文章编号：2096-1618(2020)03-0253-06

基于优化 VGG19 卷积神经网络的异常检测模型研究

王文文， 陶宏才
(西南交通大学信息科学与技术学院，四川 成都 611756)

摘要：互联网服务已经成为人们生活中必不可少的一部分，但由于网络攻击方式的不断增多，使得网络安全问题日益严重。异常检测是对 Web 攻击进行检测的方式，基于优化 VGG19 神经网络建立了一种新的异常检测模型，并在 ISCX2012 数据集上进行训练，取得了较好的检测效果。

关 键 词：网络攻击；异常检测；卷积神经网络；VGG19
中图分类号：TP391.1 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2020.03.001

0 引言

随着 Web 服务普及率的逐渐提高，用户接触到的网络应用也在逐渐增多。但由于一些网络服务器管理人员和开发者缺乏安全意识，使得各种安全漏洞存在于服务器应用中，也使得这些应用服务器成为了网络攻击的重灾区。SQL 注入、路径遍历与 DDoS 拒绝攻击等，是近年来常见的攻击手段[1]。入侵检测是抵御 Web 攻击的重要方法。误用检测与异常检测是入侵检测中常用的两种方法。

误用检测基于规则库定义 Web 攻击的特点，对于已知的攻击行为，有较高的检测率。不过，随着技术的不断发展，网络攻击的方法也在不断增多。因此，误用检测如果想获得较高的检测率，就需要不断更新自身的规则库，需要负责维护规则库的人员不断学习新知识，才能完成此项工作，导致人员学习成本较高。同时，对于未知的攻击方式，误用检测的检测成功率比较低[2]。

异常检测弥补了误用检测的缺点，它使用数据挖掘与统计分析的方法对正常样本集中的特征进行提取[3]，如果某个网络请求不能体现出正常请求的特征，则将它归为异常请求。只要构建适当的模型，异常检测对 Web 攻击的检测有较高的准确率。国内外众多学者在对异常检测模型的研究中，取得了许多研究成果。T. Komviriyavut 等[4]、S. Sahu 等[5]使用决策树来构建入侵检测模型，实验结果证明决策树适用于入侵检测位置攻击中。Livadas C 等[6]使用贝叶斯分类器对 IRC 僵尸网络进行监测，取得了较好的效果。Reddy R R 等[7]基于支持向量机来挑选判别函数，通过缩放数据集提高判别函数的性能，并将此判别函数用于入侵检测，取得了较高的检测率。赵夫群[8]阐述了 LSSVM 算法应用到网络入侵检测中的原理。

文中构建了优化 VGG19 神经网络的异常检测模型，通过将 ISCX2012 数据集中的数据转为流量灰度图，并输入至模型进行训练，取得了较好的检测效果。

1 一种新的异常检测模型

1.1 问题提出

常用的异常检测模型算法包括 SVM、LSTM 神经网络和决策树算法等，但是这些算法都需要手工提取数据的特征，模型的检测效果依赖于人工挑选出特征值的情况。网络攻击方式在不断增多，这些算法已很难适应不断增加的检测需求，并且检测精度较低。卷积神经网络在近些年来取得了飞速的发展，它基于非线性的网络结构提取属性间的内在关系，实现高维空间的特征表，避免了人工提取数据特征易出现的考虑不足和冗余性等问题[9]。VGG19[10]作为一种分类功能十分优秀的卷积神经网络，由牛津大学的研究人员在 ILSVRC2014 竞赛中提出，并在当年的分类比赛中取得了第二名。因此，提出一种基于优化 VGG19 深度卷积神经网络的异常检测模型，通过将网络请求数据转为流量灰度图，并输入到模型来完成请求检测。该模型在传统 VGG19 网络的基础上，改进全连接层的结构与数量，通过 2 标签的 Softmax 分类器替代原网络中的 Softmax 分类器，基于微型迁移学习来共享预训练模型中池化层和卷积层的参数与权值。新模型更适用于异常检测模型的分类特点，提高了训练效率，使异常检测的请求分类结果更加准确。

1.2 VGG19 模型

VGG19 卷积神经网络是一种具有深度结构且含有卷积运算的前馈神经网络，其主要特点是在卷积运

收稿日期：2020-02-27
基金项目：国家自然科学基金资助项目(61806170)

算中使用 3×3 的卷积核进行运算,从而提取出输入图片中更多的显著特征[11]。它由 Softmax 层、池化层、卷积层与全连接层堆叠而成,如图 1 所示。在全连接层与卷积层后,需利用激活函数进行计算,同时池化层穿插其中。全连接层数量为 3,最后的 Softmax 层用于分类。

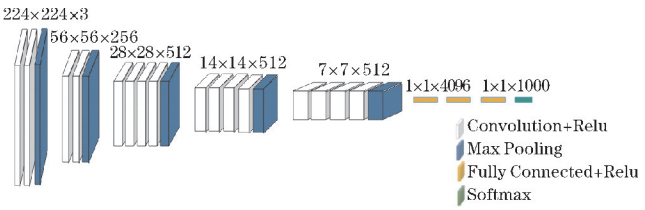

图 1 VGG19 神经网络结构

卷积层的计算目的是将输入图片的主要特征提取,将运算结果处理为新的特征图像[12],公式为

$$y_{\text{conv}}=\delta(Mat\cdot W+b) \tag{1}$$

式中,y_{conv}是输出结果,δ 是激活函数,Mat 为灰度图矩阵,W 是卷积核,$\cdot$ 表示灰度图矩阵与卷积核进行卷积运算,b 是偏置值。卷积运算的结果需进入激活函数,如公式(2)所示。

$$f_{\text{retu}}(x)=\begin{cases}x & x\geqslant 0\\ 0 & x<0\end{cases} \tag{2}$$

其中,x 为经过卷积运算后的输出值。$x\geqslant 0$ 时,输出为 x;否则输出为 0。使用此激活函数,可减少计算量,并且使网络变得稀疏,减少参数间依赖关系与过拟合现象的发生。

在池化层,需要将图片的特征做进一步提取,以减少参数。池化方法为最大池化,步长为 2,池化框尺寸为 2×2,其表达式为

$$f_{\text{pool}}=\text{Max}(x_{m,n},x_{m+1,n},x_{m,n+1},x_{m+1,n+1}) \tag{3}$$

其中,$0\leqslant m\leqslant M$,$0\leqslant n\leqslant N$,M 和 N 分别为图像二维向量的长与宽,最大池化为在所选图像区域中,挑出最大的值,f_{pool}为最大池化的结果。

Softmax 层是为了解决数据分类问题的分类器,其回归多分类数据标签 $y\geqslant 2$,训练集为 k 个标记过的样本:

$$T=\{(x_{(1)},y_{(1)}),(x_{(2)},y_{(2)}),\cdots,(x_{(k)},y_{(k)})\} \tag{4}$$

分类标签是 $y_{(i)}\in\{1,2,\cdots,k\}$,样本集合是 $x_{(i)}$,j 是不同类别,并估算 j 的概率值,对于每个样本,第 k 类标签的概率是:

$$P(y=j\mid x)\quad j=1,2,\cdots,k \tag{5}$$

将此回归样本变为概率向量,其维度为 k,推导公式为

$$h(x_{(i)}\mid\theta)=\begin{bmatrix}p(y_{(i)}=1\mid x_{(i)},\theta)\\ p(y_{(i)}=2\mid x_{(i)},\theta)\\ \cdots\cdots\\ p(y_{(i)}=k\mid x_{(i)},\theta)\end{bmatrix} \tag{6}$$

模型的学习参数公式(7):

$$\theta=[\theta_1^T,\theta_2^T,\cdots,\theta_k^T]\quad\theta_1^T,\theta_2^T,\cdots,\theta_k^T\in R^{n+1} \tag{7}$$

$$y=\frac{1}{\sum_{j=1}^{k}e^{x_{(i)}\theta_j^T}}=\frac{1}{\sum_{j=1}^{k}e^{\theta_j^Tx_{(i)}}}\begin{bmatrix}e^{\theta_1^Tx_{(i)}}\\ e^{\theta_2^Tx_{(i)}}\\ \cdots\\ e^{\theta_k^Tx_{(i)}}\end{bmatrix} \tag{8}$$

公式(8)是将概率总和相加为 1。对样本集不断进行学习与训练,Softmax 基于迭代优化来拟合数据曲线。损失函数的公式是:

$$J(\theta)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y_{(i)}=j\}\log\frac{e^{x_i\theta_j^T}}{\sum_{i=1}^{k}e^{x_{(i)}\theta_i^T}}\right] \tag{9}$$

为了使误差达到最小,需要最小化损失函数的值[13],通过调整参数 θ 来完成。其中,$1\{y(i)=j\}$ 代表如果 $y(i)=j$,值为 1,其他情况则为 0。通过损失函数的计算与迭代,模型参数进一步优化,直至达到最大迭代次数或小于指定误差。

1.3 VGG19 模型的不足与优化

传统 VGG19 卷积神经网络包含 3 个全连接层,其缺点是可能破坏图片的空间结构。并且在 VGG19 中,每一层的网络结点的值与权重都需要调整与训练,而每一个全连接层拥有最多 4096 个结点,造成神经网络的训练时间过长。同时,Softmax 层分类的数目为 1000,不符合异常检测模型的特点。

针对传统 VGG19 网络应用在异常检测模型中的不足,在其基础上进行优化,提出基于优化后的 VGG19 异常检测模型。因异常检测模型最后输出的分类结果为正常和异常,因此使用 2 标签的 Softmax 分类层来替代原网络中的 Softmax 分类层,同时模型的激活函数依然为 Relu 函数。

传统的 VGG19 网络是基于 100 万张图片的 ImageNet[14]数据库训练而成,具有很强的特征学习能力,特别是卷积层对于图片轮廓、曲线、边缘的特征提取,在各层中都拥有已经训练完成的权重与参数。为提高训练效率,将基于 ImageNet 中已训练好的 VGG19 网络作为模型的预训练模型,运用微调迁移学习的方法,将预训练模型中的卷积层参数迁移到模型中的卷积层。

同时,在传统的 VGG19 模型中,拥有 3 个全连接层,每层拥有参数为 4096 个,参数的数量是为 1000 标签的 Softmax 分类层所设计。模型是一个二分类问题,全连接层数量过多并不能提高模型的准确率,因此将 3 个全连接层删除,加入一个具有 128 个结点的全连接层,更适用于异常检测的特点,如图 2 所示为经过优化后的 VGG19 异常检测模型。

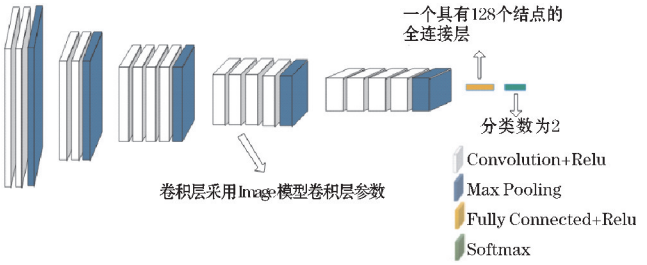

图 2　优化后的的 VGG19 异常检测模型

1.4　新模型算法代码描述

优化后的 VGG19 深度神经网络包含 16 个卷积层、5 个池化层、1 个全连接层与 1 个 Softmax 层。依据 VGG19 网络的特点,卷积核的大小(变量 FH×FW)为 3×3,步长(变量 stride)为 1,输入图像的通道数目(变量 C)与卷积核的深度(变量 FC)均为 3,填充值(变量 pad)为 0。假设输入图像大小为 H×W,输出图片大小为 OutputH×OutputW,则卷积层运算的 Python 代码如下所示:

```
class Convolution:
def_init_(self, W, b, stride = 1, pad = 0):
    self. W = W
    self. b = b
    self. stride = stride
    self. pad = pad

  def forward(self, x):
    (FN, C, FH, FW) = self. W. shape
    (N, C, H, W) = x. shape
    out_h=int(1+(H+2 * self. pad-FH)/self. stride)
    out_w=int(1+(W+2 * self. pad-FW)/self. stride)
    col=im2col(x, FH, FW, self. stride, self. pad)
    col_W=self. W. reshape(FN, -1)
    #reshape 函数会自动计算-1 维度上的元素个数,以使多维数组的元素个数前后一致
  out=np. dot(col,col_W)+self. b

    out=out. reshape(N,out_h,out_w,-1). transpose(0,3,1,2)
    #transpose 会更改多维数组的轴的顺序。
    self. x=x
    self. col=col
    self. col_W=col_W
    return out
```

Init( )函数用来初始化卷积运算所包含的 4 个变量,即:卷积核、偏置值、步长与填充。Forward( )函数中,首先确定卷积核的长、宽以及通道数量,均为 3,FN 值为卷积核个数。输入图片尺寸为 x. shape,N 为输入图片的批处理数量,输出图片尺寸为 out_h 与 out_w。之后的代码中,涉及到 python 中的 im2col( )函数,该函数能将包含批处理的多维数据转为二维数据,如图 3 所示。将卷积核与图片通过 im2col( )函数转为二维数组,通过 numpy. dot( )进行矩阵的点乘运算并加上偏置值得到输出图像,将输出经过 reshape( )函数后即得到卷积层的运算结果。

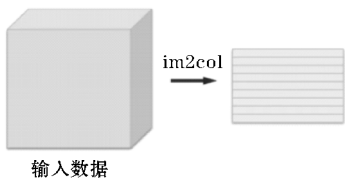

图 3　im2col( )函数原理

池化层可以进一步对图片特征进行提取,并缩小图片尺寸。本模型所涉及的池化方法为最大池化,即从目标区域中取出最大值,图 4 所示为最大池化运算的例子,输入图片尺寸 4×4,步长为 2,输出图片尺寸为 2×2。

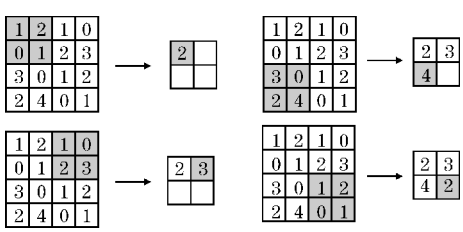

图 4　最大池化运算

```
class Pooling:
  def_nit_self,pool_h=2,pool_w=2,stride=2,pad=0):
    self. pool_h=pool_h
    self. pool_w=pool_w
    self. stride=stride
    self. pad=pad

  def forward(self,x):
    (N,C,H,W)=x. shape
    out_h=int(1+(H-self. pool_h)/self. stride)
    out_w=int(1+(W-self. pool_w)/self. stride)
    col=im2col(x,self. pool_h,self. pool_w,self. stride,self. pad)
    col=col. reshape(-1,self. pool_h * self. pool_w)

    arg_max=np. argmax(col,axis=1)
    out=np. max(col,axis=1)
    out=out. reshape(N,out_h,out_w,C). reshape(0,3,2,1)

    self. x=x
    self. arg_max=arg_max
    return out
```

依据优化 VGG19 模型池化层的特点,输入图片尺寸设为 H×W,N 为批处理数量,通道数(变量 C)为 3,池化框大小(变量 pool_h×pool_w)为 2×2,池化操作步长(变量 stride)为 2,填充值(变量 pad)为 0。池化层的 Python 代码如上个文本框所示。

Init( ) 函数对步长,填充与池化框的大小进行初始化操作。Forward( ) 函数中,输入图像的尺寸为 x. shape,之后为计算输出图片的尺寸 out_h 与 out_w,接着对输入图片进行最大池化操作,输出结果经过 reshape( ) 函数得到池化操作的最终输出。

1.5 新模型时间复杂度

优化后的 VGG19 卷积神经网络由 16 个卷积层组成,因此其时间复杂度主要来自于卷积层。对于单个卷积层来讲,其时间复杂度为

$$T(n)=O(M^2\times K^2\times C_{in}\times C_{out}) \tag{10}$$

式中,K 的值为每个卷积核的边长,M 为每个卷积核输出特征图(feature map)的边长,C_{in} 代表每个卷积核的通道数,也称输入通道数,也是上一层的输出通道数。C_{out} 为卷积层具有的卷积核个数,也即输出通道数。由此可知,每个卷积层的时间复杂度由卷积核面积、输入通道数、输出通道数以及特征图面积共同决定。而输出特征图尺寸本身又由卷积核尺寸 K、填充 Padding、步长 Stride 以及矩阵尺寸 X 这 4 个参数所决定,公式为

$$M=\frac{X-K+2\times \text{Padding}}{\text{Stride}} \tag{11}$$

VGG19 卷积神经网络整体时间复杂度公式为

$$T(n)=O(\sum_{l=1}^{16})M_l^2\times K_l^2\times C_l\times C_{l-1} \tag{12}$$

卷积神经网络的整体时间复杂度为所有卷积层的时间累加。式(12)中,l 代表神经网络的第 l 个卷积层,C_l 为神经网络第 l 个卷积层的输出通道数 C_{out},也即该层的卷积核个数。对于第 l 个卷积层而言,其输入通道数 C_{in} 就是第(l-1)个卷积层的输出通道数。

2 实验结果及分析

2.1 数据集

ISCX2012 是加拿大新布伦斯威克大学所研发出的一个网络安全数据集,内含正常请求和各种不同形式的网路攻击请求,包括内部渗透网络请求、HTTP 拒绝服务、分布式拒绝服务和强化 SSH 等,原始的数据集为 pcap,即 Wireshark 的文件存储格式。为了更方便对数据集进行分析,使用了 SpringBoot 工程将原始的数据集格式转变为 XML 的文件格式,图 5 所示为一条正常网络请求的 XML 文件格式。

```
<TestbedSatJun12>
<appName>HTTPImageTransfer</appName>
<totalSourceBytes>453</totalSourceBytes>
<totalDestinationBytes>1081</totalDestinationBytes>
<totalDestinationPackets>6</totalDestinationPackets>
<totalSourcePackets>6</totalSourcePackets>
<sourcePayloadAsBase64></sourcePayloadAsBase64>
<sourcePayloadAsUTF></sourcePayloadAsUTF>
<destinationPayloadAsBase64></destinationPayloadAsBase64>
<destinationPayloadAsUTF></destinationPayloadAsUTF>
<direction>L2R</direction>
<sourceTCPFlagsDescription>F,S,P,A</sourceTCPFlagsDescription>
<destinationTCPFlagsDescription>F,S,P,A</destinationTCPFlagsDescription>
<source>192.168.1.104</source>
<protocolName>tcp_ip</protocolName>
<sourcePort>21908</sourcePort>
<destination>130.14.29.110</destination>
<destinationPort>80</destinationPort>
<startDateTime>2010-06-12T20:45:47</startDateTime>
<stopDateTime>2010-06-12T20:45:47</stopDateTime>
<Tag>Normal</Tag>
```

图 5 ISCX2012 数据集中的一条请求(XML 格式)

2.2 实验环境

模型的开发系统为 MAC-OS X,模型的训练使用 Python 语言,使用框架为 TensorFlow 1.15.0 版本、Keras 2.2.4版本,主要依赖 Numpy 的版本为1.16.2,使用不同版本的框架与依赖可能存在不兼容的问题。

2.3 训练实验过程

模型的训练过程如图 6 所示,首先将 ISCX2012 入侵检测数据集分为训练集和测试集,ISCX2012 数据集中含有普通请求与异常请求共计 50400 条,将每一条记录转为 numpy 中的数组,同时使用 Numpy 中的函数将数据集划分为训练集与测试集,划分比率为 0.75,37800 条数据为训练集,12600 条数据为测试集,正常请求与异常请求随机分布在其中。使用 Python 中的 Matplotlib 模块将每一条数据转为流量灰度图。在实验中,将数据转为 128×128,50×50 和 256×256 这 3 种像素的流量灰度图输入到异常检测模型进行训练,以确定哪一种像素的流量灰度图能获得更好的效果。图 7 所示为随机选取 3 条像素为 50×50 的流量灰度图。

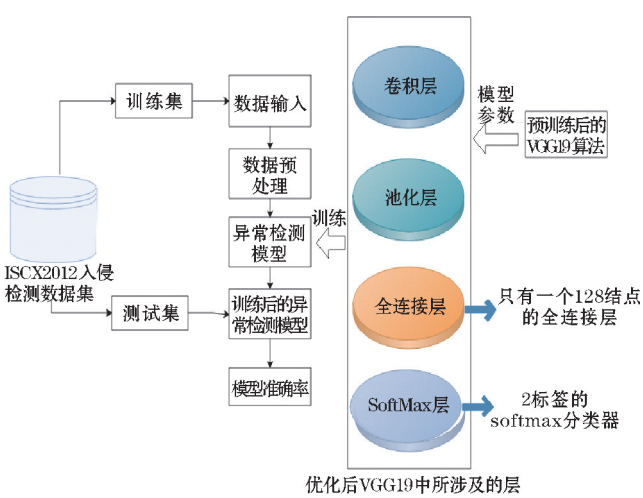

图 6 模型的训练过程

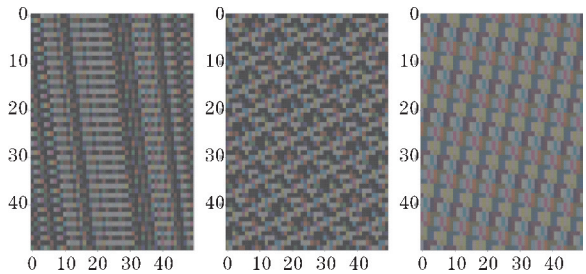

图 7　流量灰度图

优化后的 VGG19 模型卷积层与池化层的参数与权值采用 ImageNet 模型中预训练好的值,因此冻结 16 个卷积层与池化层的参数,只训练一个全连接层与 Softmax 层的参数[15]。训练集数量为 37800 条,使用 Keras 的神经网络训练框架,模型的误差函数为交叉熵,设置 batch-size 的值为 32,epoch 的值为 2,迭代次数 2500。表 1 所示为模型的每一层神经网络的结构与参数(以输入流量灰度图像素为 50×50 为例),使用 Keras 中 Model 模块的 summary( ) 函数即可获得神经网络的结构可视化。

表 1　模型的神经网络结构及参数情况

Layer(type)	Output Shape	Param
Input_1(InputLayer)	(None,50,50,3)	0
Block1_conv1(Conv2D)	(None,50,50,64)	1792
Block1_conv2(Conv2D)	(None,50,50,64)	36928
Block1_pool(MaxPooling2D)	(None,25,25,64)	0
Block2_conv1(Conv2D)	(None,25,25,128)	73856
Block2_conv2(Conv2D)	(None,25,25,128)	147584
Block2_pool(MaxPooling2D)	(None,12,12,128)	0
Block3_conv1(Conv2D)	(None,12,12,256)	295168
Block3_conv2(Conv2D)	(None,12,12,256)	590080
Block3_conv3(Conv2D)	(None,12,12,256)	590080
Block3_conv4(Conv2D)	(None,12,12,256)	590080
Block3_pool(MaxPooling2D)	(None,6,6,256)	0
Block4_conv1(Conv2D)	(None,6,6,512)	1180160
Block4_conv2(Conv2D)	(None,6,6,512)	2359808
Block4_conv3(Conv2D)	(None,6,6,512)	2359808
Block4_conv4(Conv2D)	(None,6,6,512)	2359808
Block4_pool(MaxPooling2D)	(None,3,3,512)	0
Block5_conv1(Conv2D)	(None,3,3,512)	2359808
Block5_conv2(Conv2D)	(None,3,3,512)	2359808
Block5_conv3(Conv2D)	(None,3,3,512)	2359808
Block5_conv4(Conv2D)	(None,3,3,512)	2359808
Block5_pool(MaxPooling2D)	(None,1,1,512)	0
Flatten	(None,512)	0
Dense	(None,128)	65664
Predictions(Dense)	(None,1)	129

2.4　仿真检测实验及结果

在训练集中,共迭代 2500 次,图 8 为各种不同流量灰度图像素下的准确率情况。可以看出,当迭代次数较少时,模型的学习次数不够,最后识别的准确率也比较低。

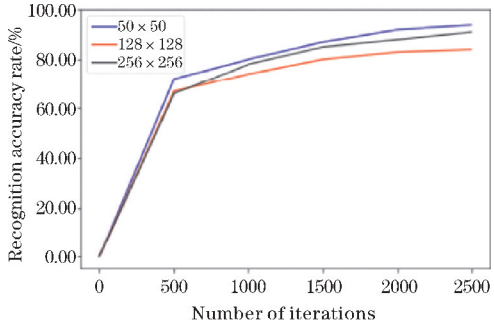

图 8　不同分辨率下流量灰度图准确率

之后,随着训练次数的增加,不同分辨率的流量灰度图的准确率都出现较大提升,但是不同分辨率的图像在准确率方面依然有差距:像素为 128×128 的流量灰度图在近 1800 次迭代后,准确率趋于稳定在 81%;而像素为 256×256 的流量灰度图在 2500 次迭代后准确率为 90%;准确率最高的图像分辨率 50×50,在进行 2500 次迭代后,准确率达到了 95%。

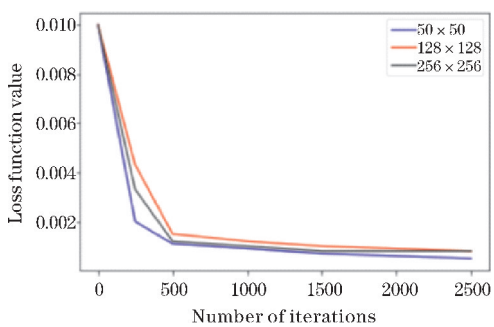

图 9　不同分辨率下流量灰度图误差情况

图 9 为不同分辨率下的误差情况,同样在分辨率为 50×50 的情况下效果最好,因此测试集将流量灰度图的分辨率设置为 50×50。共计 12600 条数据进入模型测试,准确率达到 97%,取得了比较好的效果。

3　结束语

Web 应用不断普及的今天,服务器的安全也变得越来越重要。良好的入侵检测模型可以对服务器的请求进行检测,筛选出异常请求。文中建立了基于优化后 VGG19 深度卷积神经网络的异常检测模型,在传统 VGG19 网络的基础上,基于微型迁移学习共享了预训练模型中池化层和卷积层的参数与权值。改进全连接层的结构与数量,将 Softmax 分类器的标签设置为 2,并将 ISCX2012 数据集中的数据转为分辨率为 50×50 的流量灰度图进入神经网络进行训练,取得了很好的效果,证明此模型具有实际应用的价值。

参考文献:

[1] 韩言平. SQL 注入智能检测工具的设计与实现[D]. 北京:北京邮电大学,2019.

[2] 曹棋敏. 基于 Web 日志的入侵检测模型构建[D]. 广州:华东师范大学,2019.

[3] 冯秋燕. 基于 Web 应用的日志异常检测与用户行为分析研究[D]. 广州:华南理工大学,2019.

[4] Komviriyavut T, Sangkatsanee P, Wattanapongsakorn N, et al. Network Intrusion detection and classification with decision tree and rule based approach [C]. Communications and Information Technology, ISCIT 2009. 9th International Symposium on IEEE, 2009:2046-1050.

[5] Sahu S, Mehtre B M. Network intrusion detection system using J48 Decision Tree[C]. Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on IEEE, 2015:2023-2026.

[6] Livadas C, Walsh R, Lapsley D, et al. Using machine learning Techniques to identity botnet traffic [C]. Local Computer Networks, Proceedings 2006 31st IEEE Conference on IEEE, 2006:967-974.

[7] Reddy R R, Ramadevi Y, Sunitha K V N. Enhanced anomaly Detection using ensemble support vector machine[C]. International Conference on Big Data Analytics & Computational Intelligence, 2017:300-303.

[8] 赵夫群. 基于混合核函数的 LSSVM 网络入侵检测方法[J]. 现代电子技术,2015,38(21):96-99.

[9] 张宝安. 基于深度学习的入侵检测研究与实现[D]. 北京:北京邮电大学,2019.

[10] 彭鑫. 基于卷积神经网络的特定场景下人脸识别研究[D]. 西安:西安理工大学,2019.

[11] 熊梦渔. 基于深度神经网络的图像融合方法研究[D]. 无锡:江南大学,2019.

[12] Peter Harrington. Machine Learning in Action [M]. Manning Publications, 2012.

[13] W Jin, Z J Li, L S Wei, et al. The improvements of BP neural network learning algorithm[J]. International Conference on Signal Processing Proceedings. IEEE, 2002(3):1647-1649.

[14] 李媛媛. 基于深度学习的图像分类及目标定位[D]. 北京:北京邮电大学,2018.

[15] 张成. 小样本的入侵检测技术研究与应用[D]. 成都:电子科技大学,2019.

Research on Anomaly Detection Model based on Optimized VGG19 Convolutional Neural Network

WANG Wenwen, TAO Hongcai

(College of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: Internet service has become an essential part of people´s life, but due to the increasing ways of network attacks, network security problems become increasingly serious. Anomaly detection is a very effective way to detect network attacks. In this paper, an anomaly detection model based on optimized VGG19 convolutional neural network is established, and the train and test on ISCX2012 dataset are conducted. The experiment results show that the novel model has a good detection effect for abnormal requests.

Keywords: network attack; anomaly detection; convolutional neural network; VGG19