比

4.4 模型的恶意代码检测能力验证实验

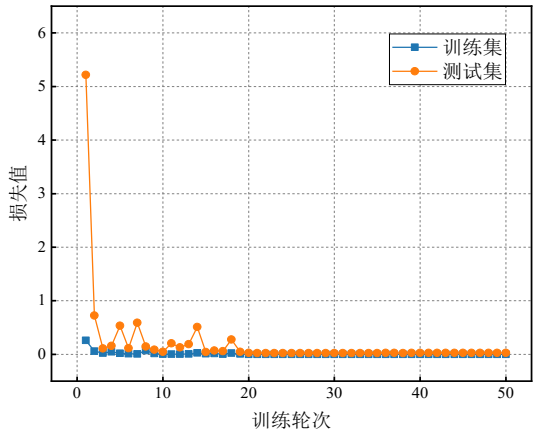
为了验证本文模型的效果,本节在Malimg数据集上首先对模型的训练性能进行了评估,然后将FFSE模型与AlexNet[29],VGGNet[30],ResNet[31]这些卓越的深度神经网络模型进行实验对比.

图7为FFSE模型的训练曲线,以50轮为基准,FFSE模型的训练时长为10 081.05 s,内存消耗为3 365.03 M. 从图7中可观察到,当训练轮次达到20时模型已经收敛,这说明收敛速度较快. 模型收敛后训练集的准确率达到100%,测试集准确率达到99.04%,模型在训练集和测试集上均表现良好,没有出现过拟合现象. 另外,模型的训练时间和GPU的性能息息相关,使用性能强大的GPU能够大大缩短模型训练时间,以满足在几十万到上百万个恶意代码上于可接受的时间内训练出模型.

表5为FFSE模型与经典神经网络模型对比实验的结果,从中可看出,本文模型不论是准确率、精确率、召回率或是F1-score都高于其他三个模型,且有较大幅度的提升. 以精确率为例,本文模型的精确率为99.18%,相较于AlexNet8的92.77%提升了6.41%,比VGG16的97.00%提升了2.18%,比ResNet的97.57%提升了1.61%. 实验证明了本文模型在恶意代码分类精度方面优于其他三个模型.



(a) 准确率曲线



(b) 损失曲线

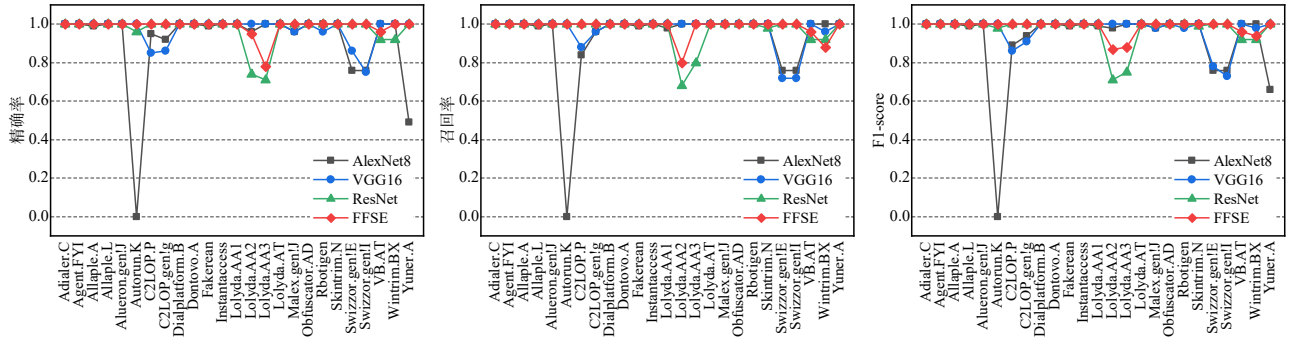
图7 FFSE模型训练曲线

表5 不同模型的实验结果

模型	准确率/%	精确率/%	召回率/%	F1-score/%
AlexNet8	94.12	92.77	94.11	93.19
VGG16	97.11	97.00	97.11	97.03
ResNet	97.54	97.57	97.54	97.53
FFSE	99.04	99.18	99.04	99.05

为了详细地观察各个模型分类的具体情况,绘制了四个模型在数据集中各个类别的分类情况分布图,结果如图8所示. 从图8中可以看出,AlexNet在Autorun.X家族分类精度仅为0,且在Yuner.A家族分类精度不足50%,这是AlexNet模型分类精度远低于其他三个模型的主要原因. 而VGG16与ResNet在Malimg数据集上分类精度相差不大,在Lolyda.AA2和Lolyda.AA3恶意家族中VGG16分类效果好于ResNet,但是在C2LOP.P,C2LOP.gen! g,

Swizzor.gen！E,Swizzor.gen！恶意家族中VGG16分类效果不如ResNet. 而FFSE模型不同程度上改善了VGG16和ResNet在一些易混淆恶意家族中分类精度不足的问题,进而提升了总体分类精度,比如在C2LOP.P,C2LOP.gen！g,Swizzor.gen！E,Swizzor.gen！家族中FFSE模型分类效果优于VGG16,在Lolyda.AA2和Lolyda.AA3恶意家族中FFSE模型的分类效果优于ResNet. 特征融合与通道注意力机制的结合具有良好的特征提取能力,能够有效地提取恶意代码关键特征,区分易混淆的恶意家族,提升模型准确率.



(a) 各模型的精确率　(b) 各模型的召回率　(c) 各模型的F1-score

图8　各模型在Malimg数据集中的精确率、召回率和F1-score

除分类精度外,模型对未知恶意代码的检测时间对于恶意代码的检测与分类也是一项重要的指标,检测时间是指检测恶意代码所产生的时间开销. 实验以测试集的恶意代码图像为单位,计算模型的参数量大小和平均检测时间,实验结果如表6所示. 实验结果表明FFSE的参数量为1.11 M,是AlexNet8模型的1/60、VGG16的1/43、ResNet的1/22,实现了轻量化的目的. 另外,FFSE模型的检测时间最短,检测一个未知样本平均需要4.2 ms,仅为IMCFN[16]的检测时间180 ms的1/20. 综上,本文的模型不仅拥有较高的准确率,并且参数量低、有着极快的检测速度. 这得益于FFSE模型特征融合模块卓越的特征提取能力,它能够有效提取恶意代码深度关键特征. 更加高效的特征提取能力意味着能够使用更少的神经网络层数获得深层神经网络的特征表示,减少神经网络层数能够减少参数量、降低浮点数运算量,进而提升模型的运算速度.

表6　不同模型准确率与预测时间实验结果

指标	FFSE	AlexNet8	VGG16	ResNet
参数量/M	1.11	60.00	43.40	22.69
检测时间/ms	4.2	10.7	11.8	20.3

4.5　模型消融实验

为进一步验证FFSE模型的在恶意代码分类中的有效性,本节将对模型进行消融实验. 我们将FFSE模型与CNN、CNN+SE和FF模型的分类性能进行实验对比,结果如表7所示.

从表7中可得,CNN+SE的准确率在数值上比CNN模型的高1.39,FFSE模型比FF模型的准确率高0.75,说明通道注意力机制能够有效的提升模型分类准确率. FF模型比CNN模型的分类准确率高1.82,FFSE模型比

表7　不同模型的实验结果

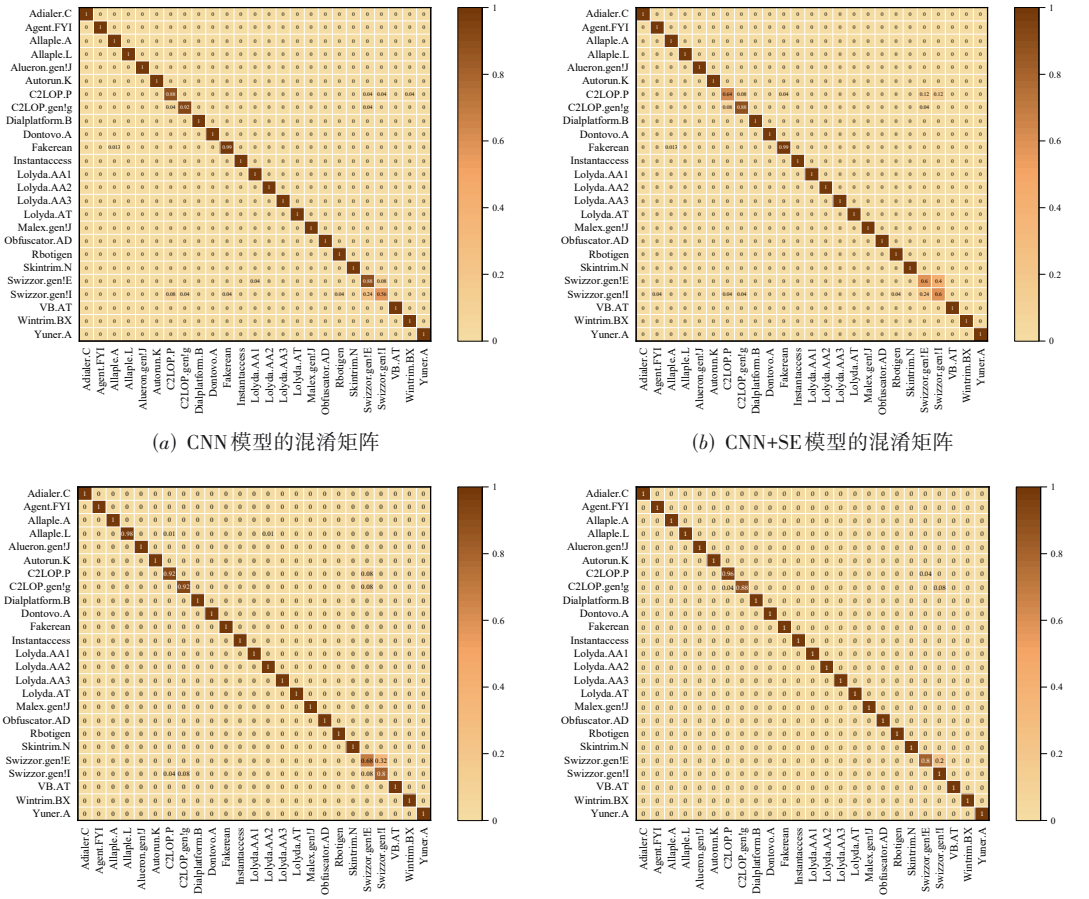
模型	准确率/%	精确率/%	召回率/%	F1-score/%
CNN	96.47	96.53	96.47	96.46
CNN+SE	97.86	97.85	97.86	97.77
FF	98.29	98.26	98.29	98.27
FFSE	99.04	99.18	99.04	99.05

CNN+SE模型的准确率高1.18,说明特征融合模块能够有效提升模型准确率. FFSE模型的分类准确率相较其它三种模型准确率有明显提升,说明特征融合与通道注意力机制的结合能够有效提升模型的分类性能. 为了更加清晰仔细的观察各个模型在每个恶意家族中的具体分类情况,绘制了各个模型的混淆矩阵,其结果如图9所示.

对比图9中的结果可得到,FFSE模型相较其它模型在C2LOP.P,C2LOP.gen！g等易混淆家族中的分类效果均有所提升,说明特征融合与通道注意力机制的结合能够增强特征表达、改善易混淆恶意家族的分类效果进而提高模型的整体分类性能,这也进一步证明了本文提出模型在恶意代码分类任务中具有着卓越的表现.

4.6　与近期恶意代码分类方法的对比实验

为了验证本文方法的分类性能,我们将其与同样使用Malimg数据集的基于可视化的最新恶意代码检测方法进行对比. 表8为各个模型的性能对比结果,显然,本模型优于现有的基于可视化的恶意代码分类方法.



(a) CNN模型的混淆矩阵 (b) CNN+SE模型的混淆矩阵

(c) FF模型的混淆矩阵 (d) FFSE模型的混淆矩阵

图9 各模型的混淆矩阵

表8 不同恶意代码分类方法的实验结果

方法	时间	方法	准确率/%	精确率/%	召回率/%	F1-score/%
GIST+KNN[5]	2011	机器学习	97.18	—	—	—
SPAM-GIST[10]	2016	机器学习	97.40	—	—	—
DRBA+CNN[18]	2018	深度学习	94.50	96.60	88.40	—
刘亚姝等人[12]	2018	机器学习	98.80	—	—	—
LGMP+KNN[13]	2019	机器学习	98.40	—	98.20	—
NSGA-II+CNN[19]	2019	深度学习	97.60	97.60	88.40	—
Venkatraman[32]	2019	深度学习	96.30	91.80	91.50	91.60
Gibert[15]	2019	深度学习	98.50	98.00	98.00	98.00
IMCFN[16]	2020	深度学习	98.82	98.85	98.81	98.75
Vinita[33]]	2020	集成学习	98.58	98.04	98.06	98.05
DCNN[34	2020	深度学习	98.79	98.79	98.47	98.46
DEAM-Densenet[35]	2021	深度学习	98.50	96.90	96.60	96.70
MFFC[36]	2021	深度学习	98.72	98.86	98.72	98.73
FFSE	**—**	**深度学习**	**99.04**	**99.18**	**99.04**	**99.05**

5 结论

本文对基于可视化的恶意代码检测方法进行了研究,针对恶意图像纹理特征提取成本高且特征鲁棒性差、检测时效性不足问题,提出了一种基于特征融合的恶意代码快速检测方法. 该方法采取模块化设计,提出了一种特征提取能力强且参数量小的轻量化卷积神经

网络结构. 实验结果表明,本文模型具有良好的恶意代码检测和分类能力,在提高检测和分类准确率的同时兼顾了时效性.

在实验中,我们发现大部分可视化后的恶意图像长宽比例不同,导致恶意图像归一化过程中出现图像被拉伸或压缩的现象. 虽然使用双线性插值算法后的图像的基本纹理特征能够保留,但是无法避免其部分纹理特征出现变化. 纹理特征的变化会影响模型的特征提取,而特征提取是影响模型分类准确率的关键. 在未来工作中,如何消除在归一化过程中恶意图像纹理特征变化给模型带来的影响并提升模型分类准确率是需要解决的问题.

参考文献

[1] 国家互联网应急中心. 2021年第2期网络安全信息与动态周报[R/OL]. (2021-01-13)[2021-12-23]. https://www.cert. org. cn/publish/main/upload/File/Weekly% 20Report% 20of%20CNCERT-Issue%2002%202021(1).pdf.

[2] ALAZAB M. Profiling and classifying the behavior of malicious codes[J]. Journal of Systems and Software, 2015, 100: 91-102.

[3] VENKATRAMAN S, ALAZAB M. Use of data visualization for zero-day malware detection[J]. Security and Communication Networks, 2018, 2018: 1-13.

[4] CONTI G, BRATUS S, et al. Automated mapping of large binary objects using primitive fragment type classification [J]. Digital Investigation, 2010, 7: S3-S12.

[5] NATARAJ L, KARTHIKETAN S, 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.

[6] 韩晓光, 曲武, 等. 基于纹理指纹的恶意代码变种检测方法研究[J]. 通信学报, 2014, 35(8): 125-136.
HAN X G, QU W, et al. Research on malicious code variants detection based on texture fingerprint[J]. Journal on Communications, 2014, 35(8): 125-136. (in Chinese)

[7] 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, 2011: 21-30.

[8] 汪嘉来, 张超, 戚旭衍, 等. Windows平台恶意软件智能检测综述[J]. 计算机研究与发展, 2021, 58(5): 977-994.
WANG J L, et al. A survey of intelligent malware detection on windows platform[J]. Journal of Computer Research and Development, 2021, 58(5): 977-994. (in Chinese)

[9] 任卓君, 陈光, 卢文科. 基于N-gram特征的恶意代码可视化方法[J]. 电子学报, 2019, 47(10): 2108-2115.
RENG Z J, CHEN G, LU W K. Malware visualization methods based on n-gram features[J]. Acta Electronica Sinica, 2019, 47(10): 2108-2115. (in Chinese)

[10] NATARAJ L, MANJUNATH B S. SPAM: Signal processing to analyze malware[J]. IEEE Signal Processing Magazine, 2016, 33: 105-117.

[11] KANCHERLA K, MUKKAMALA S. Image visualization based malware detection[C]//2013 IEEE Symposium on Computational Intelligence in Cyber Security. Singapore: IEEE, 2013: 40-44.

[12] 刘亚姝, 王志海, 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, et al. Method of anti-confusion texture feature descriptor for malware images[J]. Journal on Communications, 2018, 39(11): 44-53. (in Chinese)

[13] NAEEM H, GUO B, NAEEM M R, et al. Identification of malicious code variants based on image visualization[J]. Computers & Electrical Engineering, 2019, 76: 225-237.

[14] 卢喜东, 段哲民, 钱叶魁, 等. 一种基于深度森林的恶意代码分类方法[J]. 软件学报, 2020, 31(5): 1454-1464.
LU X D, DUAN Z M, QIAN Y K, et al. Malicious code classification method based on deep forest[J]. Journal of Software, 2020, 31(5): 1454-1464. (in Chinese)

[15] 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.

[16] DANISH V, MAMOUN A, SOBIA W, et al. IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture[J]. Computer Networks, 2020, 171: 107138.

[17] KABANGA E K, KIM C H. Malware images classification using convolutional neural network[J]. Journal of Computer and Communications, 2018, 6(1): 153-158.

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

[19] CUI Z H, LEI D, et al. Malicious code detection based on CNNs and multi-objective algorithm[J]. Journal of Parallel and Distributed Computing, 2019, 129: 50-58.

[20] HU J, SHEN L, ALBANIE S, et al. Squeeze and excitation networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(8): 2011-2023.

[21] 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)[2021-12-23]. https://datacon.qianxin.com/opendata.

[22] LI Q, MI J, LI W, et al. CNN-based malware variants detection method for internet of things[J]. IEEE Internet of Things Journal, 2021, 8(23): 16946-16962.

[23] SUDHAKAR K S. MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in internet of things[J]. Future Generation Computer Systems, 2021, 125: 334-351.

[24] DANISH V, MAMOUN A, SOBIA W, et al. Image-based malware classification using ensemble of CNN architectures (IMCEC)[J]. Computers & Security, 2020, 92: 101748.

[25] 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)

[26] 刘亚姝, 王志海, 侯跃然, 等. 一种基于概率主题模型的恶意代码特征提取方法[J]. 计算机研究与发展, 2019, 56(11): 2339-2348.
LIU Y S, WANG Z H, HOU Y R, et al. A method of extracting malware features based on probabilistic topic method, 2019, 56(11): 2339-2348. (in Chinese)

[27] GUO H, HUANG S, ZHANG M, et al. Classification of malware variant based on ensemble learning[C]//2020 International Conference on Machine Learning for Cyber Security. Guangzhou: Springer, 2020: 125-139.

[28] SAADAT S, JOSEPH R V. Malware classification using CNN-XGBoost model[C]//2020 International Conference on Advanced Computing Technology. Chennai: Springer, 2021: 192-202.

[29] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6): 84-90.

[30] SIMONVAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2015-4-10)[2021-12-23]. https://arxiv.org/abs/1409.1556.

[31] HE K M, ZHAGN X Y, REN S Q, et al. Deep residual learning for image recognition[EB/OL]. (2015-04-10)[2021-12-23]. https://arxiv.org/abs/1512.03385.

[32] 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.

[33] VINITA V, SUNIL K M, SINGH V B. Multiclass malware classification via first- and second-order texture statistics[J]. Computers & Security. 2020, 97: 101895.

[34] NAEEM H, ULLAH F, NAEEM M R, et al. Malware detection in industrial internet of things based on hybrid image visualization and deep learning model[J]. Ad Hoc Networks, 2020, 105: 102154.

[35] WANG C, ZHAO Z, WANG F, et al. A novel malware detection and family classification scheme for IoT based on DEAM and DenseNet[J]. Security and Communication Networks, 2021, 2021: 1-16.

[36] WANG S, WANG J, SONG Y F, et al. Malicious code variant identification based on multiscale feature fusion CNNs[J]. Computational Intelligence and Neuroscience, 2021, 2021: 1070586.

作者简介

王 硕 男,1998 年 11 月出生于重庆市. 现为空军工程大学硕士研究生. 主要研究方向为智能信息处理和恶意软件检测.
E-mail: luoan_w@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软件检测.
E-mail: 26471375@qq.com

王亚男 女,1988 年 9 月出生于山东省青岛市. 现为空军工程大学防空反导学院讲师. 主要研究方向为网络信息安全和人工智能.
E-mail: wyn1988814@163.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