doi: 10.3969/j.issn.0490-6756.2017.05.011

基于卷积神经网络的网络流量识别技术研究

李 勤[1], 师 维[1], 孙界平[1], 董 超[1], 曲天舒[2]
(1. 四川大学计算机学院, 成都 610065; 2. 英国利物浦大学)

摘 要: 近年来,深度包检测技术和基于统计特征的网络流量识别技术迅速发展,但它们分别存在不能识别加密流量和依赖人对特征主观选择的缺陷. 文章提出了基于卷积神经网络的流量识别方法,将网络数据按照一定的规则转换为灰度图像进行识别,并根据 TCP 数据包的有序性和 UDP 数据包的无序性,对原始的网络数据进行了扩展,以进一步提高识别率. 实验数据表明,该方法对应用程序和应用层协议两个层次的网络流量具有较高的检测率.

关键词: 网络流量; 流量识别; 卷积神经网络; 深度学习
中图分类号: TP391 **文献标识码**: A **文章编号**: 0490-6756(2017)05-0959-06

The research of network traffic identification based on convolutional neural network

LI Qin[1], *SHI Wei*[1], *SUN Jie-Ping*[1], *DONG Chao*[1], *QU Tian-Shu*[2]
(1. College of Computer Science, Sichuan University, Chengdu 610065, China; 2. University of Livepool, U.K.)

Abstract: In recent years, the deep packet inspection technology and traffic identification technology based on the statistical characteristics of data packet have developed rapidly. But they have some disadvantages. The deep packet inspection technology can't identify the encrypted network traffic, and the other technology heavily relies on subjectively chosen statistical features. A network traffic identification method based on convolutional neural network algorithm is proposed in this paper. According to certain rules, the network data is converted to gray images. In order to improve the recognition rate, the original network data is extended according to the order of the TCP packets and the disorder of the UDP packets. Experimental data shows that this method has a high detection rate both in the application and application layer protocol.

Keywords: Internet traffic; Traffic identification; Convolutional neural network; Deep learning

1 引 言

网络流量识别技术是提高网络服务质量和保障网络安全的关键技术之一. 网络流量识别技术主要包括基于端口的流量识别技术、基于网络行为特征的流量识别技术、深度包检测技术和基于统计特征的流量识别技术. 基于端口和行为特征的流量识别技术存在很多局限性. 近几年网络流量识别技术研究主要集中在深度包检测技术和基于统计特征的流量识别技术. 深度包检测技术主要研究应用程序的指纹特征选择与其他网络检测技术的结合. 基于统计特征的流量识别技术主要研究如何选择适当的统计特征. 特征选择的好坏对网络流量识别率有很大的影响.

收稿日期: 2016-06-30
基金项目: 国家自然科学基金(61332006)
作者简介: 李勤(1987—), 女, 重庆人, 硕士生, 研究方向为智能信息处理. E-mail: liqinscu@scu.edu.cn.
通讯作者: 孙界平. E-mail: sunjieping@scu.edu.cn

L7-filter[1]、nDPI[2]和 Libprotoident[3]软件是目前识别率较高的流量识别软件. 其流量识别方法综合了多种流量识别技术,主要选取数据流中部分有效负载数据与已知应用程序的数据流中的特征指纹进行匹配. 若匹配成功则识别为对应的应用程序,若匹配不成功则根据少量的统计特征,如数据流中第一个数据包的大小、整个数据流的大小、传输层协议标识和端口号等,进行识别. 这种方法的识别率依赖于应用程序的特征指纹的选择,不能识别未知流量.

Moore 等人[4]提出了 248 个统计特征作为流量识别的分类依据. 后续的研究在这些特征的基础上使用不同的机器学习算法进行流量识别. Bujlow 等人[5]基于 C5.0 的决策树算法,使用上下行流量比、上下行数据包比、有效负载字节数小于 50 字节和大于 1300 字节的数据包比例等特征进行流量识别. Riyad 等人[6]使用 AdaBoost 算法对加密流量分类,针对 SSH、Skype 等加密流量选择了上下行数据包的平均到达时间、数据包的平均大小、数据流中的平均数据包个数等特征进行识别. Alhamza 等人[7]将 K 均值、K 近邻、最大期望等算法用于流量识别,并进行了对比,文中同样使用了数据包大小、上下行流量比等统计特征. Thomas 等人[8]则通过分析网络节点的连接行为进行流量识别. Liu 等人[9]根据 IP 地址和端口的个数比等特征进行流量识别. 统计特征选择好坏对上述识别方法的识别率有一定的影响.

本文提出了基于卷积神经网络的流量识别方法. 该方法首先将网络流的数据组成灰度图像,再以图像处理的方法进行处理. 该方法可以自动提取网络流数据的特征,有效缓解识别率依赖于人为选择特征的不足,同时通过合理的组合网络流中的数据包,提高了网络流量识别的准确率.

2 卷积神经网络流量识别方法

2.1 识别流程

基于卷积神经网络的流量识别方法其识别流程如图 1 所示.

以捕获的网络数据 trace 作为数据源,首先根据协议指纹把数据包分为 TCP 和 UDP 两部分. TCP 数据包根据 TCP 建立连接和断开连接的握手信息将其组成完整的数据流. UDP 数据包根据不同的五元组进行划分,按照固定的时间间隔生成 UDP 流. 然后将生成的数据流按照一定的规则转换为灰度图像,作为卷积神经网络的输入. 最后连接 Softmax 分类器按照应用程序或者应用层协议类别完成识别.

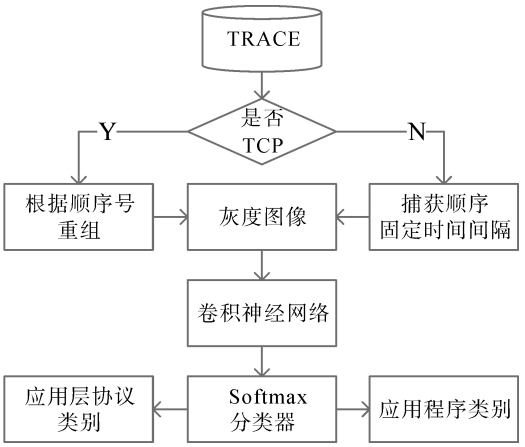

图 1 卷积神经网络流量识别流程图

Fig. 1 The flow chart of the identification of convolution neural network traffic

2.2 网络数据转换为灰度图像

数据在网络中以字节流的方式传输,每个字节的取值范围为[0,255]. 这恰好与灰度图像的像素取值范围一致. 通常情况下,每个网络流包含多个数据包,以每个网络流为基本单位,提取数据包的有效负载,以每个数据包的数据作为行,同一网络流的多个数据包作为列,组成灰度图像. 不同类别的网络流组成的灰度图像如图 2 所示.

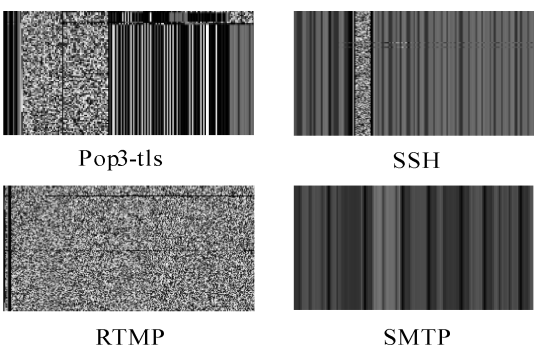

图 2 不同网络流的可视化表示

Fig. 2 Visualization of different network flows

2.3 数据集扩展

通常情况下,数据集的大小一定程度上决定了算法训练的好坏. 由于本文数据集以网络流为基本单位,即网络流与类标一一对应. 以网络流作为基本分类单位的深度包检测技术,通常使用网络流前一部分一定数量的有效负载. 每个网络流中包含多个数据包,网络传输具有时间不确定性,实际捕获数据包的顺序与发送方或接受方的数据包顺序很可能不同. 虽然 TCP 协议在接受端进行数据包的重组,但

UDP 协议的数据包顺序却不能保证.针对不同的传输层协议,同一个网络流可以生成不同的输入数据,扩展了数据集的规模.对于数据集的扩展本文根据不同的传输层协议采用不同方法进行研究.

(1) TCP 协议,根据 TCP 连接和断开的标识对数据包进行重组,保证数据包的正确顺序.如取每个数据流前 24 个数据包的前 24 个字节组成输入数据.不足 24 个数据包的网络流或有效载荷不足 24 字节的数据包补 0.此外,对于大于 24 个数据包的 TCP 流,可以随机选取一个位置,然后选取之后连续的 24 个数据包作为卷积神经网络输入.

(2) UDP 协议,根据数据包默认的顺序,即捕获的时间顺序,如取数据流前 24 个数据包的前 24 个字节,组成输入数据.除了采用 TCP 扩展数据集的方式外,还采用随机选取数据包的扩展方式.根据网络传输的时间不确定性,每个网络流随机选取 24 个数据包,将这些数据包依次排列组成灰度图像.随机选取数据包的方法符合网络传输的实际情况,同时可以扩展数据集的规模,提高网络流量的识别率.

3　流量识别的卷积神经网络结构设计

卷积神经网络沿用了传统神经网络的多层结构[10,11],主要应用于图像[12]、音频识别等方面.每层由多个特征图(Feature Map)组成,每个特征图表示一个特征,主要包含两个类型的层,即卷积层和下采样层.这二者与其相邻的层之间通常不采用全连接方式,而层与层之间的映射也是非线性的.卷积神经网络通过局部感知、权值共享、池化等方法,减少训练参数.

3.1　卷积神经网络结构设计

本文选取每个网络流中数据包负载的前 24 个字节,取 24 个不同数据包负载组成一幅 24×24×1 的灰度图像,"1"表示图像的通道数.具体网络结构如下.

C1 层:卷积核的大小表示为 $kernel\_width \times kernel\_height \times kernel\_channels$ 其中,$kernel\_width$、$kernel\_height$ 分别表示卷积核的宽和高;$kernel\_channels$ 表示卷积核的通道数;C1 层中设置卷积核为 5×5×1,步长(stride)为 1,填充(padding)为 2,共有 8 个卷积核,步长指每次卷积后卷积核移动的像素位数,填充指图像的四个方向填充 0 的行(列)数.设输入数据的大小为 $W_1 \times H_1 \times D_1$,输出特征图的大小为 $W_2 \times H_2 \times D_2$,根据输入数据的大小和卷积核大小可以确定输出特征图大小计算方法,如式(1)所示.

$$W_2 = \frac{(W_1\text{-}kernel\_width + padding \times 2)}{stride} + 1$$

$$H_2 = \frac{(H_1\text{-}kernel\_height + padding \times 2)}{stride} + 1$$

$$D_2 = \text{卷积核数量} \tag{1}$$

经过 C1 层后,每个 24×24×1 的输入图像生成 24×24×8 的特征图.每个卷积核附加一个偏置参数,C1 层共有 8×(5×5×1+1)=208 个参数.C1 层采用"Relu"激活函数.

S2 层:此层为下采样层,即池化层.S2 用来减小 C1 层生成特征图的大小,同时增强了整个网络对噪音和扰动的鲁棒性.池化窗口大小表示为 $pool\_width \times pool\_height$,本文设置为 2×2,步长为 2,填充为 0,池化方式采用最大池化,此层的输入为 24×24×8 大小的特征图,输出为 12×12×8 大小的特征图.计算方法如式(2)所示.

$$W_2 = \frac{(W_1\text{-}pool\_width + padding \times 2)}{stride} + 1$$

$$H_2 = \frac{(H1\text{-}pool\_height + padding \times 2)}{stride} + 1$$

$$D_2 = D_1 \tag{2}$$

C3 层:C3 层为第 2 个卷积层,与 C1 层类似.卷积核大小设置为 5×5×1,步长为 1,填充为 2,共有 16 个卷积核.其输入为 S2 的输出,大小为 12×12×8 的特征图.输出特征图大小为 12×12×16.同样,每个卷积核附加一个偏置参数,C3 层共有 16×(5×5×8+1)=3216 个参数.

S4 层:为第 2 个下采样层,设置池化窗口大小为 3×3,步长为 3,填充为 0,池化方式采用最大池化,此层的输入为 12×12×16 大小的特征图,输出为 4×4×16 大小的特征图.

R5 层:为光栅化层,将 S4 层的输出特征图,按顺序排列为 4×4×16=256 个节点.同理,其残差也按相应的顺序排列为 256 个节点.

F6 层:F6 层为全连接层,其每个节点与 R5 层的每个节点均有连接,F6 层设置 10 个节点,共有 10×(256+1)=2570 个参数.F6 层连接 Softmax 分类器,其输出值为样本属于每一类的概率.

3.2　卷积神经网络参数更新

卷积神经网络的前向传播过程与传统神经网络是相同的,不同点在于连接方式不同.传统神经网络层与层之间是全联通网络,而卷积神经网络包含卷积层、下采样层等.虽然连接方式不同,但前向

传播都是在当前的权值参数下由输入层依次计算各层的激活函数值作为下一次输入,直到输出层.而反向传播的参数更新过程则略有不同.传统神经网络除输出层外,其它各层的参数更新方式一致,卷积神经网络仅仅在最后的输出层和全连接层与传统神经网络的参数更新方法一样.卷积层和下采样层参数更新方法根据不同的连接方式采用不同的参数更新方法.

卷积神经网络采用梯度下降法进行参数更新,通过前向传播计算输出结果和损失函数,然后通过反向传播更新参数.对样本$(x^{(i)},y^{(i)})$,其损失函数如式(3)所示.

$$J(W,b;x^{(i)},y^{(i)})=\frac{1}{2}\|h_{W,b}(x^{(i)})-y^{(i)}\|^2 \tag{3}$$

(a) 卷积层:卷积层和下采样层之间是部分联通网络.前向传播时,由卷积核对输入该层的特征图或原始图像进行卷积操作,然后通过激活函数运算,生成一个特征图.每个输出特征图可能由多个输入特征图与卷积核操作组合而成,如式(4)所示.

$$X_j^{(l+1)}=f\Big(\sum_{i\in M_j}X_i^{(l)}*W_{ij}^{(l)}+b_j^{(l)}\Big) \tag{4}$$

其中,"$*$"表示卷积运算;M_j 表示所选择输入特征图的集合.对不同的输出特征图,如果包含相同的输入特征图,则输入特征图的卷积核是不同的,即如果 $X_j^{(l+1)}$ 和 $X_k^{(l+1)}$ 的输入特征图中都存在 $X_i^{(l)}$,则其对应的 $W_{ij}^{(l)}$ 与 $W_{ik}^{(l)}$ 是不同的.

假设第 $l+1$ 层为下采样层,其对应的残差为 $\delta^{(l+1)}$,第层为卷积层,其对应的残差为 $\delta^{(l)}$,则根据残差反向过程,由连接第 $l+1$ 层与第 $l+1$ 层的权值矩阵的转置乘以第 $l+1$ 层的残差,再乘以第层激活函数 f 对第 l 层线性组合 z 的导数.因为下采样过程中,每个输入特征图中的一小块对应于输出特征图中的一个节点,即第 l 层特征图每个池化规模的节点仅对应于第 $l+1$ 层的一个节点.因此 $\delta^{(l)}$,为了计算,首先需要将第 $l+1$ 层的残差 $\delta^{(l+1)}$ 进行上采样.记上采样过程为 $upsample(.)$,则

$$\delta^{(l)}=\beta_j^{(l+1)}(upsample(\delta^{(l+1)}).*f'(z^{(l)})) \tag{5}$$

根据下采样时采用的池化方法,使用对应的上采样方法.

(1) 最大池化:将残差拷贝到对应最大值节点,其他节点为 0.或者将残差拷贝到对应池化规模的所有节点上.

(2) 均值池化:将残差平均到池化规模的所有节点上.

那么,第 l 层第 i 个特征图与第 $l+1$ 层第 j 个特征图之间的导数可由式(6)计算.

$$\frac{\partial J(W,b)}{\partial W_{ij}^{(l)}}=\delta_j^{(l+1)}*(X_i^{(l)})^T$$

$$\frac{\partial J(W,b)}{\partial b_j^{(l)}}=\sum_{u,v}\delta_j^{(l+1)} \tag{6}$$

其中,u,v 表示第 $l+1$ 层第 j 个特征图的节点维度.

(b) 下采样层:对下采样层的 N 个输入特征图,生成 N 个输出特征图,记下采样过程为,前向传播过程计算方法如式(7)所示.

$$X_j^{(l+1)}=f\Big(\sum_{i\in M_j}\beta_j^{(l)}downsample(X_i^{(l)})+b_j^{(l)}\Big) \tag{7}$$

对下采样过程 $f(.)$通常为线性激活函数,每个下采样存在一个乘积因子 β 和偏置输入项 b.假设第 $l+1$ 层为卷积层,其对应的残差为 $\delta^{(l+1)}$,第 l 层为下采样层,其对应的残差为 $\delta^{(l)}$,则

$$\delta_i^{(l)}=\sum_{j\in M_i}\delta_j^{(l+1)}*W_{ij}^{(l)} \tag{8}$$

其中,表 M^i 示第层第 i 个节点参与输入的所有第 $l+1$ 层节点的集合.由于下采样层激活函数为线性激活函数,所以公式没有相应的导数项.第层第 i 个特征图与第 $l+1$ 层第 j 个特征图之间的导数可由式(9)计算.

$$\frac{\partial J(W,b)}{\partial \beta_j^{(l)}}=\delta_j^{(l+1)}*(downsample(X_i^{(l)}))^T$$

$$\frac{\partial J(W,b)}{\partial b_j^{(l)}}=\sum_{u,v}\delta_j^{(l+1)} \tag{9}$$

4 实 验

4.1 实验环境与数据

本文使用的数据集由文献[13]的作者提供.该数据集在一个由 8 台计算机组成的网络内生成,包括 3 台 Windows 7 主机,3 台 Ubuntu 主机,1 台 Windows XP 主机,还有 1 台 Ubuntu 服务器.该数据集最大可能的模仿真实的网络行为,包括多种常用的应用程序,如 BitTorrent、FTP、PPLive 和一些游戏软件等,同时包含常用的应用层通信协议,如 HTTP、DNS、SMTP 等.数据集的捕获由 VBS 工具包[14]生成,同时生成进程名称、五元组等相关信息.类标通过这些信息人工生成,确保了类标的准确性.其标签格式具体如下:

流标示＃起始时间＃终止时间＃源 IP＃目的 IP＃源端口＃目的端口＃传输层协议＃操作系统类型＃进程名称＃HTTP URL＃.

应用程序对应的流大小和流数量如表 1 所示.

表 1　应用程序流数量和大小分布

Tab. 1　Distribution table of the number and size of the application flows

应用程序	＃流数量	大小(Mb)
4Shared	144	13.39
Americas Army	350	61.15
BitTorrent clients(non-encrypted)	261527	6779.95
Dropbox	93	128.66
eDonkey clients(non-obfuscated)	13852	8480.48
Freenet	135	538.28
FTP clients(active)	126	341.17
FTP clients(passive)	122	270.46
iTunes	235	75.4
PPLive	1510	83.86
PPStream	1141	390.4
RDP Clients	153837	13257.65
Skype	2197	174.74
Sopcast	424	109.34
Spotify	178	195.15
Steam	1205	255.84
TOR	185	47.14

整个数据集包含数据包的所有有效负载，共有 767690 个数据流，其中 98.96%标记为应用程序，97.41%标记为应用层协议，共计 54.2 GB. 将其中每个应用程序中的 2/3 作为训练数据，剩下的 1/3 作为测试数据. 本文主要以网络流的检测率作为流量识别方法的评价标准.

4.2　实验过程与结果分析

4.2.1　*扩展数据集与原始数据集对比实验*　根据上文建立的卷积神经网络模型，对数据集在应用程序类别层次进行训练和分类. 激活函数采用 Relu 函数，分类器采用 Softmax 分类器，其输出层与前一层均为全联通网络，其损失函数的优化算法均使用随机梯度下降算法. 对比数据集在扩展前后的分类效果如图 3 所示.

卷积神经网络结构参考了其在手写体识别上的应用，共有 2 个卷积层-下采样层，最后以全连接的方式连接 Softmax 分类器. 实验数据显示，通过数据集扩展，有效提升了算法的分类效果. 该算法在原始数据集上测试集检测率为 63.12%，在扩展数据集的检测率为 79.96%，较原始数据集提升较明显. 再经过一定迭代次数，算法基本收敛. 扩展后的数据集由于同一类别数据包具有很大的相似性，近似于图像平移生成的数据集，而卷积神经网络的卷积层和下采样层对图像的平移、缩放和形变保持不变性，因此扩展后的数据集能有效提高算法的准确率和鲁棒性.

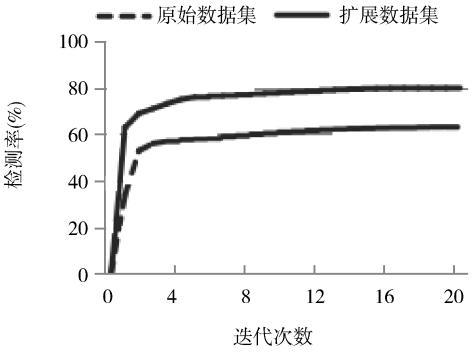

图 3　扩展前后网络数据流检测率对比

Fig. 3　Contrastive figure of the network data flow detection rate between before and after extention

4.2.2　*与其它流量识别方法对比*　通过本文提出的方法与 nDPI 和 LibProtoident 两种深度包检测技术进行对比，验证本文方法的有效性. 不同算法对应用层协议识别的检测率如表 2 所示；不同算法对应用程序识别的检测率如表 3 所示.

表 2　不同算法的检测结果(应用层协议)

Tab. 2　Detection result between different algorithms

应用层协议	DNS	POP3-TLS	RTMP	NETBIOS Name Service	SMTP-TLS	SOCKSv5	SSH
NDPI	100.00	88.12	70.90	99.97	3.85	92.99	93.98
LibProtoident	99.96	100.00	86.51	0.04	100.00	100.00	94.19
本文算法	100.00	95.13	100.00	100.00	76.00	97.63	99.83

表 3　不同算法的检测结果(应用程序)

Tab. 3　Detection result between different algorithms

应用程序	BitTorrent	Sopcast	Spotify	Steam	TOR	PPStream
NDPI	54.41	63.68	0.56	76.02	33.51	0.53
Libprotoident	60.31	46.70	0.56	75.85	33.51	0.96
本文算法	95.22	93.26	43.50	95.87	86.52	97.75

从表 2 和表 3 可以看出,本文算法以原始网络数据作为输入,通过卷积神经网络提取特征,对多数应用程序和应用层协议都具有较好的分类效果,可以识别部分加密协议.较其他两种流量识别方法在检测率上有一定的提高,证明了本文算法的有效性.

5 结 论

本文根据相同协议和相同应用程序的网络流量的相似性、数据包之间的关联性和数据流具有灰度图像的局部相关性特点,提出了基于卷积神经网络的流量识别方法,并对原有的数据集进行了扩充,一定程度上提高了网络流量识别率.但对部分加密协议的识别率偏低,该方法仍有很大的提升空间,下一步工作主要针对加密协议流量进行识别研究.

参考文献:

[1] Levandoski J, Sommer E, Strait M, *et al*. Application layer packet classifier for linux [EB/OL]. (2009-07-07). [2016-01-02]. http://l7-filter.sourceforge.net/.

[2] Ntop Company. Ntop:open and extensible LGPLv3 deep packet inspection library[EB/OL]. (2015-08-18). [2016-01-02]. http://www.ntop.org/products/deep-packet-inspection/ndpi/

[3] Edgewall Software. Libprotoident[EB/OL]. (2011-10-18). [2016-01-02]. http://wand.net.nz/trac/libprotoident

[4] Moore A, Zuev D, Crogan M. Discriminators for use in flow-based classification[EB/OL]. (2005-08-08). [2016-01-02]. http://www.cl.cam.ac.uk/~awm22/publications/moore2005discriminators.pdf.

[5] Bujlow T, Riaz T, Pedersen J M. A method for classification of network traffic based on C5.0 cachine learning algorithm[C] //Proceedings of 2012 International Conference on Computing, Networking and Communications. Maui, Hawaii, USA: IEEE, 2012.

[6] Alshammari R, Zincir-Heywood A N. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection [J]. Comput Netw, 2011, 55: 1326.

[7] Alalousi A, Razif R, AbuAlhaj M, *et al*. A preliminary performance evaluation of K-means, KNN and EM unsupervised machine learning methods for network flow classification[J]. Int Elec Comput Eng, 2016, 6: 778.

[8] Karagiannis T, Papagiannaki K, Faloutsos M. BLINC: multilevel traffic classification in the dark [J]. ACM Sigcomm Comput Commun Rev, 2005, 35: 229.

[9] Liu H, Feng W, Huang Y, *et al*. A peer-to-peer traffic identification method using machine learning [C]//Proceedings of International Conference on Networking, Architecture, and Storage. Guilin, China: IEEE, 2007.

[10] 冯晓伟,孔祥玉,马红光,等. 一种主奇异三元组提取的快速神经网络算法[J]. 四川大学学报:自然科学版,2016, 53: 572.

[11] 杨可心, 桑永胜. 基于 BP 神经网络的 DDoS 攻击检测研究[J]. 四川大学学报:自然科学版,2017, 54: 71.

[12] 赵鹏, 王斐, 刘慧婷, 等. 基于深度学习的手绘草图识别[J]. 四川大学学报:工程科学版,2016, 48: 94.

[13] Bujlow T, Carela-Español V, Barlet-Ros P. Independent comparison of popular DPI tools for traffic classification [J]. Computer Networks, 2015, 76: 75.

[14] Bujlow T, Balachandran K, Riaz M T, *et al*. Volunteer-based system for classification of traffic in computer networks[C]//Proceedings of 19th Telecommunications Forum. Belgrade, Serbia: TELFOR, 2011.