Journal of Computer Applications
计算机应用, 2020, 40(4): 1109 - 1114
ISSN 1001-9081
CODEN JYIIDU
2020-04-10
http://www.joca.cn

文章编号:1001-9081(2020)04-1109-06
DOI:10.11772/j.issn.1001-9081.2019081380

面向移动应用识别的结构化特征提取方法

沈　亮,王　鑫,陈曙晖*
(国防科技大学 计算机学院,长沙 410073)
(*通信作者电子邮箱 shchen@nudt.edu.cn)

摘　要:针对移动应用流量监控及行为分析等需要,为有效识别移动网络流量所属的应用,提出一种超文本传输协议(HTTP)流结构化特征提取方法。采取一款自研的基于虚拟专用网络(VPN)的流量采集工具获取研究数据,该工具能够精确标识每一条数据流归属的应用。在特征提取阶段,不预先设计特征构成,通过流聚类、获取最长公共子序列、字符替换得到应用HTTP流的结构化特征。从42种应用的117 772条HTTP流中提取特征,并对测试集的50 387条HTTP流进行识别,所提方法的平均准确率达99%,平均查全率为90.63%,单个应用最大误报率为0.52%。实验结果表明,该结构化特征提取方法能够有效识别移动应用的流量。

关键词:流量采集;移动应用识别;流量分类;深度包检测;特征提取
中图分类号:TP393.08　　**文献标志码**:A

Structural signature extraction method for mobile application recognition

SHEN Liang, WANG Xin, CHEN Shuhui*
(*College of Computer, National University of Defense Technology, Changsha Hunan* 410073, *China*)

Abstract: Focusing on the needs of mobile application traffic monitoring and behavior analysis, a Hyper Text Transfer Protocol (HTTP) traffic structural signature extraction method was proposed to effectively identify the application to which mobile network traffic belongs. A self-developed Virtual Private Network (VPN)-based traffic collection tool was used to obtain the research data, which was able to accurately identify the application that each data stream belongs to. In the signature extraction stage, the signature composition was not pre-designed, and the structural signatures of the HTTP traffic were obtained through three steps of flow clustering, obtaining the longest common subsequence and character substitution. The signatures of 42 applications were extracted from 117 772 HTTP traffic to identify 50 387 HTTP traffic in test set. The proposed algorithm has the average accuracy of 99%, the average recall of 90.63%, and the maximum false positive rate of single application of 0.52%. The experimental results show that the proposed structural signature extraction method can effectively identify the traffic of mobile applications.

Key words: traffic collection; mobile application identification; traffic classification; Deep Packet Inspection (DPI); signature extraction

0　引言

随着移动互联网基础设施建设不断优化升级以及智能手机的快速普及,我国形成了全球最大的移动互联网应用市场。中国互联网络信息中心(China Internet Network Information Center, CNNIC)发布的第43次《中国互联网络发展状况统计报告》显示,截至2018年12月,我国市场上监测到的移动应用程序在架数量为449万款[1]。面对数量庞大的移动应用,如何高效、准确地识别这些应用的流量,对于网络运营和管理机构具有重要的意义,这是研究差异性服务、流量控制、恶意应用识别以及用户行为分析的前提和基础。

网络流量识别是指通过对网络流量的分析,确定网络流量对应的应用协议,并基于此对网络流量进行分类。在传统的互联网平台上,主要通过端口识别[2]、深度包检测(Deep Packet Inspection, DPI)[3-4]、基于主机行为或流量行为的识别技术[5-6]、协议逆向[7]和机器学习[8-10]等技术来实现。传统网络识别大多只能进行粗粒度的流量识别,如网络流对应的应用层协议、恶意流量识别、异常流量检测等。

在传统网络流量识别技术的基础上,很多研究工作专门针对移动应用的特点提出了相应的识别方法。当前的主要研究方向是对应用超文本传输协议(Hyper Text Transfer Protocol, HTTP)流的识别。这是因为绝大部分移动应用都是通过HTTP和超文本传输安全协议(Hyper Text Transfer Protocol over Secure socket layer, HTTPS)与服务器进行通信[11],而这两种协议的实现机制不同,需要分开研究。Xu等[12]通过提取移动应用网络流量中的应用标识符(唯一标识应用的数字或字符串,如Youku、taobao_android等)来识别应用流量。他们根据互联网服务提供商提供的网络流量,对移动应用特征进行了大规模研究,提出使用HTTP报文中的User-Agent字段来识别应用程序。但是,Tongaonkar等[13]在对超过10万个Android和iOS应用程序研究后发现,iOS系统的许多应用程序都遵循在User-Agent字段放置应用标识符的规则,但Android系统的应用程序并没有强制遵循这一规则。因此,该方法并不适用于识别Android应用。

收稿日期:2019-08-08;**修回日期**:2019-10-10;**录用日期**:2019-10-28。　**基金项目**:国家重点研发计划项目(2016QY11W2004)。
作者简介:沈亮(1989—),男,安徽蚌埠人,硕士研究生,主要研究方向:移动应用流量分类与识别;　王鑫(1991—),男,山东济南人,博士研究生,主要研究方向:网络空间安全;　陈曙晖(1974—),男,湖南益阳人,研究员,博士,主要研究方向:网络空间安全、网络体系结构、高速互联网监测。

Dai等[11]构建了一个应用特征生成系统NetworkProfiler。应用特征有两个组成部分：第一部分由主机名Host组成；第二个部分是将HTTP请求行中的请求方法（Get/Post/Head等）、请求路径名和查询关键字及其值域中的固定不变内容转换为状态机。NetworkProfiler只是获取HTTP报文请求行中的固定字符串和Host作为应用的特征，存在以下两个问题：1）当前主流应用朝着体系化、平台化方向发展，应用相互集成，如手机QQ中集成了QQ空间、微视、QQ音乐、京东购物等。当多个关联应用从同一个服务器获取数据时，产生的报文在Host、请求行等位置可能完全一致，NetworkProfiler忽略了其他位置可能存在的有用信息，难以有效识别关联应用的流量。2）为了对抗网络监听和爬虫，应用开发人员引入可变路径技术，对请求行中的关键路径段和参数值进行编码或加密，NetworkProfiler难以有效识别这类流量。

Ranjan等[14]将应用安装包进行反编译，从配置文件中获取指定HTTP消息报头的值作为特征。这种方法不需要采集应用流量，直接从应用市场下载应用安装包即可获得研究数据。但是也存在两个比较突出的问题：1）不同应用在开发时所遵循的规范不统一，面对数量庞大的应用，难以形成有效的自动化方法将应用配置文件中的全部有用信息结合起来；2）需要人工设计应用特征的构成，可能会忽略应用自定义的HTTP报头及其内容，而这些信息是识别应用流量的关键。

也有研究者[15-16]尝试利用卷积神经网络（Convolutional Neural Network，CNN）进行准确的移动应用流量识别。将数据包转换为固定长度的向量，利用CNN提取HTTP中的抽象统计特征，并为每个应用程序建立了一个检测模型。这种做法的好处是实现了应用HTTP特征无关化，存在的主要问题有：1）需要较大的样本集才能实现较好的效果；2）模型比较复杂，难以在网络上进行在线实时检测，更适合做离线处理；3）背景流量对分类器的性能影响较大。

移动应用流量识别存在比较突出的难点，就是没有可用的移动应用网络流量集。有研究者[17]使用移动平台上的虚拟专用网络（Virtual Private Network，VPN）应用程序编程接口（Application Programming Interface，API）来获取应用程序生成的网络流量。这种方法能够将应用程序与网络流相关联，可用于构建移动应用的流量数据集。

从以上研究可以看出，对移动应用流量识别主要采用两种技术：DPI和机器学习。以上研究都能够解决一定的问题，但是都存在局限性：1）DPI和传统的机器学习算法如支持向量机、随机森林等，都需要预先设计特征，这样会丢失应用流量中广泛存在的个性化信息，可能导致识别效果不理想；2）深度学习虽然实现了特征无关化，但是模型复杂，难以进行在线实时检测，用于处理加密的HTTPS流量可能更加合适。

由于HTTP流中的字符可见，其中有足够多的可用信息，关键是如何获取这些信息来构建有效的应用流量特征。本文针对移动应用HTTP流量，提出了一种基于传统DPI技术的移动应用HTTP流结构化特征提取方法。与现有工作的不同之处在于：1）不需要预先设计特征，对数据不作特殊处理，可以保留报文中的全部特征片段，直接采用HTTP报文结构作为聚类标签，适用于所有HTTP流；2）在提取应用特征前先进行一次聚类，避免了对毫无关联的流进行操作，既便于保留报文中的共同点，也便于发现不同点；3）实验数据全部来自现实环境，结果更加可靠，通过开发一款基于Android的流量采集工具，在设备端捕获流量的同时精确地为每条数据流产生标签，此标签可以确定每一条流的归属，避免了其他流量获取方法带来的不确定性。

1 结构化特征提取系统框架

本文构建了一个基于DPI的移动应用特征提取系统，由流量采集、预处理、特征提取、特征筛选4个模块组成，如图1所示。

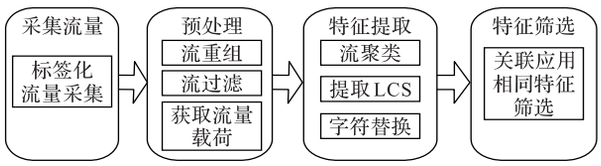

图1 移动应用特征提取系统基本框架

Fig. 1 Basic framework of mobile application signature extraction system

1）采集流量。从流量入手开展移动应用特征提取研究，首先要获取移动应用的网络流量。由于没有标准的移动应用流量数据集可供使用，研究者要独立采集移动应用的流量。采集流量必须要解决网络流的实际归属问题，即采用一定的技术手段来准确判定每条网络流是由哪个应用的产生的；否者，从不纯净的应用流量中提取的特征将存在很大的误差。本文将在2.1节介绍标签化的流量采集方法。

2）预处理阶段。根据报文的五元组信息将采集的混合网络流量进行重组，形成独立的网络流。在完成流重组后剔除非正常流以及利用HTTP报文进行DNS查询的数据流。正常的HTTP流必须具有完整的TCP连接建立过程，且服务器返回的状态码为“2XX”系列。最后获取HTTP请求报文的载荷信息，存入对应的应用程序流量库中，每条载荷信息代表一条网络流。

3）特征提取阶段。将每个应用的流聚类成具有相同结构的集合，并分别提取每个集合内所有流的最长公共子序列（Longest Common Sequence，LCS），最后替换掉LCS中的可变字段和无关信息，就形成了每一类流的字符串特征。

4）特征筛选阶段。将多个关联应用的相同特征进行筛选，根据该特征代表的网络流在不同应用中出现的频率来判定特征最后的归属。

2 关键技术实现

2.1 标签化流量采集

标签化流量采集就是通过一定的技术手段确定每一条网络流的归属。由于不同的手机操作系统原理不同，目前我们开发了一款基于Android的免Root流量采集工具NetLog，通过Android4.0+提供的VPN Service模块监听设备上所有应用的接口。NetLog在开启后会自动记录设备产生的网络流量，每隔一定时间生成一个pcap文件及相应的网络流标签文本，并压缩上传至服务器。流量标签如图2所示，包括开始的时间、应用名称、协议类型（TCP/UDP）、源IP地址、源端口号、目的IP地址、目的端口号。通过该标签，可以在后续的预处理阶段对pcap文件中的混合流量进行精确的区分，从而得到纯净的应用流量。

```
08:57:38, QQ,TCP, 10.1.10.1:41415=>61.151.225.117:80
08:57:37, QQ, TCP, 10.1.10.1:37008=>113.96.12.27:80
08:57:34,天气, TCP, 10.1.10.1:48344=>118.26.252.165:80
08:57:31,微信,TCP, 10.1.10.1:44290=>47.92.21.5:80
08:57:18,微信,TCP, 10.1.10.1:44098=>203.119.217.103:80
08:57:17,微信, TCP, 10.1.10.1:43412=>61.151.165.102:80
```

图2 Netlog流量标签

Fig. 2 Traffic labels of Netlog

2.2　结构化特征提取

移动应用操作界面很多,功能十分丰富,为了实现每一个界面的每一种功能,应用需要向对应的服务器请求数据。针对这些功能,开发人员会在应用中制定对应的数据获取计划,在应用运行过程中触发时就形成了不同类型的网络流。网络数据获取计划的内容包括采取的数据传输协议(HTTP、HTTPS等)、请求路径、各种参数名及参数值、各个字段的先后顺序、不同字段之间的分隔符等。网络数据获取计划相当于构建了一个流量框架,当触发时各个字段填充上相应的数据就构成了现实中的网络流量。由于不同公司的应用开发规范不同、不同开发人员的个人习惯不同,应用每一种功能所对应的网络数据获取计划可能存在差异,这些差异最终会体现在报文中,而这正是流量特征。

本节将介绍如何提取应用HTTP流的结构化特征。

2.2.1　流聚类

在对应用流量进行分析后发现,应用在获取不同的数据时产生的HTTP请求报文存在较大的差异。当请求方法、报文结构、服务器域名有任意一处不同时,报文可能完全不同。本文期望在提取应用流量特征时保留报文的结构,为此,需要将每个应用的HTTP流进行聚类,使每一类流趋向于相同的数据获取行为。在进行多次聚类实验及效果评估后,制定了流聚类标签,聚类标签由HTTP请求报文的请求方法、消息报头及其先后顺序、Host或路径中的域名组成,这个标签适用于任何移动应用HTTP流,当两条流的标签一致时则认为是同一类流,具体流程如图3所示。

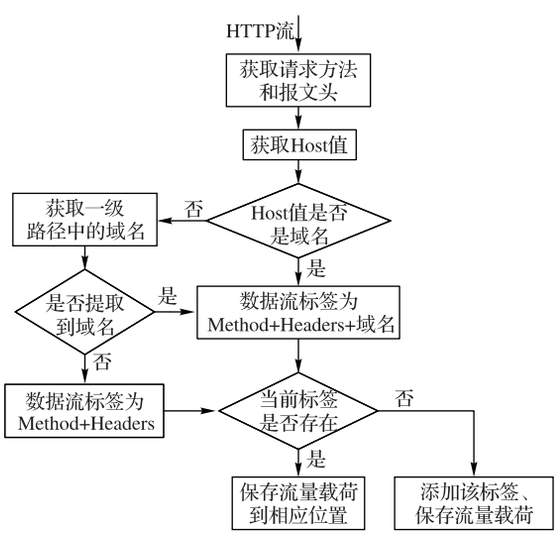

图3　流聚类流程

Fig. 3　Traffic clustering process

经过聚类,每一类的数据流已经高度相似。图4为两台不同设备上的爱奇艺产生的HTTP请求报文,根据本文的聚类原则,这两条数据流属于一类。从图4可以观察到这两条请求报文的结构完全一致,只有部分字段的值不同,可以在后续处理中提取出其中的报文结构和固定字符串作为特征。

2.2.2　特征生成

应用产生的HTTP流经过聚类后,得到了多个高度相似的集合。在提取特征时,要将每一类流中的固定不变信息保留下来。本文提出的特征生成算法是:在应用流聚类的基础上,使用LCS算法分别提取应用的每一类HTTP流的LCS。由于每一类HTTP流具有相同的结构,为了减少不必要的计算,将HTTP报文分成不同的行,即请求行、报文头行、报文体行,再分别提取每一行的子LCS后并组合成最终的结果。

```
GET /wangsucdncnc.inter.iqiyi.com/cdn/qiyiapp/20190130/0039b261e42d7/clubpage.bundle?dis_dz=CNC-HuNan&dis_st=36&wshc_tag=2&wsts_tag=5c513c7f&wsid_tag=6e355ddf&wsrid_tag=5c513c7f_yangwa&wsiphost=ipdbm HTTP/1.1(CRLF)
Connection: Keep-Alive (CRLF)
User-Agent: Android8.1.0-Xiaomi-jason(Mi Note 3)(CRLF)
Range: bytes=0- (CRLF)
Qyid: 86542203673386324aed5010c89ab17a20 (CRLF)
NetType: wifi (CRLF)
Host: 58.20.141.50 (CRLF)
Accept-Encoding: gzip (CRLF)

GET /wangsucdncnc.inter.iqiyi.com/cdn/qiyiapp/20190128/1832f65267dcc/index.android.zip?dis_dz=CNC-HuNan&dis_st=37&wshc_tag=2&wsts_tag=5c4f058b&wsid_tag=6e355ddf&wsrid_tag=5c4f058b_PShnh&wsiphost=ipdbm HTTP/1.1(CRLF)
Connection: Keep-Alive (CRLF)
User-Agent: Android9-HUAWEI-ALP(ALP-AL00) (CRLF)
Range: bytes=0- (CRLF)
Qyid: 86615803839515722e13156b5ebfd36521CZ15Z(CRLF)NetType: wifi (CRLF)
Host: 61.54.121.53 (CRLF)
Accept-Encoding: gzip (CRLF)
```

图4　爱奇艺HTTP流

Fig. 4　HTTP traffic of iQiYi

LCS算法实现简单,但存在结果碎片化的问题。为此,本文引入阈值*Length_min*来解决碎片化问题。具体思路为:在生成两个字符串的LCS状态矩阵时,当前位置字符相同,且其前后共*Length_min*个字符也相同时才计入结果。由于HTTP报文有明显的结构,不同意义的字段由一定的间隔符分隔,最短的关键字可以是1个字符,如“pt=0&t=6&tl=7&”中的关键字“t”加上前后两个分隔符“&”“=”,所以*Length_min*取值最小应为3。*Length_min*值越大,最后的结果越精简。详细描述如算法1所示。

算法1　LCS生成算法。

输入　字符串*str*1、*str*2,长度分别为*m*、*n*。

输出　最长公共子序列*LCS*。

1) 初始化二维状态数组,*flag*[*m*+1][*n*+1]保存搜索方向,*dp*[*m*+1][*n*+1]保存LCS长度
2) for *i* from 0 to *m*+1
3)　for *j* from 0 to *n*+1
4)　if *i*==0 or *j*==0, then *flag*[*i*][*j*]=0, *dp*[*i*][j]=0
5)　else if 两个字符串当前位置前后连续三个字符都相同且不是特殊分隔符, then *dp*[*i*][*j*] = *dp*[*i*−1][*j*−1] +1 and *flag*[*i*][*j*] ='diagonal'
6)　else if *dp*[*i*][*j*−1]>*dp*[*i*−1][j], then *flag*[*i*][*j*] ='left' and *dp*[*i*][*j*] =*dp*[*i*][*j*−1]
7)　else *flag*[*i*][*j*] = 'up' and *dp*[*i*][*j*] =*dp*[*i*−1][*j*]
8) *i* = *m*; *j* = *n*; 初始化LCS为空字符串　// 回溯
9) while *flag*[*i*][*j*]! = 0 do
10)　if *flag*[*i*][*j*] == 'left', then *j*−−
11)　else if *flag*[*i*][*j*] == 'up', then *i*−−
12)　else if *flag*[*i*][*j*] == 'diagonal', then do
13)　*LCS*←Append *str*1[*i*−1]
14)　*j*=*j*−1; *i*=*i*−1
15)　if *flag*[*i*][*j*]== 'left' or *flag*[*i*][*j*]== 'up'
16)　*LCS*←不连续处添加特殊间隔符(如'~')
17) 反转*LCS*字符串,并返回*LCS*

根据本文的特征生成算法,提取图4中两条爱奇艺HTTP流的LCS,其结果如图5所示。可以看出,结果保留了图4两条HTTP请求报文中符合本文要求的公共字符串序列,且保留了报文的结构,由于非连续处插入了特殊间隔符,可以在后续处理中剔除可变化的字段。

```
GET /wangsucdncnc.inter.iqiyi.com/cdn/qiyiapp/201901`?dis_dz=
CNC`&dis_st=3`&wshc_tag=2&wsts_tag=5c`&wsid_tag=6e355dd
f&wsrid_tag=5c`&wsiphost=ipdbm HTTP/1.1 (CRLF)
Connection: Keep-Alive (CRLF)
User-Agent: Android` (CRLF)
Range: bytes=0- (CRLF)
Qyid: 86` (CRLF)
NetType: wifi (CRLF)
Host: ` (CRLF)
Accept-Encoding: gzip (CRLF)
```

图5 图4的LCS结果示例

Fig. 5 Result example of LCS for Fig. 4

2.2.3 字符替换

将应用的每一特征中存在的可变字段和无关项进行替换，最后添加转义符将特征转化为正则表达式形式，可直接用正则匹配来识别应用流量。

1)替换可变字段。如前所述，在提取了每一组数据流的LCS后，会在每一个不连续处插入特殊间隔符，而HTTP报文具有明显的结构，可依据常用间隔符如空格、换行，以及“/”“,”“=”“&”“;”等划分成不同的字段，如果某一字段中存在特殊间隔符，则将当前字段替换为“(.*)”。

2)替换无关项。应用程序的流中通常具有某些与应用程序无关的字段，如“WIFI”“4G”“G4”“LTE”“NONE”“NULL”等，将这些字段替换为“(.*)”。

3)转义字符替换。最后保留“(.*)”不变，将各个转义字符前添加转义符“\”，需要转义的字符包括“.”“*”“?”“(”“)”等。

图6为图5所示的LCS经过字符替换后的结果，其中存在变化的字段和无关项已替换为正则表达式中代表任意字符的“(.*)”。当网络流量来源比较广泛时，就可以排除出所有的可变换字段，剩下的固定不变字段即为该类网络流的特征。

2.3 特征筛选

由于移动应用的开放性，不同应用可以从相同的服务器获取数据，所以可能存在不同应用产生完全相同的HTTP请求报文，即提取的特征无法有效识别数据流的源头。这类完全相同的流，主要涉及一些系统功能相关的数据，包括获取服务器时间、网络测试、上传日志等。如图7所示，飞猪、闲鱼、手机淘宝、手机天猫和优酷视频都能提取到这样一条特征。对于这种不能明确地区分数据流归属的特征，依据其在应用数据流中出现的频率来判定，如果在某一应用中出现的频率明显高于其他应用，则将此特征归为出现频率较高的应用；如果在各个应用中出现的频率没有明显的差异，则将此特征删除。

本文设置临界频率倍数阈值P，P代表了对提取的应用特征误报率的容忍度。P=0表明完全接受应用特征产生的误识别；P值越大，则本文方法提取的特征在实际流量识别中的误报率越低。本文将P设置为5，现实意义为如果特征A代表的流在应用1中所占比率高于其他应用5倍，则特征A归属于应用1，其他应用中的特征A删除。5是一个经验值，在实验中已经可以达到较好的效果。

```
GET /wangsucdncnc\.inter\.iqiyi\.com/cdn/qiyiapp/(.*)\?dis_dz=(.*)&di
s_st=(.*)&wshc_tag=2&wsts_tag=(.*)&wsid_tag=6e355ddf&wsrid_tag=
(.*)&wsiphost=ipdbm HTTP/1\.1 (CRLF)
Connection: Keep-Alive (CRLF)
User-Agent: (.*) (CRLF)
Range: bytes=0- (CRLF)
Qyid: (.*) (CRLF)
NetType: (.*) (CRLF)
Host: (.*) (CRLF)
Accept-Encoding: gzip (CRLF)
```

图6 爱奇艺特征示例

Fig. 6 Signature example of iQiYi

```
GET /gw/mtop\.common\.getTimestamp/\* HTTP/1\.1 (CRLF)
Connection: close (CRLF)
User-Agent: Dalvik/2\.1\.0 (.*) (CRLF)
Host: acs\.m\.taobao\.com (CRLF)
Accept-Encoding: gzip (CRLF)
```

图7 多种应用的共同特征

Fig. 7 Common signature of multiple applications

3 实验及结果分析

3.1 实验数据采集

为了评估本文的特征提取方法，在多台设备上安装采集工具NetLog，并收集2019年5月20日—6月30日产生的流量。其中6月15日前的流量作为样本集，用于提取应用的特征；6月16日—30日的数据作为测试集，用于测试实验提取的特征的识别效果。为了排除设备型号对应用特征的影响，挑选出其中至少出现在两台不同设备上的42种应用所产生的HTTP流作为实验数据。样本集共含有117 772条HTTP流，其详细分布见表1；测试集共含有50 387条HTTP流，其详细分布见表2。

表1 样本集应用及HTTP流分布

Tab. 1 Applications and HTTP traffic distribution in sample dataset

应用名	流数量	应用名	流数量	应用名	流数量	应用名	流数量	应用名	流数量	应用名	流数量
QQ空间	4 877	微视	5 481	UC浏览器	5 057	京东	2 218	嘀嗒出行	1 244	哔哩哔哩	705
QQ浏览器	3 120	手机天猫	224	聚划算	105	京东到家	497	滴滴出行	92	喜马拉雅	17 830
QQ音乐	2 354	手机淘宝	810	虾米音乐	2 990	京东金融	201	百度地图	230	天涯社区	16 754
企鹅电竞	2 897	淘票票	151	闲鱼	311	今日头条	3 022	爱奇艺	10 217	微博	436
手机QQ	6 648	口碑	732	优酷视频	1 651	抖音短视频	997	快手	855	知乎	132
腾讯视频	7 789	飞猪	219	大众点评	414	火山小视频	1 000	斗鱼直播	667	去哪儿旅行	439
微信	6 256	饿了么	76	美团	1 801	西瓜视频	1 751	虎牙直播	924	苏宁易购	3 598

3.2 评价标准

本文使用查全率(Recall)、准确率(ACCuracy, ACC)和误报率(False Positive Rate, FPR)来评价实验结果。查全率$Recall = TP/(TP + FN)$，代表正确识别的应用数据流数量占该应用数据流总数的比例；准确率$ACC = (TP + TN)/(TP + TN + FP + FN)$，代表被正确识别的数据流数量占所有流量的比例，评估的是对整体样本的分类能力；误报率$FPR = FP/(TN + FP)$代表的是实际不是A但被识别为A的比例。对于移动应用A，TP(True Positive)代表被正确识别为应用A的网络流数量；FP(False Postive)代表被错误识别为应用A的网络流数量；TN(True Negative)代表被正确识别为非应用A的网络流数量；FN(False Negative)代表未被识别出的应用A的数据流数量，如表3所示。

表2　测试集应用及HTTP流分布

Tab. 2　Applications and HTTP traffic distribution in test dataset

应用名	流数量	应用名	流数量	应用名	流数量	应用名	流数量	应用名	流数量	应用名	流数量
QQ空间	1 205	微视	1 620	UC浏览器	2 082	京东	765	嘀嗒出行	582	哔哩哔哩	624
QQ浏览器	3 016	手机天猫	153	聚划算	111	京东到家	150	滴滴出行	351	喜马拉雅	4 088
QQ音乐	850	手机淘宝	349	虾米音乐	1 429	京东金融	253	百度地图	348	天涯社区	2 659
企鹅电竞	1 515	淘票票	141	闲鱼	128	今日头条	2 326	爱奇艺	4 644	微博	974
手机QQ	2 294	口碑	579	优酷视频	1 948	抖音短视频	876	快手	564	知乎	216
腾讯视频	3 531	飞猪	148	大众点评	524	火山小视频	1 381	斗鱼直播	189	去哪儿旅行	699
微信	1 436	饿了么	74	美团	1 354	西瓜视频	590	虎牙直播	655	苏宁易购	2 966

表3　混淆矩阵

Tab. 3　Confusion matrix

预测类别	真实类别	
	应用A	其他应用
应用A	*TP*	*FP*
其他应用	*FN*	*TN*

3.3　实验结果

为验证本文提出的应用特征提取方法的有效性,使用该方法提取样本集中42种应用的HTTP流特征,并使用这些特征去识别测试集中的应用流量,得出每个应用特征的*TP*、*FP*、*FN*、*TN*值,并计算每个应用的特征在测试集上的查全率、准确率和误报率。由于提取的特征是正则表达式形式的字符串,识别时直接使用正则匹配的方法将每一个特征同应用的HTTP请求报文进行匹配即可。评估结果见表4。由表4可见,本文提出的应用特征提取方法具有良好的识别效果,其中平均准确率*ACC*达99%以上,单个应用最大误报率为QQ空间的0.52%,查全率最低为71%、最高为99%,平均查全率为90.63%。

由表4也可得出,本文方法可以有效区分具有关联性的同一体系的应用。如腾讯公司的QQ空间、QQ浏览器、手机QQ、企鹅电竞、腾讯视频、微视、微信,阿里巴巴旗下的淘宝、天猫、淘票票、口碑、飞猪、饿了么、聚划算、闲鱼,字节跳动公司的今日头条、抖音短视频、火山小视频、西瓜视频等。同一公司开发的应用具有明显的关联性,功能相互集成,本文方法可以以极低的误报率取得较高的查全率。

表4　应用特征在测试集上的评估

Tab. 4　Evaluation of application signatures on test dataset

应用名	*Recall*	*ACC*	*FPR*	应用名	*Recall*	*ACC*	*FPR*	应用名	*Recall*	*ACC*	*FPR*
QQ空间	0.962 7	0.994 0	0.005 2	UC浏览器	0.874 6	0.994 5	0.000 3	嘀嗒出行	0.953 6	0.999 5	0.000 0
QQ浏览器	0.944 0	0.994 9	0.001 9	聚划算	0.936 9	0.999 8	0.000 0	滴滴出行	0.923 1	0.999 5	0.000 0
QQ音乐	0.875 3	0.997 9	0.000 0	虾米音乐	0.980 4	0.999 4	0.000 0	百度地图	0.922 4	0.999 5	0.000 0
企鹅电竞	0.862 7	0.995 9	0.000 0	闲鱼	0.953 1	0.999 9	0.000 0	爱奇艺	0.987 7	0.998 9	0.000 0
手机QQ	0.782 0	0.987 6	0.002 6	优酷视频	0.839 8	0.993 8	0.000 0	快手	0.916 7	0.999 1	0.000 0
腾讯视频	0.760 7	0.979 5	0.004 1	大众点评	0.973 3	0.999 7	0.000 0	斗鱼直播	0.952 4	0.999 8	0.000 0
微信	0.880 2	0.996 6	0.000 0	美团	0.883 3	0.996 9	0.000 0	虎牙直播	0.761 8	0.996 9	0.000 0
微视	0.924 7	0.997 6	0.000 0	京东	0.898 0	0.997 7	0.000 7	哔哩哔哩	0.903 8	0.998 8	0.000 0
手机天猫	0.843 1	0.999 5	0.000 0	京东到家	0.880 0	0.999 6	0.000 0	喜马拉雅	0.995 8	0.999 6	0.000 0
手机淘宝	0.945 6	0.999 6	0.000 0	京东金融	0.913 0	0.996 0	0.003 6	天涯社区	0.985 3	0.999 2	0.000 0
淘票票	0.936 2	0.999 8	0.000 0	今日头条	0.913 0	0.996 0	0.003 6	微博	0.985 6	0.999 7	0.000 0
口碑	0.986 2	0.999 8	0.000 0	抖音短视	0.989 7	0.999 8	0.000 0	知乎	0.713 0	0.998 8	0.000 0
飞猪	0.945 9	0.999 8	0.000 0	火山小视	0.813 2	0.993 8	0.001 1	去哪儿旅	0.892 7	0.998 5	0.000 0
饿了么	0.783 8	0.999 7	0.000 0	西瓜视频	0.959 3	0.999 5	0.000 0	苏宁易购	0.894 8	0.993 8	0.000 0

3.4　对比实验

本节进行两组对比实验:第一组,通过改变特征筛选阶段的阈值*P*,观察它对识别结果的影响;第二组,选取其他已发表文献的应用特征提取技术与本文方法进行对比。

3.4.1　对比实验1

本文在特征筛选阶段设置了阈值*P*=5,即将多个应用出现的相同特征归属于流占比高于其他应用5倍的应用,该阈值可以较低的误报率获得较高的查全率。在对比实验1中,将阈值*P*设置为无穷大,其现实意义为:如果多个应用具有一个相同的特征,则排除此特征,从而使得在样本集上获取的特征可以唯一指向某一个应用。对比实验同样使用样本集提取特征,用测试集来验证识别效果,对比结果见表5。从表5可见,能够容忍一定程度误报率的*P*取值为5,与完全不容忍误报率的*P*取值为无穷大相比,平均查全率由88.21%提高到90.63%,但平均误报率仅由0.01%提高为0.05%。表6列出了三种结果差异较大的应用,查全率有较大幅度的提升,但误报率最高仅为0.52%。由此可见,在容忍一定误报率的前提下,可以大幅提高部分应用流量的查全率。

表5　对比实验1结果　　单位:%

Tab. 5　Result of comparative experiment 1　　unit:%

阈值*P*	平均查全率	平均误报率
5	90.63	00.05
无穷大	88.21	00.01

表6　对比实验1详细结果　　单位:%

Tab. 6　Details of comparative experiment 1　　unit:%

应用名	*P*=5		*P*=无穷大	
	Recall	*FPR*	*Recall*	*FPR*
QQ空间	96.27	0.52	41.00	0.00
京东金融	91.30	0.36	56.52	0.00
手机QQ	78.20	0.26	71.53	0.00

3.4.2 对比实验2

本节选取其他三种特征提取技术来评估本文方法：1)基于HTTP头字段中的显式应用标识符[12]；2)基于NetworkProfiler方法[11]的URL状态机及Host组合；3)应用逆向的方法[14]。前两种方法与本文方法都是从应用流量入手，根据原文的思路进行复现，从样本集中提取特征，并测试所提取的特征在测试集中的识别效果；第三种应用逆向的方法，由于不具备复现的能力，本文根据文献[14]的实现机制和实验数据进行对比分析。

由表7可见，本文方法与应用标识符的方法相比，平均查全率提高了47%，平均误报率仅为0.05%。与NetworkProfiler的方法相比，平均查全率提高了22%，平均误报率不足NetworkProfiler方法的1/25。由此可见，本文方法与其他从应用流量入手的方法相比，具有较高的查全率和较低的误报率。

表7 对比实验2结果 单位：%

Tab. 7 Result of comparative experiment 2 unit：%

方法	平均查全率	平均误报率
应用标识符	61.62	0.00
NetworkProfiler	73.96	1.27
本文方法	90.63	0.05

最后，对应用逆向的方法进行对比分析。文献[14]对应用安装包进行反编译，从配置文件中获取特定的字符串(例如服务器域名、User-Agent等)填充到统一构建的特征框架中，从而形成应用流量特征。此方法无须采集应用流量，直接利用应用安装包构建应用流量特征，其优势是便于开展大规模的应用特征提取，缺点是统一的特征框架难以充分利用配置文件中的关键信息，造成特征不够精细，难以有效区分同体系的应用流量。文献[14]的实验结果表明，安卓应用的整体流覆盖率为40.76%，引入“Application Families”概念(将具有一定关联性的应用作为一个整体)后整体流覆盖率提升为81%。由此可见，本文提出的方法对于识别具有关联性的应用流量具有明显的优势。

4 结语

本文提出了一种提取移动应用HTTP流结构化特征的方法，避免了预先设计特征带来的识别精度底、适用性差的问题，能够有效识别存在数据关联性的应用的流量。本文方法不需要对数据做特殊处理，适合开展大规模、高吞吐量的实时在线检测。

本文方法存在两点不足：1)采集流量需要人工运行应用程序完成；2)流聚类还不够精细，造成聚类后的类别较多。下一步的主要工作包括：1)优化聚类算法，在不影响特征精度的前提下，尽可能减少特征的数量；2)与应用自动化运行工具相结合，构建一个全自动的移动应用HTTP特征提取系统。

参考文献(References)

[1] 中国互联网络信息中心. 第43次中国互联网络发展状况统计报告[R]. 北京：中国互联网络信息中心，2019：2.(China Internet Network Information Center. The 43rd statistical report on the development of Internet in China[R]. Beijing：CNNIC，2019：2.)

[2] MOORE A W，PAPAGIANNAKI K. Toward the accurate identification of network applications[C]// Proceedings of the 2005 International Workshop on Passive and Active Network Measurement，LNCS 3431. Berlin：Springer，2005：41-54.

[3] TEUFL P，PAYER U，AMLING M，et al. InFeCT — Network traffic classification[C]// Proceedings of the 7th International Conference on Networking. Piscataway：IEEE，2008：439-444.

[4] WANG Y，XIANG Y，ZHOU W，et al. Generating regular expression signatures for network traffic classification in trusted network management[J]. Journal of Network and Computer Applications，2012，35(3)：992-1000.

[5] LEE S，KIM H，BARMAN D，et al. NeTraMark：a network traffic classification benchmark[J]. ACM SIGCOMM Computer Communication Review，2011，41(1)：22-30.

[6] JIN Y，DUFFIELD N，HAFFNER P，et al. Inferring applications at the network layer using collective traffic statistics[J]. ACM SIGMETRICS Performance Evaluation Review，2010，38(1)：351-352.

[7] CABALLERO J，YIN H，LIANG Z，et al. Polyglot：automatic extraction of protocol message format using dynamic binary analysis[C]// Proceedings of the 14th ACM Conference on Computer and Communication Security. New York：ACM，2007：317-329.

[8] MOORE A W，ZUEV D. Internet traffic classification using Bayesian analysis techniques[J]. ACM SIGMETRICS Performance Evaluation Review，2005，33(1)：50-60.

[9] ESTE A，GRINGOLI F，SALGARELLI L. On-line SVM traffic classification[C]// Proceedings of the 7th International Wireless Communications and Mobile Computing Conference. Piscataway：IEEE，2011：1778-1783.

[10] 赵双，陈曙晖. 基于机器学习的流量识别技术综述与展望[J]. 计算机工程与科学，2018，40(10)：1746-1756.(ZHAO S，CHEN S H. Review：traffic identification based on machine learning[J]. Computer Engineering and Science，2018，40(10)：1746-1756.)

[11] DAI S，TONGAONKAR A，WANG X，et al. NetworkProfiler：towards automatic fingerprinting of Android apps[C]// Proceedings of the 32nd IEEE International Conference on Computer Communications. Piscataway：IEEE，2013：809-817.

[12] XU Q，ERMAN J，GERBER A，et al. Identifying diverse usage behaviors of smartphone apps[C]// Proceedings of the 2011 Internet Measurement Conference. New York：ACM，2011：329-344.

[13] TONGAONKAR A. A look at the mobile app identification landscape[J]. IEEE Internet Computing，2016，20(4)：9-15.

[14] RANJAN G，TONGAONKAR A，TORRES R. Approximate matching of persistent LExicon using search-engines for classifying mobile app traffic[C]// Proceedings of the 35th IEEE International Conference on Computer Communications. Piscataway：IEEE，2016：1-9.

[15] CHEN Z，YU B，ZHANG Y，et al. Automatic mobile application traffic identification by convolutional neural networks[C]// Proceedings of the 2016 IEEE International Conference on Trust，Security and Privacy in Computing and Communications/International Conference on Big Data Science and Engineering/International Symposium on Parallel and Distributed Processing with Applications. Piscataway：IEEE，2016：301-307.

[16] ZHAO S，CHEN S. Smartphone application identification by convolutional neural network[C]// Proceedings of the 2018 International Conference on Machine Learning and Intelligent Communications，LNICST 251. Cham：Springer，2018：105-114

[17] LE A，VARMARKEN J，LANGHOFF S，et al. AntMonitor：a system for monitoring from mobile devices[C]// Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big(Internet) Data. New York：ACM，2015：15-20.

This work is partially supported by the National Key Research and Development Program of China (2016QY11W2004).

SHEN Liang，born in 1989，M. S. candidate. His research interests include mobile traffic classification and identification.

WANG Xin，born in 1991，Ph. D. candidate. His research interests include cyberspace security.

CHEN Shuhui，born in 1974，Ph. D.，research fellow. His research interests include cyberspace security，network architecture，high-speed Internet measurement.