doi: 10.3969/j.issn.0490-6756.2020.02.008

基于数据流多维特征的移动流量识别方法研究

武思齐[1], 王俊峰[2]
(1. 四川大学计算机学院, 成都 610065; 2. 四川大学空天科学与工程学院, 成都 610065)

摘 要: 随着移动互联网的快速发展,移动设备的数量激增至历史新高.从大量混杂流量中识别出移动流量并对流量进行分析,是深入研究移动互联网特性的第一步,同时可以为移动网络测量与管理、移动安全和隐私保护提供有价值的信息.本文综合整理了网络流量识别的常见方法,提出了基于数据流多维统计特征的移动流量识别方法.该方法从硬件特征、操作系统指纹和用户使用习惯三个方面提取了数据流中具有代表性的特征并对特征进行分析,使用集成学习的方法生成识别模型.移动流量的识别准确率和主流的 5 种操作系统流量分类的准确率都达到了 99%以上.本文方法比 UAFs 方法准确率提高了 8%左右.本方法提取的特征具有多维性并且具有实际意义,整合了网络层和传输层的数据流特征,相较于使用深度数据包检测的方法,基于数据流多维特征的方法同样适用于加密流量的分类.

关键词: 数据流; 移动流量识别; 操作系统分类; 机器学习; 集成学习
中图分类号: TP309.7 **文献标识码**: A **文章编号**: 0490-6756(2020)02-0247-08

Research on mobile traffic identification based on multidimensional characteristics of data flow

WU Si-Qi[1], *WANG Jun-Feng*[2]
(1. College of Computer Science, Sichuan University, Chengdu 610065, China;
2. School of Aeronautics and Astronautics, Sichuan University, Chengdu 610065, China)

Abstract: With the rapid development of mobile Internet, the number of mobile devices has surged to a record high. Recognizing and analyzing mobile traffic from a large number of mixed traffic is the first step to study the characteristics of mobile Internet. It can also provide valuable information for mobile network measurement and management, mobile security and privacy protection. This paper summarizes the common methods of network traffic identification, and proposes a mobile traffic identification method based on multidimensional statistical characteristics of data flow. This method extracts the representative features of data stream from three aspects: hardware features, operating system fingerprints and user usage habits, and analyses the features. An ensemble learning method is used to generate the recognition model. The accuracy of mobile traffic identification and five mainstream operation classification results are more than 99%. Compared with the UAFs method mentioned in this paper, the accuracy is improved by about 8%. The features extracted by this method are multidimensional and have practical significance. The features integrate the data flow characteristics network layer and transport layer.

收稿日期: 2019-07-03
基金项目: 国家重点研发计划项目(2018YFB0804503); 装备预研教育部联合基金(6141A02011607,6141A020223); 四川省重点研发计划项目(18ZDYF3867, 2017GZDZX0002)
作者简介: 武思齐(1995-), 女, 硕士生, 研究方向为网络信息安全. E-mail: wusiqi_1995@163.com
通讯作者: 王俊峰. E-mail: wangjf@scu.edu.cn

Compared with the method using deep packet inspection detection, this method is suitable for the classification of encrypted traffic.

Keywords: Data flow; Mobile traffic identification; Operating system classification; Machine learning; Ensemble learning

1 引 言

移动互联网的快速发展造成了网络流量的急剧增长,其中移动流量逐渐占据了运营网络中的主要流量. 移动流量是指由手机、平板电脑、电子书等智能移动终端设备产生的网络流量. 2016 年 11 月,全球智能手机和平板电脑的互联网使用量首次超过传统桌面设备[1]. 2018 年,仅智能手机的 web 流量在全球 web 流量中的份额已经超过了 52%[2]. 预计到 2021 年,仅智能手机将占总 IP 流量的 33%,流量增长率将达到 49%[2]. 随着移动终端的普及与发展,移动终端已经成为研究用户行为特征的一种理想的探测器[3],移动流量的海量增长给移动网络安全、网络测量和服务质量等方面带来了更多挑战. 根据剑桥大学的研究数据显示,近 87%的安卓智能手机有至少一个严重漏洞;Zimperium Lab 在 2018 年发现,黑客只需通过一条简单的短信便能对 95% 的 Android 设备发起攻击[4];2018 年全球移动设备上检测到的恶意软件安装包数量超过 600 万个;越来越多的家庭智能设备、穿戴设备需要手机实时监测数据并对其进行操控;移动设备需要保障更多的手机游戏、手机视频软件的并行操作等等. 由此可见,移动设备面临诸多领域的风险和挑战[5-6].

从大量混杂流量中识别出移动流量并对流量进行分析,是深入研究移动互联网特性的第一步,同时可以为移动网络测量与管理、移动安全和隐私保护提供有价值的信息. 传统的移动流量分类方法采用的特征取决于协议字段的差异,对于差别较小的协议特征无法区别,识别性能较低;深度数据包检测的方法在识别的过程中会遇到大量不包含特征信息的无效数据包,覆盖率低、计算代价很大并且不适用于加密流量的识别. 为了解决上述问题,本文提出了基于数据流多维统计特征的移动流量识别方法. 该方法从硬件特征、操作系统指纹和用户使用习惯三个方面提取了数据流中具有代表性的特征并对特征进行分析,使用集成学习的方法生成识别模型. 在移动流量识别的准确率和主流的 5 种操作系统流量分类的准确率都达到了 99%以上. 相较于目前其他方法[7-9],基于数据流多维特征的移动流量识别方法准确率较高. 本方法采用的特征整合了网络协议中网络层和传输层的数据流特征,适用于加密流量的识别. 特征选择考虑了硬件特征、操作系统指纹和用户使用习惯三个方面,相较于其他使用传统网络协议特征识别方法的研究[10-12],本文的特征选择更具有多维性. 同时,本文对于移动流量的识别和操作系统流量分类使用了不同类型的机器学习分类算法,对比了 bagging 和 boosting 分类算法的分类效果. 本文说明了在使用机器学习分类方法时,同样需要对使用场景进行细分,根据具体场景选择分类算法才能实现较好的分类效果.

2 移动流量识别方法

移动流量的识别问题可以通过终端操作系统的分类来实现,将携带有移动操作系统(Android、iOS 等)特征的流量识别为移动流量[13]. 目前移动流量的识别方法主要分为四类:基于端口的分析方法、网络协议特征识别方法、基于统计和机器学习的分析方法和深度数据包检测方法. 不同方法可以基于不同的流量测量粒度实现移动流量的识别.

(1) 基于端口的分析方法. 基于端口的分析方法是最基础的移动流量识别技术,端口是应用层进程的标识符号,不同端口号对应着不同应用程序. 可以通过提取报文头的端口号,与互联网数字分配机构(Internet Assigned Numbers Authority)为应用程序分配的端口列表进行对比,从而推断出网络流量的来源[14]. 但是诸如 P2P 应用等许多新的应用程序,采用的是动态端口或者采用其他协议的端口号来伪装,单纯使用基于端口的分类方法逐渐被淘汰. 胡婷等人[15]结合端口号匹配和机器学习分类方法,引入了目前较流行的自适应深度学习机制,采用输出结果可视化的自组织映射网络算法实现网络流量在应用层的分类,时间复杂度很低,实现简单,适用于在高速网络环境下的应用层协议识别. 将基于端口的识别和其他技术结合,能够兼顾识别的准确性和识别效率.

(2) 网络协议特征识别方法. 网络协议特征识

别方法是检测数据包头字段的特征，如链路层数据包的 MAC 地址、网络层数据包头的各字段以及传输层数据包头的各字段，方法根据相同类型设备具有相似的数据包头字段，来推测流量的设备类型. Chen 等人[7]使用 TCP 协议头部字段 TTL、IP ID、TCP 窗口大小选项以及时钟频率识别 Android、iOS 和 Windows 操作系统. Nmap[16]是一款支持各种操作系统的开源网络扫描工具，它使用 TCP/IP 协议栈 fingerprinting 进行远程操作系统探测，Nmap 发送一系列 TCP 和 UDP 报文到远程主机，检查响应中的每一个比特. 然后把结果和数据库 nmap-os-fingerprints 中已知的操作系统的 fingerprints 进行比较，如果有匹配，就打印出操作系统的详细信息. 每个 fingerprint 包括一个自由格式的关于 OS 的描述文本和一个分类信息，它提供供应商名称、操作系统版本和设备类型. Nmap 通过实现操作系统流量的分类实现移动流量的识别. 网络协议特征取决于协议字段的差异，需要对各类型设备的流量特征建立特征库，开销较大，对于较为混淆的协议特征无法区别.

(3) 深度数据包检测方法. 深度数据包检测方法(Deep Packet Inspection, DPI)是通过对应用层数据流的报文内容进行特征字符串的匹配. 检测数据包的有效载荷的特征字段以此确定流量类型. 刘翼等人[8]提出了一种采用轻量级流表与深度数据包检测技术相结合的移动流量实时分类方法 UAFs，将网络流量按照时间间隔组成时序流，并提取 HTTP 头部“User-Agent”字段中的简单特征字符串进行移动流量识别. 深度数据包检测方法在会遇到大量的不包含特征字段的无用数据包，覆盖率较低；很多数据包都采用 HTTPS 加密的方式[17]，很多数据包中的特征也无法提取. DPI 技术需要事先提取移动操作系统的模式，利用提取到的模式在未知网络流量中区分不同流量的来源. 由于移动流量的特征可能会随时间推移发生变化，因而 DPI 技术需要定期更新模式，才能在匹配的过程中准确的检测流量类型.

(4) 基于统计和机器学习的分析方法. 基于统计和机器学习的流量分析方法是近年来较为主流的方法. 不同类型设备产生的网络流量具有独特的统计分布特性，在已知网络流量中训练获得分布特性并应用于分析未知网络流量，可以实现流量分类[18]. 机器学习应用于网络流量的分类方式主要分为两种形式：有监督和无监督. 基于统计和机器学习的方法适用于传统和新型网络，可扩展性强，但是分类的速度较慢，耗费资源较多，并且受到样本分布不均等问题的影响.

针对上述分析，本文提出了基于数据流多维统计特征的移动流量识别方法. 该方法从硬件特征、操作系统指纹和用户使用习惯三个方面提取了数据流中具有代表性的特征并对特征进行分析，使用机器学习的方法生成识别模型，实现了移动流量的识别和操作系统的分类.

3　方法设计

3.1　移动流量识别架构

基于数据流多维统计特征的移动流量识别方法架构如图 1 所示.

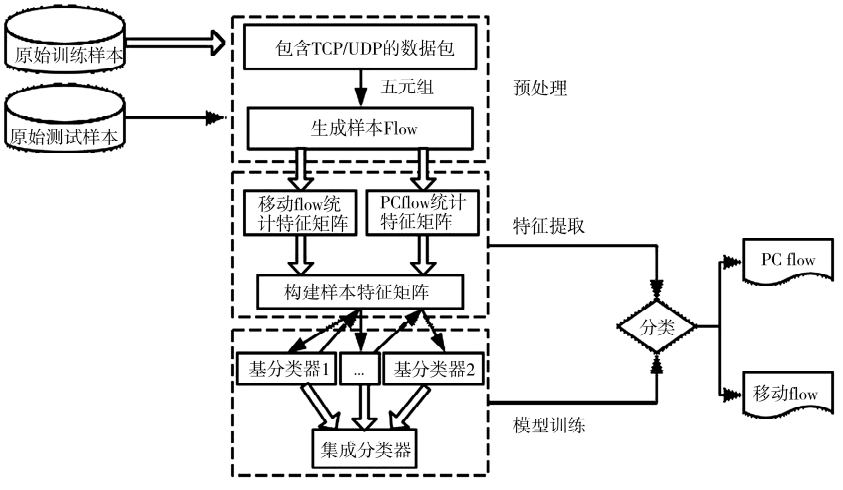

图 1　移动流量识别框架

Fig. 1　Mobile traffic identification framework

该框架主要包括 4 个部分:样本预处理、特征提取、生成流量分类模型以及流量类型预测. 在样本采集和预处理部分,通过网络采集设备将镜像端口的网络流量数据抓取后,进行过滤,去掉 IPv6、局域网广播等干扰数据包以及过滤了非 TCP、UDP 的数据包,选取源 IP 为移动设备的数据包,然后将五元组(Tuple<源 IP,源端口,目标 IP,目标端口,传输层协议>)相同的数据包组成流,完成数据的预处理. 然后从数据流中多个方面提取了具有代表性的特征,构建样本特征矩阵. 借助使用集成方法形成分类模型,达到从传统网络流量中识别出移动流量以及对各类操作系统分类的目的.

3.2 特征提取

对于移动设备,操作系统主要是 Android、iOS,对于非移动设备,操作系统主要是 Windows、Mac OS、Linux. 操作系统对 TCP/IP 的实现,都是严格遵从 RFC 文档的,因为必须遵从相同的协议才能实现网络通信. 但是在具体实现上还是有略微的差别,这些差别是在协议规范之内所允许的,大多数操作系统指纹识别工具都是基于这些细小的差别进行探测分析的.

移动设备和非移动设备在尺寸、耗能、内存等硬件上存在差异,移动设备尺寸和体积都要小于非移动设备;移动设备使用的芯片一般都是 RISE 指令集 CPU,而电脑等非移动设备一般用的都是 CISC 指令集 CPU,RISC CPU 比 CISC CPU 成本低,发热少,因此移动设备耗能可能会少于非移动设备. 人们在使用移动设备和非移动设备的习惯也存在差异,一般使用手机进行简单单一的任务,非移动设备更适用于进行大型文件下载、开发程序、渲染视频等需要具有较高性能的处理器支持的任务. 同时,电脑在多个任务并行的情况下,可以保持各方面的性能,因此,在相同时间内,移动设备的流量消耗可能会低于电脑等非移动设备. 移动设备的键盘输入是虚拟键盘,通过触碰屏幕点击事件,非移动设备的键盘输入是实体键盘,通过鼠标点击事件,对于小的按钮或者需要精细操作的任务,用户更习惯于使用非移动设备. 综合考虑移动设备和非移动设备在尺寸、耗能、内存等硬件上的需求差异、不同操作系统的差异以及用户使用设备习惯的差异,本文总结了以下 10 个具有较高区分度的特征.

(1) 数据流中数据包间的平均间隔. 即一条数据流中的数据包之间的平均间隔时间,如式(1)所示.

$$\bar{T}=\frac{f(T)}{[F]_{pkt}} \tag{1}$$

其中,$[F]_{pkt}$ 为数据包总数;$f(T)$ 为每两个数据包的时间间隔. 不同类型设备会根据无线信道传输协议存在不同的数据包间隔,比如网络抖动会影响数据包间间隔等.

(2) 数据流中数据包间隔标准差. 指一条数据流的数据包包间间隔对于平均大小的分散程度,如式(2)所示.

$$\sigma_T=\frac{\sqrt{\sum t(i)-\bar{T}}}{\bar{T}} \tag{2}$$

根据对大量样本的统计和观察,移动流量的标准差小于非移动流量的标准差,并且移动流量标准差的最大值远小于非移动流量标准差的最大值.

(3) 数据流持续时间. 即流最后一个数据包与初始数据包之间的间隔. 如式(3)所示.

$$f(T)=\sum_{i=2}^{n} P_i(T) \tag{3}$$

其中,$P_i(T)$ 为流中第 i 个数据包的包间隔. 根据对样本的概率统计可以发现,移动设备的数据流持续最长时间远小于非移动设备的数据流持续最长时间. 这与移动设备和非移动设备在内存、CPU 速度方法有明显差异有关.

(4) 数据流中数据包总数. 即一条数据流中的数据包数目. 对采集样本进行随机抽样,统计移动设备和非移动设备的数据包总数,其中移动流的数据包总数为 9 万左右,而非移动流的数据包总数为 239 万左右. 可以发现,不同类型设备数据流的数据包总数存在明显的差异,移动设备和非移动设备的内存大小、耗能,屏幕大小以及用户使用的应用类型都会对该特征产生影响.

(5) 数据流中数据包平均大小. 数据包的平均大小是指一条数据流的数据包平均长度,如式(4)所示.

$$\bar{B}=\frac{\sum P_L(i)}{[F]_{pkt}} \tag{4}$$

移动设备的平均数据包长度的最大值基本上大于非移动设备的平均数据包长度. 这同样与设备的硬件特性和无线信道传输方法有关.

(6) 数据流中数据包大小标准差. 指一条数据流的数据包平均长度的分散程度. 如式(5)所示.

$$\sigma_P=\sqrt{\sum (b_L(i))-\bar{B}}\,/\,\bar{B} \tag{5}$$

根据对移动设备和非移动设备数据包大小标

准差的统计，移动流量的标准差较小，并且标准差的最大值也远小于非移动流量的标准差. 这同样与设备的硬件特性和无线信道传输方法有关.

(7) 数据流生存时间. IP 报头中的数据流生存时间(Time To Live，TTL)值指定数据包可以遍历的最大跃点. 不同操作系统设置不同的初始 TTL 值. 比如，Windows 10 发出的数据包，其 TTL 值是 128，iOS 和安卓默认值为 64，而不同版本的 Linux 操作系统，其数据包的 TTL 值各有所不同. 可以根据这些细微的差别识别远程主机的操作系统，从而更准确地识别操作系统流量.

(8) 数据流标识单调性. 即 IP 报头的标识字段在流中的单调性，主要用于 IP 去碎片化. 根据对不同操作系统数据包的观察，Windows 设备的包中 IP ID 随着时间的推移单调递增，iOS 设备的包中 IP ID 总是随机变化的. 而 Android 设备的 IP ID 则完全随机，其他设备的 IP ID 则单调递增一段时间，并定期重置为随机值.

(9) 数据流类型. 数据流类型包括 TCP、UDP 两种类型. 移动和非移动设备使用的操作系统不同，操作系统在协议栈实现上有差异.

(10) 数据流生成时间. 即形成一条流的时间. 数据流的生成时间与用户的不同设备的使用习惯相关. 移动设备和非移动设备在使用时间分布上存在差异，移动设备的使用时间基本小于非移动设备的使用时间.

在以上的 10 个特征中，特征(1)～(6)根据移动设备和非移动设备在尺寸、耗能、内存等硬件上的需求差异，以及无线信道传输的差异提取特征；特征(7)～(9)根据操作系统指纹提取特征；特征(10)根据用户的使用习惯提取特征. 提取上述 10 个统计特征，构建样本的特征矩阵，作为后续分类模型训练的输入.

3.3　分类模型

该模块使用了机器学习中的集成学习的方法生成分类器. 集成学习是一种强学习器，它通过一定的方式集成一些弱学习器，达到了超过所有弱学习器的准确度的效果，同时它能避免单一模型的过拟合现象，有较好的泛化能力. 本方法以样本特征矩阵作为输入，采用 AdaBoostM1 集成学习算法生成移动流量识别模型，在对流量进行操作系统的细分类上，采用随机森林集成学习算法生成操作系统识别模型.

3.3.1　*移动流量识别模型*　从训练样本中提取 3.2 节提到的 10 个特征形成特征矩阵，使用 AdaBoostM1 集成学习算法进行训练，形成移动流量识别模型，如图 2 所示.

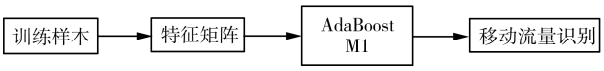

图 2　移动流量识别模型
Fig. 2　Mobile traffic identification model

AdaBoostM1 使用 boosting 方法[19]，本模型使用的 AdaBoostM1 的函数表示形式如式(6)所示.

$$\text{classifier}^{[trer,tser,w]}=\text{adaboostM1}(type,tr,tr_{\text{fea}},ts,ts_{\text{fea}},M,Y) \tag{6}$$

其中，$trer$ 为训练错误率；$tser$ 为测试错误率；w 为弱分类器的权重. $type$ 为弱分类器的类型；tr、tr_{fea} 为训练样本和训练样本的特征，ts、ts_{fea} 为测试样本和测试样本的特征；M 为轮训次数；Y 为样本类别数. 为每个用来训练的数据流集合赋予一个权重，权重的大小代表了该数据流被下一个弱分类器列入训练样本集的概率. 如果某个流样本能被当前弱分类器准确分类，那么在构造下一个弱分类器的流训练样本时，该样本被选中的概率就降低；反之，它被选中的权重就相应提高. AdaBoostM1 算法只要有足够的数据以及弱分类器就能够达到任意预测精度，因此非常实用于移动流量识别研究.

3.3.2　*操作系统流量分类模型*　可以通过操作系统流量的分类实现移动流量的识别，将分类为 Android、iOS 的流量识别为移动流量，将分类为 Windows、Linux 和 Mac OS 的流量识别为非移动流量. 操作系统多分类模型如图 3 所示. 从训练样本中提取 3.2 节提到的 10 个特征形成特征矩阵，使用随机森林算法进行训练，形成操作系统流量分类模型.

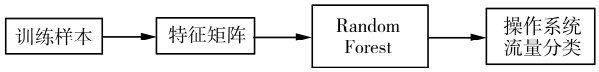

图 3　操作系统流量分类模型
Fig. 3　Operating system traffic classification model

随机森林使用 bagging 方案，随机森林由一定数量的决策树组成，森林中的每棵决策树根据数据流样本的属性得出一个分类结果，然后把这些分类结果以投票的形式保存下来，随机森林选出票数最高的分类结果令其成为这个森林的分类结果. 随机

森林能够大幅降低偏差和方差,提高机器学习的稳定性和准确性,同时避免了过度拟合. 对于操作系统的分类研究使用的流量数据均为在实验室采集的流量,样本数量和样本分布具有一定的局限性,随机森林算法在运算量没有显著提高的前提下提高了预测精度,可以很好的弥补数据集分布不均衡的问题.

4 实验评估

4.1 实验数据

利用实验室搭建的无线网络环境,如图 4 所示. 使用网络封包分析软件 Wireshark 采集了为期一周的流量数据,首先对数据进行过滤,去掉 IPv6、局域网广播等干扰数据包以及过滤了非 TCP、UDP 的数据包. 然后利用数据包的物理地址和接入网络的终端设备的对应关系,统计出具体设备类型. 最后选择源 IP 为已知设备的流量数据. 随后以数据流作为后续分析处理的对象. 为了使数据集尽可能的分布均匀,本实验一共接入网络终端设备 23 台,其中移动设备 11 台,传统台式设备 12 台,共分配 IP 数量 21 个. 终端设备操作系统包括 Android、iOS、Mac OS、Linux、Ubuntu、Windows 7、Windows 8、Windows 10、Windows XP. 表 1 是进行移动流量识别使用的流数据集,表 2 是进行操作系统识别使用的流数据集. 其中 D_1、D_3 为训练集,D_2、D_4 为测试集.

4.2 实验结果分析

流量类型识别和操作系统分类的所有检测结果如表 3 和表 4 所示.

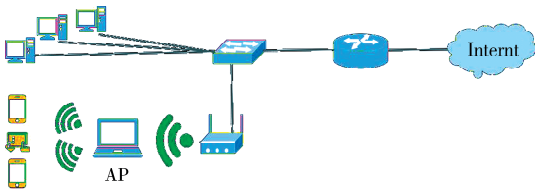

图 4 数据集采集环境

Fig. 4 Data collection environment

表 1 移动流量识别数据集

Tab. 1 Mobile traffic recognition data set

流数据集	设备总数	移动流量数/个	非移动流量数/个	数据流总数/条
D_1	18	5 825	5 770	11 595
D_2	8	2 594	2 563	5 157

表 2 操作系统分类数据集

Tab. 2 Operating system classification data set

流数据集	设备总数	Android 流量数/个	iOS 流量数/个	Mac OS 流量数/个	Windows 流量数/个	Linux 流量数/个	数据流总数/条
D_3	18	3 166	2 954	3 053	3 333	2 500	15 006
D_4	15	1 300	1 118	1 100	994	1 283	5 795

表 3 流量类型识别结果

Tab. 3 Traffic type recognition results

训练集	验证方法	分类算法	检测率/%	误报率/%	准确率/%	AUC/%
D_1	交叉验证(十倍)	Random Forest	100	0.017 3	99.991 4	100
		AdboostM1	99.965 6	0.017 3	99.974 1	100
D_1	D_2独立测试	Random Forest	100	0.010 0	66.209 3	100
		AdboostM1	92.980 0	7.500 0	92.400 0	97.8

表 4 操作系统分类结果

Tab. 4 Operating system classification results

训练集	验证方法	分类算法	检测率/%	误报率/%	准确率/%	AUC/%
D_3	交叉验证(十倍)	Random Forest(iOS)	99.12	0.24	99.56	100
		AdboostM1(Windows)	84.02	6.05	80.07	87.9
D_3	D_4独立测试	Random Forest(iOS)	74.25	3.58	100	92.0
		AdboostM1(Windows)	85.33	7.58	78.00	86.9

在移动流量的识别结果中，使用十倍交叉验证随机森林和 AdaBoostM1 的检测准确率都很高，准确率达到 99%以上. 使用 D_2 样本数据集独立测试模型的泛化能力，随机森林检测准确率基本没有变化，AdboostM1 的结果为 92.98%左右.

在操作系统的分类结果中，使用十倍交叉验证随机森林的结果很好，准确率达到了 99%左右. 使用十倍交叉验证 AdaboostM1 的结果略低，准确率只有 80%，其中 AdaboostM1 对于 Windows、Android 的准确率略高，但是 iOS 和 mac OS 的分类效果比较差，导致整体的准确率降低. 使用 D_4 样本数据集独立测试模型的泛化能力，随机森林的检测率结果降低了 25%准确率结果没有变化，AdaboostM1 的准确率基本没有变化.

由此可以证明，本文方法可以准确识别移动流量，并且能够较为准确的识别目前主流使用的 5 种操作系统类型. 对于所采用的 10 个特征，本文采用 Best first 和 CFS Subset Evaluator 方法，对特征进行筛选，得到具有较高区分度的特征. 其中数据流生成时间、数据流标识单调性和数据流平均生存 TTL 这三个特征的识别和分类效果最好.

4.3　实验对比

本文进行了三组对比实验，第一组实验使用加拿大网络安全研究所的入侵检测评估数据 CIC-IDS2017[11]，该数据集中包含 Windows、Mac OS 和 Linux 三类非移动设备的良性样本. 本实验目的是验证本文提出的操作系统流量分类模型的鲁棒性. 使用本文方法进行分类识别，可以准确的识别 Linux、Windows 和 Mac OS，准确率如表 5 所示.

表 5　本文方法识别未知数据集操作系统

Tab. 5　Accuracy of unknown data set operating system by method in this paper

实验	Windows	Mac OS	Linux
准确率/%	100	94.1	97.2

第二组实验对比了基于数据流和基于数据包的移动流量识别方法. 在基于数据包的移动流量识别方法中，提取了包括 TCP 数据包、IP 数据包中的部分字段，一共 12 个特征. 相同数据量的样本的检测时间如表 6 所示. 虽然基于数据包的移动流量的识别准确能够达到 99%，但是基于数据流多维特征的移动流量识别模型生成时间远少于基于数据包的识别方法，同时十倍交叉验证的时间更是远少于基于数据包的识别方法. 本文提出的方法在识别效率上的优势非常明显.

表 6　基于数据流的方法与基于数据包的方法效率对比

Tab. 6　Efficiency comparison between flow-based method and packet-based method

方法	基于数据流多维特征的移动流量识别	基于数据包的移动流量识别
移动流量识别模型使用时长/s	0.16	8.70
操作系统流量分类模型生成时长/s	0.49	181.66

第三组实验对比了基于数据流的多维特征和基于数据流传统特征的识别方法. Chen 等人[7]使用了 IP Time to live、IP ID 单调性、TCP 时间戳选项、TCP 窗口大小选项和时钟频率 5 个特征，对移动设备的 iOS、Android 和 Windows 三种操作系统进行识别. 上述的 5 个特征属于被广泛使用的流量特征，本实验使用这些特征，在相同的数据集上进行移动流量的识别，识别准确率如表 7 所示. 在移动流量的识别上，使用本文的多维特征比传统特征的准确率更高.

表 7　基于数据流传统特征的方法与本文方法结果对比

Tab. 7　Accuracy comparisons between traditional flow feature-based method and the method in this paper

方法	基于数据流传统特征的方法	本文方法
准确率/%	80.00	99.97

5　结　论

本文以“流”为分析粒度，设计了一种利用网络流量中的统计特征来识别移动流量以及设备的操作系统的方法. 本方法从硬件平台、操作系统指纹及用户使用习惯三个不同角度提取了移动设备和非移动设备间具有差异性的 10 个特征，构建了基于流的样本特征库；以此特征库结合 AdboostM1 算法和随机森林算法训练分类器模型，所得模型在完全无交叉的两个数据集上取得较好的检测准确率和泛化性能. 本文评估了一些特征的有效性，通过分析样本发现 TTL、IP ID 的单调性在识别移动流量时有较强的区分度. 本文同时结合试验结果分析了两种机器学习分类方法对于样本数据集以及分类结果有影响的具体表现原因. 实验结果表明，本方法所选取的特征能够较好的识别移动流量以

及具体操作系统,识别准确率达到 99%以上. 与 P0F 方法比较,本文提出的方法的识别准确率较高. 该方法的特征具有多维性,方法同样适用于加密流量的分类,且分类准确率达到 99%,高于现有的其他方法. 移动流量的识别可以为进一步恶意移动流量、移动流量管理等研究奠定基础.

参考文献:

[1] Stat C. Mobile and tablet now gets more usage than desktop[EB/OL]. (2016-11-01). [2018-12-25]. https://blog. statcounter. com/2016/11/mobile-and-tablet-now-gets-more-usage-than-desktop/.

[2] CISCO 中国. 皆字节时代:趋势与分析[R/OL]. (2016-03-12). [2018-12-26]. https://www. cisco. com/c/m/zh _ cn/express/case _ center/vertical/wprdc001405. html.

[3] 彭大芹, 罗裕枫, 江德潮, 等. 基于移动信令数据的城市热点识别方法[J]. 重庆邮电大学学报:自然科学版, 2019, 31: 99.

[4] Zimperium. 2018 H1 Report: global threat report [R/OL]. (2018-05-18). [2018-12-28]. https://get. zimperium. com/threat-report-1h-2018/.

[5] Aceto G, Ciuonzo D, Montieri A, *et al*. Traffic classification of mobile apps through multi-classification[C]// Proceedings of the Globecom IEEE Global Communications Conference. Singapore: IEEE, 2018.

[6] 董刚, 余伟, 玄光哲. 高级持续性威胁中攻击特征的分析与检测[J]. 吉林大学学报:理学版, 2019, 57: 155.

[7] Chen Y C, Liao Y, Baldi M, *et al*. OS fingerprinting and tethering detection in mobile networks[C]//Proceedings of the 2014 Conference on Internet Measurement Conference. Vancouver, Canada: ACM, 2014.

[8] 刘翼, 嵩天, 廖乐健. 基于时序流的移动流量实时分类方法[J]. 北京理工大学学报, 2018, 38: 101.

[9] 邹腾宽, 汪钰颖, 吴承荣. 网络背景流量的分类与识别研究综述[J]. 计算机应用, 2019, 39: 802.

[10] Zalewski M. p0f v3 (version 3. 09b)[EB/OL]. (2014-08-16). [2018-12-29]. http://lcamtuf. coredump. cx/p0f3/

[11] Sharafaldin I, Lashkari A H, Ghorbani A A. Toward generating a new intrusion detection dataset and intrusion traffic characterization [C]//Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP). Funchal, Madeira, Portugal: IEEE 2018.

[12] Barnes J, Crowley P. k-p0f: A high-throughput kernel passive OS fingerprinter [C]// Proceedings of the Architectures for Networking and Communications Systems. [s. l.]: IEEE, 2013.

[13] Ranjan G, Tongaonkar A, Torres R. Approximate matching of persistent LExicon using search-engines for classifying Mobile app traffic [C]// Proceedings of the IEEE INFOCOM 2016-IEEE Conference on Computer Communications. San Francisco, CA: IEEE, 2016.

[14] 徐明, 杨雪, 章坚武. 移动设备网络流量分析技术综述[J]. 电信科学, 2018, 34: 98.

[15] 胡婷, 王勇, 陶晓玲. 混合模式的网络流量分类方法[J]. 计算机应用, 2010, 30: 2653.

[16] Shafiq M, Yu X, Bashir A K, *et al*. A machine learning approach for feature selection traffic classification using security analysis [J]. J Supercomput, 2018, 2018: 1.

[17] 贾军, 杨进, 李涛. 一种基于 DPI 自关联数据包检测分类方法 [J]. 四川大学学报: 自然科学版, 2019, 56: 29.

[18] Nguyen T, Armitage G. A survey of techniques for internet traffic classification using machine learning [J]. IEEE Commun Surv, 2009, 10: 56.

[19] 王海, 李诚, 蔡英凤, 等. 基于 DSP 平台的实时视觉车辆检测方法[J]. 江苏大学学报:自然科学版, 2019, 40: 6.

引用本文格式:

中 文: 武思齐, 王俊峰. 基于数据流多维特征的移动流量识别方法研究[J]. 四川大学学报: 自然科学版, 2020, 57: 247.

英 文: Wu S Q, Wang J F. Research on mobile traffic identification based on multidimensional characteristics of data flow [J]. J Sichuan Univ: Nat Sci Ed, 2020, 57: 247.