计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330055
61(9): 2321−2333, 2024

面向加密流量的社交软件用户行为识别

吴 桦[1,2,3] 王 磊[1] 黄瑞琪[1] 程 光[1,2,4] 胡晓艳[1,2,3]
[1](东南大学网络空间安全学院 南京 211189)
[2](网络通信与安全紫金山实验室 南京 211111)
[3](网络空间国际治理研究基地(东南大学) 南京 211189)
[4](江苏省泛在网络安全工程研究中心(东南大学) 南京 211189)
(hwu@seu.edu.cn)

Social Software User Behavior Identification for Encrypted Traffic

Wu Hua[1,2,3], Wang Lei[1], Huang Ruiqi[1], Cheng Guang[1,2,4], and Hu Xiaoyan[1,2,3]
[1](*School of Cyber Science and Engineering*, *Southeast University*, *Nanjing* 211189)
[2](*Purple Mountain Laboratories for Network and Communication Security*, *Nanjing* 211111)
[3](*International Governace Researach Base of Cyberspace* (*Southeast University*), *Nanjing* 211189)
[4](*Jiangsu Provincial Engineering Research Center of Security for Ubiquitous Network* (*Southeast University*), *Nanjing* 211189)

Abstract As smart terminals and social networks are increasingly integrated into people's daily life, user behavior identification for social software plays an increasingly important role in network management, network environment supervision, and market research. Social software commonly uses end-to-end encryption protocols for encrypted data transmission, and existing methods usually extract statistical features of the encrypted data for behavior identification. However, these methods have unstable identification performance and require a large amount of data, and these drawbacks affect the practicality of these methods. We propose a social software user behavior identification method for encrypted traffic. First, stable control flow data are identified from the encrypted traffic, and the control service packet payload length sequence is extracted. Then, two neural network models are then designed to automatically extract features from control flow payload length sequences to identify user behavior at a fine granularity. Finally, experiments are conducted with WhatsApp as an example, and the precision, recall, and *F*1-*score* of the two neural network models for recognizing WhatsApp user behavior are over 96%. The experimental comparison with similar work proves the stability of the identification performance of the method. In addition, the method can achieve high identification precision with a few control packets, which is of great relevance to the study of real-time behavior identification.

Key words social network; user behavior; service frequency; control flow; length sequence

摘 要 随着智能终端和社交网络越来越融入人们的日常生活,针对社交软件的用户行为识别在网络管理、网络环境监管和市场调研等方面发挥越来越重要的作用.社交软件普遍使用端到端加密协议进行加密数据传输,现有方法通常提取加密数据的统计特征进行行为识别.但这些方法识别的性能不稳定且需要的数据量多,这些缺点影响了方法的实用性.提出了一种面向加密流量的社交软件用户行为识别方法.首先,从加密流量中识别出稳定的控制流数据,并提取控制服务数据分组负载长度序列.然后设计了2种神经网络模型,用于自动从控制流负载长度序列中提取特征,细粒度地识别用户行为.最后,以WhatsApp

收稿日期: 2023-01-30; **修回日期**: 2023-11-01
基金项目: 国家重点研发计划项目(2021YFB3101403)
This work was supported by the National Key Research and Development Program of China (2021YFB3101403).

为例进行了实验，2种神经网络模型对WhatsApp用户行为的识别精准率、召回率和*F*1-*score*均超过96%. 与类似工作的实验比较证明了该方法识别性能的稳定性，此外，该方法能够通过很少的控制流数据分组达到较高的识别精准率，对实时行为识别的研究具有重要的现实意义.

关键词 社交网络；用户行为；服务频次；控制流；长度序列

中图法分类号 TP399

随着移动互联网的快速发展，智能终端的移动应用已经成为人们社会生活的一部分. 移动应用的用户行为通过加密流量在互联网中传输，网络管理部门可以通过流量分析，在不影响应用正常使用的情况下获取网络空间安全管理所需的信息，以应对潜在的有害用户行为.

在所有的移动应用中，社交软件以其便捷的功能改变了人们的生活方式，并且拥有着庞大的用户群体，因此其针对社交网络的用户行为识别具有重要的意义. 一方面，识别出用户行为可用于市场调研、广告精准推送，也可作为运营商评估服务质量的依据. 另一方面，也存在利用社交软件传播恶意消息扰乱社会治安的现象，对这些行为进行识别有助于对网络空间进行监管. 由于社交软件普遍使用加密传输，无法通过解析分组信息得到有效信息. 近年来面向加密流量的社交软件用户行为识别技术成为一个研究热点.

从社交软件用户行为产生的用户流量中识别出用户行为，面临的主要难题是由于信息被加密，无法通过分组的载荷信息分析出具体的用户行为. 最初，研究者们通过检测端口号[1-2]或分析分组载荷[3-4]来分析流量. 然而，随着加密流量的增加，许多应用程序都开始通过TCP的443端口进行传输，这导致无法仅通过端口号区分具体的应用程序. 此外，由于分组载荷被加密，也无法对分组载荷进行有效的检测. 目前，识别加密流量的方法主要是通过提取流量特征并使用机器学习[5-7]或者深度学习[8-10]的方法对加密流量进行识别. 虽然通过加密可以保护加密数据的有效载荷，但加密流量的传输特征(如数据分组的长度、间隔、数量、网络请求及应答的时间信息和数据方向等)可用来对加密流量进行识别，这种流量分析方法方法被称为侧信道攻击[11]. 近年来随着内容分发网络(content delivery network, CDN)、负载均衡等技术的广泛使用，域名与IP地址的映射关系变得更加复杂，使得网络流量的复杂程度进一步提升，这样导致从流量中提取的一些特征对分类的贡献降低.

此外，这些研究方法主要依赖于加密流量的统计特征，而这些特征容易受到传输路径的影响. 例如，数据分组长度会受到路径的最大传输单元(maximum transmission unit, MTU)的影响，数据分组之间的间隔会受到网络传输信道带宽的影响. 这导致使用这些特征训练的模型在不同的网络环境下的识别效果差异很大、泛化能力不足. 相同的用户行为在不同的网络环境下产生的流量具有不同的特征，这影响了现有研究方法结果的稳定性.

本文提出了基于控制流的用户行为识别方法，从社交软件产生的加密流量中识别控制信息，然后使用深度学习的方法自动从控制信息中提取特征进行行为识别. 控制信息是一种特殊的流量数据，实现移动应用时，为了支撑多样化的功能，移动应用需要使用不同类型的服务器和标准Web服务接口API(application programming interface)，需要对不同类型的数据传输进行协调和控制，本文将此类信息称为控制信息，这些控制信息是功能实现的基础，但并非用户传递的信息，具有相对的稳定性. 相比现有研究方法提取的流量特征，从控制信息提取的特征具有更好的稳定性，因此分类结果也具有较好的稳定性.

据GlobalWebIndex关于社交媒体最新趋势报告，WhatsApp是目前除中国大陆外使用最为广泛的即时通信应用之一，本文将WhatsApp作为研究对象，提取WhatsApp用户行为产生加密流量中的控制流进行研究，实验证明该方法不仅可以对WhatsApp细粒度的用户行为进行识别，还具有很高的识别准确度和稳定性.

本文的贡献主要包括2点：

1)基于社交软件产生的加密流量提出了控制流的概念，并以WhatsApp为例提出了一种控制流提取的方法；

2)设计了LS-CNN和LS-LSTM两种神经网络模型，以控制流负载长度序列作为输入对用户行为进行分类，通过实验表明，2种神经网络模型的识别精准率、召回率和*F*1-*score*均超过96%.

1 相关工作

面向加密流量的用户行为分析关键在于加密流

量分析方法. 近几年, 许多研究机构在加密流量分析领域进行了广泛研究, 并取得了一系列的成果. 这些研究主要关注如何从加密流量中提取能够用于分类的特征, 并利用人工智能方法对流量进行分类. 机器学习方法需要进行人工分析和特征提取, 而深度学习方法则能够自动提取特征.

许多传统的机器学习算法首先被应用于网络流量分类, 如 K-均值算法[12]、朴素贝叶斯算法[13]、支持向量机[14]、随机森林[15] 等. 机器学习算法通常需要进行特征工程, 要求研究者具备丰富的相关知识, 对流量进行深入分析, 提取具有区分度的多维特征. 一些研究者采用了深度学习的方法自动提取流量特征. 例如: 文献 [9] 将深度循环神经网络应用于 18 个应用程序的分类中, 从原始流量中学习代表性特征, 取得了良好的分类效果; 文献 [10] 提出了一种基于深度学习的方法, 将特征提取和分类阶段集成到 1 个系统中对网络流量进行分类.

社交网络的快速发展导致对用户行为识别的需求不断提升, 而通过加密流量识别用户行为是一种快速、便捷的方法. 面向加密流量的识别主要包括移动应用的识别[16-20] 和用户行为的识别[21-27].

一些研究提出了通用的识别方法, 利用流量的传输速率等特征进行识别. 然而, 由于这些特征会随着网络服务质量的变化而变化, 导致识别结果会受到网络环境波动的影响. 文献 [17] 通过使用传输速率等特征, 对 35 个应用程序进行识别, 以揭示服务器的行为模式, 并进一步识别用户的细粒度行为. 由于不同的应用数据采集时段不一样, 且经过的路径也不一致, 该方法在识别不同的移动应用和应用内行为时识别准确率和召回率波动较大.

由于单个特征的使用会导致模型的泛化能力不足, 因此有不少研究提取了多个维度的特征进行识别. 文献 [22] 对微信和 WhatsApp 的加密流量进行了研究, 通过将用户的行为模式、加密流量特征和时间序列关系进行联合建模, 来识别服务的使用情况. 文献 [25] 对微信使用的 MMTLS 协议进行了分析并比较了微信中 7 种典型活动的加密流量, 提取了上下行数据量、数据分组长度分布、流量持续时间等作为特征来进行用户行为识别. 然而, 当网络性能下降导致丢包和重传时, 这些特征都会发生变化, 因此该识别方法的泛化能力也受到影响.

为了解决流量统计特征在不同传输环境下导致模型泛化能力不足的问题, 文献 [26] 提出了一种方法, 该方法提取了不受环境影响的应用层数据长度特征, 用于对 Instagram 的 9 种用户行为进行分类, 并取得了良好的分类性能. 然而, 该方法需要从网络层数据中恢复应用层数据单元的特征, 而在网络环境较差的情况下, 需要大量的缓存空间和处理能力, 才能准确地恢复应用层特征, 因此该方法需要大量的资源才能进行特征提取.

随着深度学习的发展, 特征提取的过程可以自动进行, 从而简化了特征提取的步骤. 文献 [27] 提出了一种分 2 步进行的应用分类方法: 首先设计了一个基础分类器, 该分类器能够自动提取流量特征并进行初步预测, 其中包括了多个时间尺度的特征. 然后通过应用注意力机制对初步预测进行聚合, 从而得到最终的分类结果. 尽管该方法的分类效果尚不理想, 但使用深度学习来自动提取特征已经成为加密流量分析领域的一个研究趋势.

但是在实际应用中, 流量特征和行为之间的关系并不稳定. 由于缓存使用和应用程序设计的差别, 相同用户行为产生的加密流量有可能差异很大, 例如短时间内传输同样的文件时, 最开始会传输整个文件, 但随后只会传输很少的数据用于从缓存中调用已有的数据. 又例如当用户收到图片或视频消息提醒时, 需要点击下载才能完全传输数据, 而不是直接完成传输. 由于上述情况的存在, 对所有流量信息进行同等的分析很难稳定地区分出用户行为.

针对上述问题, 本文提出了一种新的方法. 该方法可以从加密流量中提取控制信息, 并使用深度学习的方法自动提取控制信息中的特征, 以适应多变的用户行为. 与同类研究相比, 该方法可以更准确、更细粒度地识别社交软件的用户行为.

2 方　　法

本文提出的模型如图 1 所示, 主要由 3 个部分组成: 控制流的识别、控制流的特征提取和分类模型的训练.

2.1 控制流的定义

在说明控制流之前, 首先介绍本文分析的数据流和服务流的概念.

定义 1. 数据流. 具有相同的五元组(源 IP 地址, 源端口号, 传输协议, 目的 IP 地址, 目的端口号)的分组序列为数据流. 用户向服务器发送的数据流称为上行流, 而服务器向用户发送的数据流则称为下行流.

定义 2. 服务流. 具有相同的三元组(IP 地址, 端口号, 传输协议)的分组序列为服务流. 由于负载均

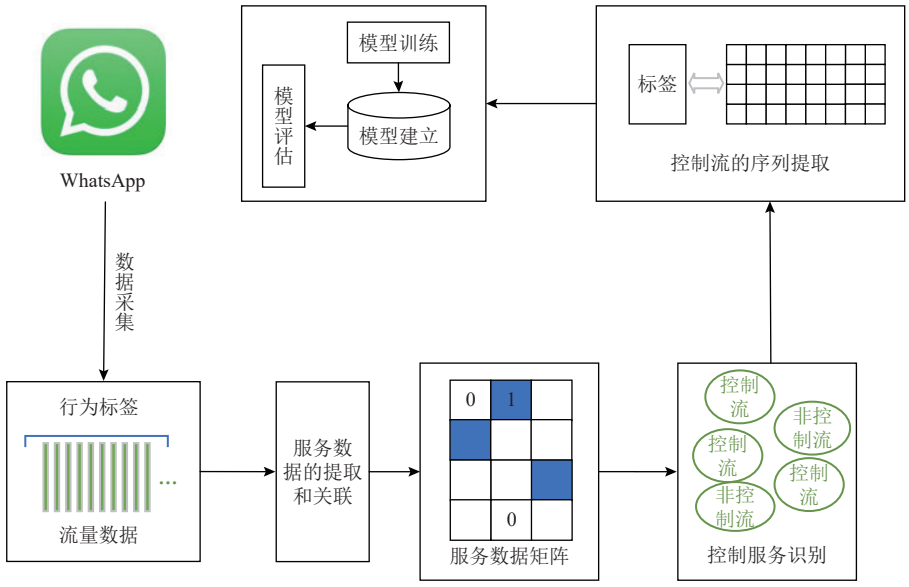


Fig. 1 Construction process of user behavior identification model based on control flow

图 1 基于控制流的用户行为识别模型的构建流程

衡的使用, 同一个服务可以由多个 IP 地址提供, 但它们使用的是同一个域名. 在本文中, 我们将使用同一个域名的多个三元组流称为关联服务.

控制数据流是用于对服务进行控制协调的五元组数据流. 由于移动应用采用了大型软件模块化设计, 使用来自不同开发者的标准服务接口, 用户行为所触发的流量不仅包括与内容相关的流量, 还包括调用这些标准服务接口的控制信息. 将这些控制信息按照一定的规则组合起来形成的数据流被称为控制流.

触发控制流的是一些公用的服务, 这些服务的流量特征相对稳定, 可用于训练分类模型. 因此, 本文将对这些具有稳定性的控制流进行研究.

2.2 基于服务频次识别控制流

根据控制流的定义可知每个用户行为都会触发控制流, 在数据集中会存在大量的控制流数据, 基于此本文引入了服务频次的概念, 并设计了基于服务频次的控制流识别方法. 相对于数据流, 服务流的特点是会多次重复出现. 本文方法借鉴了文本处理中词频的概念, 认为出现频率较高的服务在分类中具有较高的重要性. 因此, 我们对每个用户行为中出现的服务进行统计, 计算其频次, 并根据频次来确定控制流.

2.2.1 服务频次

假定数据集中有 n 个用户行为的流量样本, 每个样本中完成 1 种用户行为, 包含了 u 种不同的用户行为. 数据集中共产生了 m 种不同的服务 $s_1, s_2, \cdots, s_m$, 第 p 个样本在用户行为时间内产生的服务可以使用向量 $\boldsymbol{S}_p = \left(S_1^p, S_2^p, \cdots, S_m^p\right)$ 表示, 向量 $\boldsymbol{S}_p$ 是按服务是否触发对流量样本 p 编码形成的独热编码, 其中 S_m^p=0 代表样本 p 在特定时间内没有出现服务 s_m, S_m^p=1 代表样本 p 在特定时间内出现了服务 s_m. 样本的服务向量 $\boldsymbol{S}_p$ 组成的服务矩阵如式(1)所示. 式(2)定义了该数据集中服务 s_m 的服务频次, 其中分子项表示该服务在所有样本中出现的次数, 分母项表示在所有样本中出现的全部服务的次数.

$$\boldsymbol{S} = \begin{pmatrix} S_1^1 & S_2^1 & \cdots & S_m^1 \\ S_1^2 & S_2^2 & \cdots & S_m^2 \\ \vdots & \vdots & & \vdots \\ S_1^n & S_2^n & \cdots & S_m^n \end{pmatrix}, \tag{1}$$

$$f_{S_m} = \frac{\sum_{i=1}^{n} S_m^i}{\sum_{i=1}^{n}\sum_{j=1}^{m} S_j^i}. \tag{2}$$

在计算上述的服务频次时, 我们并不需要完整地采集用户从开始使用到结束的所有数据, 而只需要采集足够的样本数据即可. 此外, 对于一些特殊的数据分组, 例如载荷为 0 的分组或者流的控制信息分组, 我们会直接进行过滤, 不加入统计计算.

2.2.2 关联同一服务

由于负载均衡的使用, 同一服务可能会有多个不同的三元组流标识. 关联同一服务是将这些三元组流关联起来, 在分析过程中统一计算它们的频次, 从而得到同一服务的总频次.

为了关联同一个服务, 需要建立同一个服务的

域名和多个 IP 地址之间的映射关系. 尽管可以通过 DNS 查询获取域名与 IP 地址之间的映射关系，但是由于存在 DNS 的负载均衡机制，即时的 DNS 查询只能获得部分 IP 地址. 为了解决这个问题，本文基于多次 DNS 查询构建了服务的域名和 IP 地址的对应关系库. 表 1 给出了本地域名和 IP 地址关系库中 WhatsApp 相关的记录，第 1, 2 行记录表明每个服务对应 1 个域名和多个别名，以及多个 IP 地址，其中别名的 3 级域名标识了服务器的功能. 第 3 行记录表明 WhatsApp 调用了谷歌公司的位置服务.

Table 1 WhatsApp Related Major Domain Names Mapping

表 1 WhatsApp 相关的主要域名映射

序号	域名	别名	IP 地址
1	mmx-ds.cdn.whatsapp.net	graph.whatsapp.net static.whatsapp.net mmg.whatsapp.net dit.whatsapp.net scontent.whatsapp.net www.whatsapp.com	157.240.22.53 31.13.70.49 31.13.64.51 157.240.210.60
2	chat.cdn.whatsapp.net	g.whatsapp.net	157.240.206.61 157.240.27.55 157.240.21.53 31.13.64.53
3	maps.googleapis.com		172.217.164.106 172.217.6.42 216.58.200.234

关联同一服务后，可以使用三元组(域名，端口，传输协议)来统一表示一个服务. 需要注意的是，WhatsApp 应用时使用的服务较多，并且为了提高用户的服务感受，大型 APP 服务商都使用了 CDN 技术和任播(anycast)技术进行数据加速，这会导致不同地区用户使用的服务器也会有区别. 因此无法在本文中罗列全部的域名映射，表 1 是基于本文的数据集给出的主要域名映射.

2.2.3 基于服务频次和传输特征识别控制流

尽管我们已经关联了同一个服务，但在加密传输的情况下，无法确定它是控制流还是数据流. 因此，在本节中，我们将基于服务频次和数据传输特征来识别控制流.

为了计算服务频次，我们需要基于已有的训练数据构建服务矩阵，然后根据式(2)进行计算. 图 2 显示了对 WhatsApp 数据集进行计算得到的服务频次. 关联后的服务使用传输协议、域名和端口来表示，例如 T_www.whatsapp.com:443 表示该服务使用了 TCP 协议的 443 端口，域名为 www.whatsapp.com.

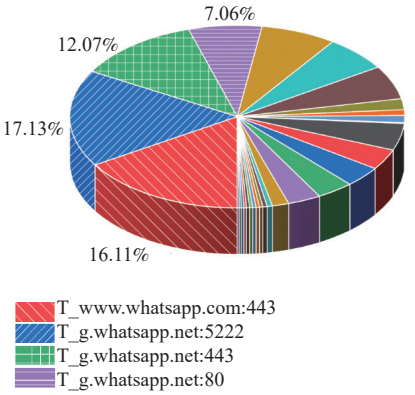


Fig. 2 Composition diagram of service frequency after association in WhatsApp traffic data

图 2 WhatsApp 流量数据中关联后服务频次构成图

图 2 展示了使用 WhatsApp 过程中各个服务频次的占比. 其中，服务频次最高的 4 个服务由 g.whatsapp.net 和 www.whatsapp.com 这 2 个服务器产生. 具体地，g.whatsapp.net 通过 TCP 协议的 80, 443, 5222 端口提供了 3 种服务，而 www.whatsapp.com 只通过 443 端口提供 1 种服务. 尽管这 2 个服务器都提供了通过 443 端口的 TCP 服务，但是 g.whatsapp.net 的 443 端口连接并没有进行 TLS 握手过程，实际上并没有使用标准的 TLS 协议，而 www.whatsapp.com 的 443 端口连接遵循了标准的 TLS 协议. 这说明 g.whatsapp.net 提供的服务可能不是面向常规的客户端，而更可能是应用内部的控制服务.

此外，相对于数据服务而言，控制服务的数据量较小，但需要在应用过程中始终保持连接. 基于这一特点，我们对 WhatsApp 的流量数据进行分析. 图 3 展示了频次最高的 2 个服务器数据传输的速率. 尽管这 2 个服务器的传输时间相近，但是 g.whatsapp.net 的传输数据速率明显较低.

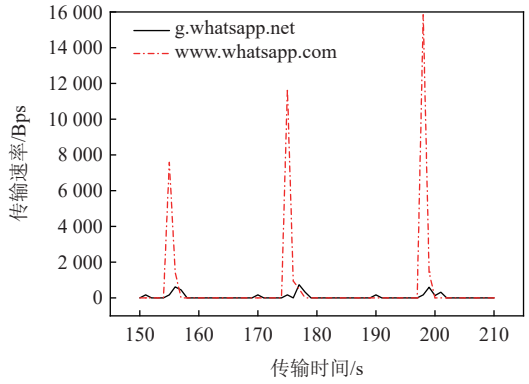


Fig. 3 Comparison of data transmission rates between two servers

图 3 2 个服务器传输数据速率的比较

综合控制流的传输特征和关联服务的服务频次，以及域名 g.whatsapp.net 对应服务器的数据流特征，本文将 g.whatsapp.net 对应服务器产生的数据流识别为控制流.

2.3 控制流分析

不同的用户行为所需要的控制服务组合是不一样的，如果能识别出数据中的控制服务并据此分类，可以有效地对用户行为分类. 但是服务数据被加密传输后，无法通过分析分组中的负载内容识别服务类型，本文使用分组的长度作为识别依据.

为进一步分析控制流的特征，提取了控制流的数据分组负载长度组成序列并进行分析. 不同用户行为下的数据分组负载长度序列如图 4、图 5 所示，展示了 8 种不同用户行为所对应的控制流数据分组负载长度的变化. 在图例中，使用“a_”代表该行为包含主动和被动 2 种状态，而其余 2 种行为只有主动状态. 图 4 展示了不同用户行为触发的上行控制流负载长度序列具有明显的差异，而图 5 显示了不同用户行为触发的下行控制流负载长度序列具有一定的相似性.

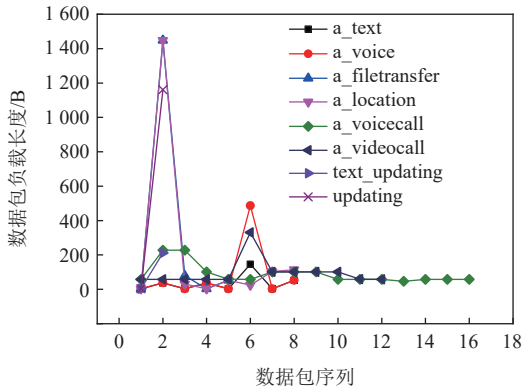


Fig. 4 The payload length sequence of upward control flow triggered by different user behaviors

图 4 不同用户行为触发的上行控制流负载长度序列

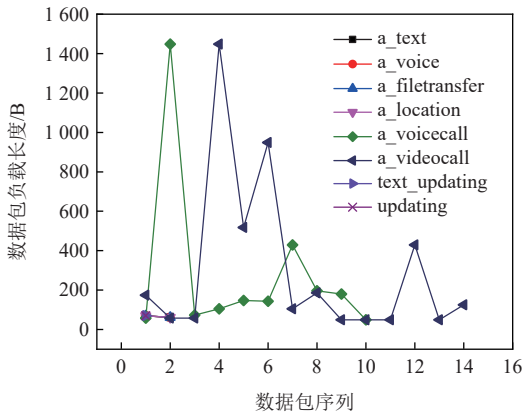


Fig. 5 The payload length sequence of downward control flow triggered by different user behaviors

图 5 不同用户行为触发的下行控制流负载长度序列

本文使用余弦相似度度量上下行数据流中负载长度序列的相似性. 为了区分上行数据和下行数据，本文采用负值表示上行数据分组负载长度，而使用正值表示下行数据分组负载长度，完整的控制流负载序列由正值和负值的负载长度序列组成. 计算余弦相似度时将这 2 个序列看作向量，使用 2 个向量间夹角的余弦值来表征 2 个序列的相似性，这可以由式(3)计算：

$$cos_sim(\boldsymbol{X},\boldsymbol{Y})=\frac{\sum_{i=1}^{L}X_i\times Y_i}{\sqrt{\sum_{i=1}^{L}(X_i)^2}\times\sqrt{\sum_{i=1}^{L}(Y_i)^2}},\tag{3}$$

其中 X_i, Y_i 分别为序列 $\boldsymbol{X}$, $\boldsymbol{Y}$ 中的第 i 个元素，L 为序列的长度，如果某个序列的长度小于 L，则会对该序列进行补 0 处理. 根据式(3)计算得到的结果范围为 [−1, 1]. 当结果接近 1 时，表示序列相似度高；反之，当结果接近 0 时，表示序列相似度低.

式(3)所得到的 2 个序列之间的相似度具有一定的偶然性，为此我们进一步计算 2 组序列之间的相似度. 在本文中，我们使用式(4)计算 2 组序列的平均余弦相似度，其中 $\boldsymbol{A}_i$ 表示第 1 组序列中的第 i 个序列，$\boldsymbol{B}_i$ 表示第 2 组序列中的第 j 个序列，P 和 Q 分别为序列 $\boldsymbol{X}$, $\boldsymbol{Y}$ 的序列个数.

$$avg_sim(\boldsymbol{A},\boldsymbol{B})=\frac{\sum_{i=1}^{P}\sum_{j=1}^{Q}cos_sim\left(\boldsymbol{A}_i,\boldsymbol{B}_j\right)}{P\times Q}.\tag{4}$$

各用户行为触发的控制流分组负载长度序列的平均余弦相似度如图 6 所示，每个单元格中的数值表示横轴和纵轴对应行为之间的平均余弦相似度. 不同用户行为产生的控制流分组负载长度序列之间的平均余弦相似度存在显著差异，因此使用这个特征对用户行为进行分类应该具有较好的效果.

然而，也存在一些特殊情况：a_location 和 a_filetransfer 行为的控制流序列之间具有较高的相似度，而 a_voicecall 和 a_videocall 行为内部的控制流序列相似度较低. 这种差异可能与应用程序的设计有一定的关系.

总的来说，WhatsApp 的控制流分组负载长度序列在不同用户行为下具有明显的可区分性，因此本文利用控制流分组负载长度序列来区分用户行为是可行的.

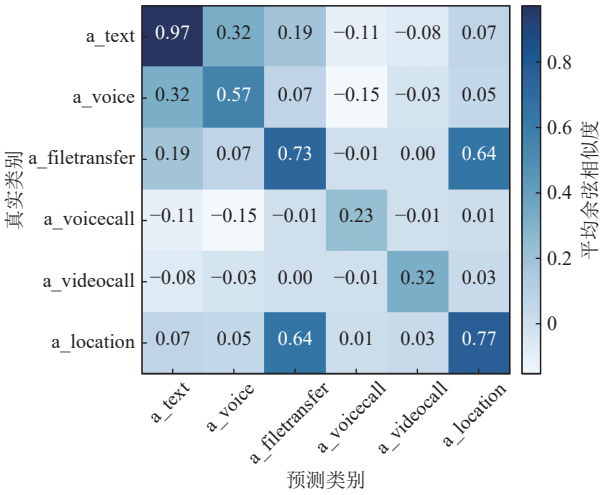


Fig. 6 Average cosine similarity of load sequence of control flow triggered by user behaviors

图 6 各用户行为触发控制流负载序列的平均余弦相似度

2.4 基于控制流负载长度序列的行为分类

卷积神经网络(convolutional neural network，CNN)和循环神经网络(recurrent neural network，RNN)主要用于自然语言处理领域的分类，但是 RNN 的计算代价比较大，而 1 维 CNN 具有轻量级的特点，相比之下，它所需的计算代价远小于 RNN，因此，在本文中我们选择了 1 维 CNN 作为其中一个分类算法. 此外，长短期记忆网络(long short-term memory，LSTM)因其能够应对 RNN 中的梯度消失问题，也得到了广泛应用，本文也将其作为分类算法之一.

本文利用了 1 维 CNN 和 LSTM 模型处理文本信息的方法，将每一个 WhatsApp 控制流数据分组负载长度视为文本向量中单词的索引值. 通过 Keras 中的 Embedding 层，本文将控制流数据分组负载长度序列映射到一个固定维度的空间中，并分别利用 1 维 CNN 和 LSTM 模型对映射后的负载长度序列进行处理. 图 7 和图 8 展示了本文的模型架构. 图 7 中，LS-CNN 模型采用了 1 个 CNN 层来提取 Embedding 层输出序列的特征，然后通过全局池化和 2 个全连接层完成了分类任务. 图 8 中，LS-LSTM 模型利用 1 个 LSTM 层来提取 Embedding 层输出序列的特征，并通过 1 个全连接层来完成分类任务. 这 2 个模型的超参数设置如表 2 所示，其中，LS-CNN 模型中的全局池化层采用默认设置，并自动设置了神经元个数.

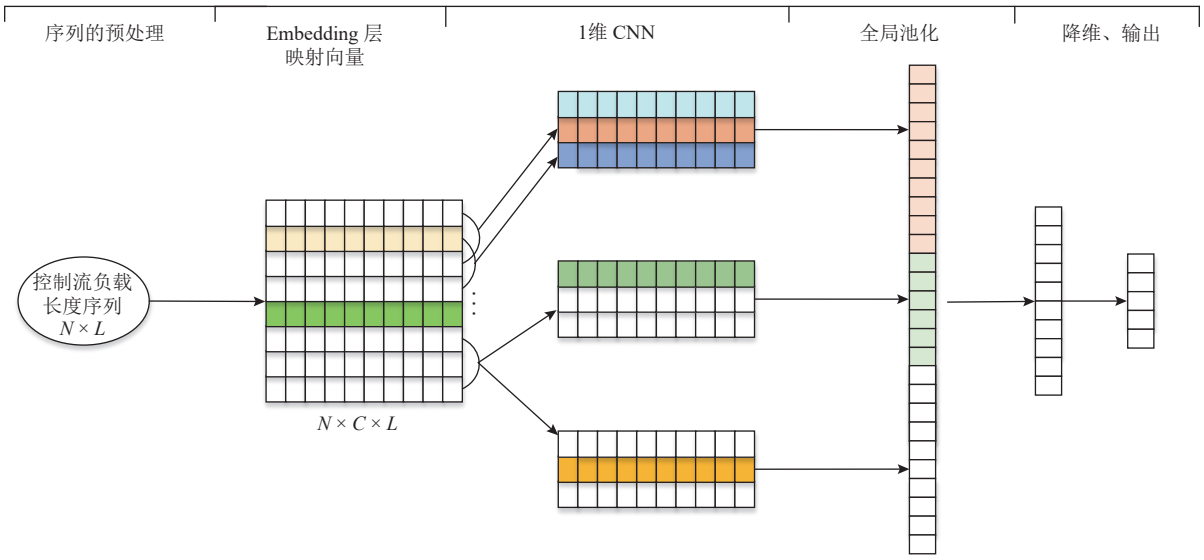


Fig. 7 Architecture of LS-CNN neural network model

图 7 LS-CNN 神经网络模型的架构

2.4.1 输入序列的预处理

输入序列的预处理有 2 个目标：第 1 个目标是将长度序列中的负值转化为非负值；第 2 个目标是确保所有序列的长度相等. 由于本文中使用正负值来区分上行数据和下行数据，但是模型的 Embedding 层只能输入非负值，因此本文通过平移将负值转换为非负值. 此外每条控制流的长度是不固定的，在训练模型时通常要求输入具有相同维度的数据，本文通过对输入的序列进行填充和截断，确保序列长度的一致性.

假设样本中有 N 个控制流分组负载长度序列，$\boldsymbol{l}_p=(L_1^p,L_2^p,\cdots,L_{n_p}^p)$ 表示第 p 个样本的长度序列，序列长度为 n_p. 式(5)描述了将序列平移为非负值的方法. 通过应用式(5)，上行数据分组负载长度的数值被映

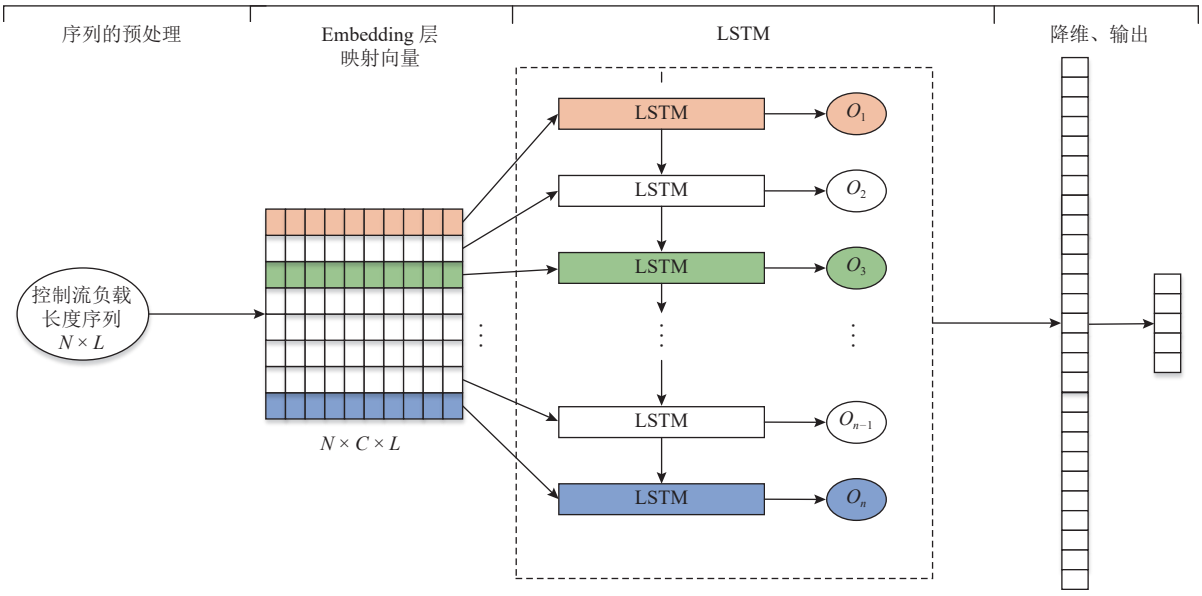


Fig. 8 Architecture of LS-LSTM neural network model

图 8 LS-LSTM 神经网络模型的架构

Table 2 Super Parameters Setting of LS-CNN and LS-LSTM

表 2 LS-CNN 和 LS-LSTM 的超参数设置

模型	Embedding 层		CNN 层		全连接层	输出层
	输入	输出	卷积核	输出	输出	输出
LS-CNN	10 000	32	3	32	64	14

模型	Embedding 层		LSTM 层	输出层
	输入	输出	隐藏层	输出
LS-LSTM	10 000	32	32	14

射到 [0, 1 500)的范围内，而下行数据分组负载长度的数值被映射到 [1 500, 3 000)的范围内. 由于网络路径的 MTU 通常小于 1 500 B, 因此经过式(5)处理的上下行序列长度数值不会发生重叠. 需要注意的是，该过程只是进行了平移，也不会改变分组载荷长度序列的语义.

$$L_i^p = L_i^p + 1\,500,\ 0 < i \leqslant n_p,\ p \in (0, N). \tag{5}$$

为了使长度序列具有相同的长度，可以通过填充或截断处理来实现，这样我们可以形成一个 $N \times L$ 维张量，其中 N 表示样本总数，L 表示序列长度. 通过对样本数据进行分析，我们发现控制流中分组的数量较少，因此我们选择将 L 设定为 100.

2.4.2 Embedding 层

Embedding 层的作用是将单词的向量表示从二进制、稀疏的高维向量映射到低维的浮点数向量，可以利用浮点数向量之间的几何距离来表征单词之间的语义关系，向量表示可以通过从数据中学习得到.

本文将经过预处理后的长度序列中的元素看作单词的索引，使用 Embedding 层将长度序列映射到一个固定维度的空间中. 因为控制序列分组个数较少，Embedding 层的输入维度被设置为 10 000, 输出维度被设置为 32.

2.4.3 卷积层和 LSTM 层

CNN 结合池化层的降采样技术展现出强大的局部感知能力，能够在不同感受野上提取特征，从而减少问题的复杂度并提升模型的泛化能力. LS-CNN 模型采用 1 维卷积层来提取控制流分组负载长度序列的局部特征，其中卷积核的大小设置为 3.

LSTM 是 RNN 的一种变体，解决了 RNN 网络难以处理的长期依赖问题，在处理长序列问题时表现出了优异的性能. LSTM 通过其内部的记忆单元，能够有效地存储信息并学习长期依赖关系. LS-LSTM 模型采用了 LSTM 层来提取控制流分组负载长度序列的特征，隐藏层节点个数设置为 32.

2.4.4 全连接层

模型中的全连接层具有 2 个作用：一是实现全局池化层或隐藏层输出的降维，同时保留有效信息；二是进行分类.

在多分类问题中，全连接层的输出结果是一个多维的特征向量，经过 softmax 函数处理后可以将其归一化为具体的类别. 在多分类任务中，输出的结果是一个 1 维向量，其中每个值代表对应的属性属于某一类别的概率. 这些概率值经过归一化处理，使得它们的总和为 1. 因为 softmax 函数能够将向量中的

每个分量映射到 [0, 1] 之间，并对整个向量的输出进行归一化，以确保所有分量输出的总和为 1，这正好符合对输出结果的要求，因此在多分类问题中，最后需要对提取的特征进行 softmax 函数处理. 本文中的 2 个模型均采用了输出维度为 14 的全连接层，并使用 softmax 函数作为激活函数.

2.4.5 优化器

本文中 2 个模型的优化器采用交叉熵作为损失函数，通过梯度下降来优化目标函数：

$$L=-\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{M}y_{ic}\lg(p_{ic})+\frac{\lambda}{2n}\sum_{i=1}^{n}\omega_i^2. \tag{6}$$

式(6)中的y_{ic}表示样本 i 是否属于类别 c，其取值为 0 或 1；p_{ic}表示样本 i 被预测为类别 c 的概率；M 为样本中待分类的类别数；n 为样本总数. 式(6)等号右边的第 1 项为常规交叉熵的表达式，第 2 项为对权重的平方和量化调整的正则化项，本文中 2 个模型的正则化参数均为 0.001.

3 实验与评估

3.1 实验数据集介绍

如表 3 所示，我们选择了 8 种常见的 WhatsApp 用户行为. 除了状态更新外，这些行为都可以分为接收和发送 2 种状态. 其中，用户通过 WhatsApp 客户端发出的行为被称为主动行为，而用户接收的行为被称为被动行为.

Table 3 User Behaviors of WhatsApp

表 3 WhatsApp 的用户行为

用户行为	描述
文本信息	接收或者发送文本或者表情符号
语音信息	接收或者发送语音
文件传输	传输图片、视频和文档等
位置服务	发送位置或者位置共享
文本状态更新	上传文本或者表情信息
状态更新	上传图片、视频等
语音电话	通过 WhatsApp 拨打电话
视频电话	通过 WhatsApp 视频通话

为了使采集的流量数据具有更丰富的类型，贴近 WhatsApp 实际使用时产生的流量，我们选择了 2 种操作系统和 4 个品牌的智能手机作为实验设备. 表 4 给出了这些设备的平台及版本. 图 9 展示了流量采集的拓扑结构，实验手机通过无线接入点 AP1 和 AP2 连接到 Internet. 我们使用自动化脚本来模拟人工操作，并实现了表 3 列出的 8 种主、被动行为的自动操作及行为的自动标签. 我们使用 Wireshark 工具来捕获实验期间产生的流量，并将其存储为 pcap 文件.

Table 4 Intelligent Device and Software Version

表 4 智能设备及软件版本

实验手机	操作系统及版本	WhatsApp 版本
Iphone11	IOS14.41	2.21.8.18
Xiaomi 9	Android9-Miui10 215.0	2.21.8.18
Redmi k-30	Android10-Miui 12 20.8.13	2.21.8.18
Huawei P40	Android10-EMUI 9.1.0.139	2.21.8.18

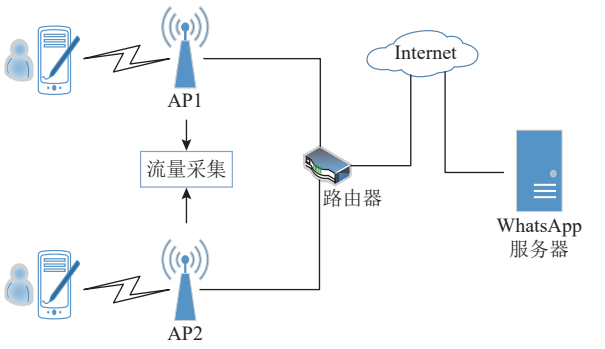


Fig. 9 Traffic collection environment of WhatsApp

图 9 WhatsApp 流量采集环境

最终，在 1 周的数据收集过程中，我们成功收集了 WhatsApp 的 14 种主、被动用户行为所产生的流量，总计达到了 8.47 GB，表 5 展示了这些数据的概要信息. 经过数据处理，我们发现并非每个样本都能够采集到控制流数据. 约 0.41% 的样本中没有发现控制数据，可能的原因有：

1)样本中控制服务的 IP 地址没有在 WhatsApp 的域名和 IP 地址的对应关系库中被记录，因此无法发现控制数据；

Table 5 Dataset Summary

表 5 数据集概要

用户行为	主动	被动	占比/%
文本信息	778	503	12.5
语音信息	764	522	12.6
文件传输	1 488	1 003	24.3
位置服务	1 356	747	20.5
文本状态更新	612		5.9
状态更新	1 157		11.3
语音电话	349	347	6.8
视频电话	331	278	6.0
总样本数		10 235	

2)由于智能手机的网络连接出现异常或者自动化控制脚本出现异常，导致用户行为没有被正常执行，这些样本被归类为异常样本，所占比例很小.

本文将上述 2 种样本标记为不可分类样本，使用本文方法无法对其进行分类.

3.2 实验结果分析

3.2.1 评价指标

本文使用精准率(*Precision*)、召回率(*Recall*)和 *F*1-*score* 来评估方法的性能，计算公式如式(7)~(9)所示. 其中，*TP*(true positive)是指被正确分类的用户行为，*FP*(false positive)是指被错误分类的用户行为，*FN*(false negative)是指被错误判定为不属于某类用户行为.

$$Precision=\frac{TP}{TP+FP}, \tag{7}$$

$$Recall=\frac{TP}{TP+FN}, \tag{8}$$

$$F1\text{-}score=\frac{2\times Precision\times Recall}{Precision+Recall}. \tag{9}$$

3.2.2 LS-CNN 模型和 LS-LSTM 模型的性能比较

首先比较了 LS-CNN 模型和 LS-LSTM 模型的分类性能. 选择序列长度为 100，随机选取数据集中 30% 的样本作为测试集，LS-CNN 和 LS-LSTM 的混淆矩阵分别如图 10 和图 11 所示. 计算各分类性能指标的算术平均数如表 6 所示，LS-CNN 和 LS-LSTM 的整体精准率分别为 97.13% 和 96.33%，可见在分类性能上 LS-CNN 优于 LS-LSTM.

然后比较了 LS-CNN 模型和 LS-LSTM 模型的时间性能. 对于模型训练过程和训练好的分类器，通过 10 次训练，并使用 1 个训练好的模型，对验证集进行

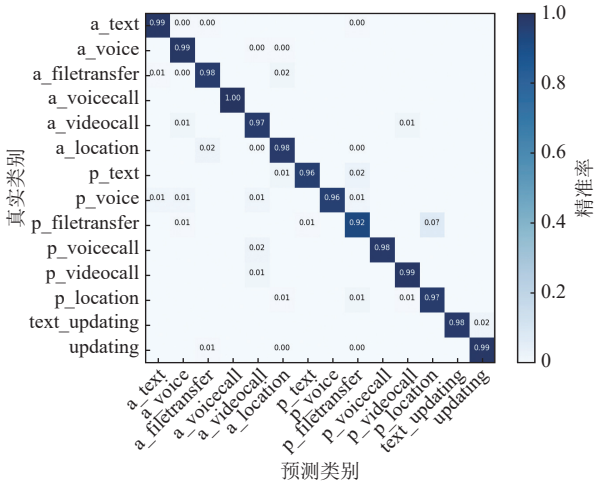


Fig. 10 Confusion matrix of LS-CNN classification results

图 10 LS-CNN 的分类结果混淆矩阵

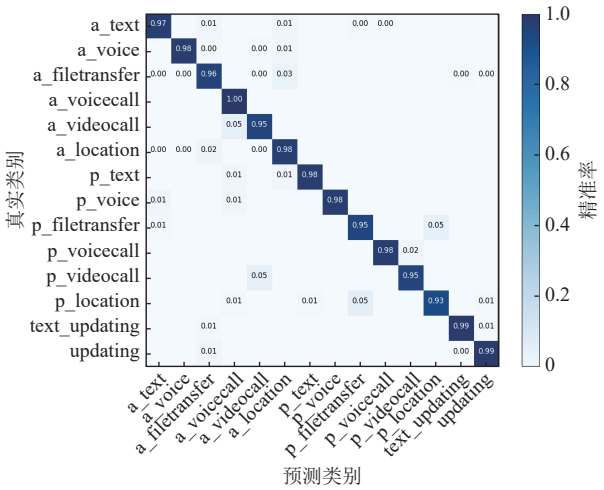


Fig. 11 Confusion matrix of LS-LSTM classification results

图 11 LS-LSTM 的分类结果混淆矩阵

Table 6 Recognition Results of LS-CNN and LS-LSTM Models

表 6 LS-CNN 与 LS-LSTM 模型的识别结果

序号	用户行为	样本数	LS-CNN			LS-LSTM		
			Precision/%	*Recall*/%	*F*1-score/%	*Precision*/%	*Recall*/%	*F*1-score/%
1	a_text	354	98.4	98.8	98.6	98.2	97.3	97.7
2	a_voice	321	97.8	99.1	98.5	99.1	98.2	98.7
3	a_filetransfer	647	98.0	97.6	97.8	96.8	96.6	96.4
4	a_voicecall	86	100.0	100.0	100.0	97.0	100.0	94.0
5	a_videocall	110	93.8	97.4	95.5	92.8	95.1	93.9
6	a_location	523	97.0	97.8	97.4	96.0	97.6	96.8
7	p_text	120	98.8	96.4	97.6	98.8	97.7	98.3
8	p_voice	136	100.0	95.8	97.8	100.0	97.9	99.0
9	p_filetransfer	263	95.5	92.4	93.9	94.8	94.8	94.8
10	p_voicecall	84	100.0	98.3	99.1	98.0	98.0	98.0
11	p_videocall	117	97.6	98.8	98.2	98.4	95.2	96.8
12	p_location	207	92.2	97.2	94.6	94.8	92.9	93.9
13	text_updating	214	100.0	98.0	99.0	98.1	98.7	98.4
14	updating	426	99.0	98.7	98.8	99.0	98.7	98.8

10 次预测验证，每次批量预测 3 000 个样本，LS-CNN 和 LS-LSTM 的训练时间和测试时间如表 7 所示，LS-CNN 在时间性能上也优于 LS-LSTM.

3.2.3 选择不同序列长度对识别结果的影响

不同的序列长度会对分类结果造成影响，如果长度过短，会导致控制流分组负载长度序列中过多的数据被截断，不能送入神经网络进行训练或预测. 表 8 给出了 LS-CNN 模型在不同长度序列下的分类性能. 当序列长度为 5 时，模型的分类精准率可达到

Table 7 Training Time and Testing Time of LS-CNN and LS-LSTM Models

表 7 LS-CNN 和 LS-LSTM 模型的训练时间和测试时间 s

模型	训练时间	测试时间
LS-CNN	53.11	0.42
LS-LSTM	212.57	0.82

Table 8 Classification Performance of LS-CNN Model with Different Sequence Lengths

表 8 不同序列长度下 LS-CNN 模型的分类性能

序号	序列长度	*Precision*/%	*Recall*/%	*F*1-*score*/%
1	5	93.3	93.2	93.2
2	10	95.8	95.8	95.8
3	20	96.4	96.4	96.4
4	30	96.8	96.8	96.7
5	40	96.9	96.8	96.8
6	50	97.0	97.0	97.0
7	60	96.8	96.7	96.7
8	70	96.4	96.3	96.3
9	80	96.8	96.8	96.8
10	90	96.5	96.4	96.4
11	100	97.7	97.6	97.6
12	110	97.6	97.5	97.6

93.3%; 随着序列长度的增加, 精准率也会增加; 当序列长度超过 20 时, 模型的分类精准率大于 96%; 当序列长度为 100 时, 分类精准率可达到 97.7%. 这说明只需检测控制流中很少的数据分组就可以达到很高的识别精度.

3.2.4 与相似工作的比较

为了验证本文提出的方法, 将本文方法与类似工作进行了比较.

由于数据采集较为方便, 微信的行为识别相对来说研究成果较多. 文献 [25] 研究了微信使用的 MMTLS 协议, 并且分析对比了不同行为所触发的不同协议流量, 使用上下行数据量、数据分组长度分布等作为特征进行了用户行为识别的分析.

图 12 显示了使用文献 [25] 的方法对本文 WhatsApp 用户数据集进行识别的结果. 在整体上文献 [25] 提出的方法识别精准率较低, 特别是文本状态更新的识别精准率仅为 65%. 显然, 使用该方法无法有效区分不同类型的文本行为, 例如发送/接收文本、文本状态更新. 此外, 由于数据集中的一些被动行为仅传输了缩略图而没有传输完整数据, 因此文件传输和部分被动行为的识别精准度较低; 同时文件缓存的存在使某些文件传输的数据大大减少, 造成使用传输数据量作为特征的方法的识别性能降低.

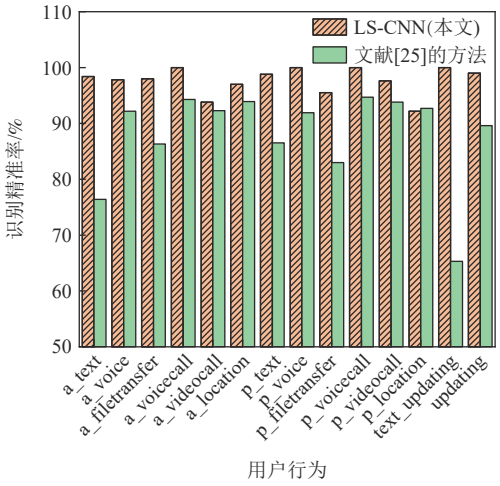


Fig. 12 Recognition results of two methods

图 12 2 种方法的识别结果

从识别的原理分析, 由于文献 [25] 提出的方法使用的数据是每条流的上行、下行的数据量和数据包长度分布, 这些数据会受到网络丢包和重传的影响, 在不同的数据传输环境中会有很大的波动, 这导致了该方法的结果不稳定、整体识别精准率低. 此外, 该方法要求处理的流量中不包含背景流量. 相比之下, 本文方法是从全部数据中提取出控制流, 背景流量的存在与否对本文方法的影响很小.

根据以上比较结果可见, 本文方法只需要对包含少量分组的序列进行检测, 就能够实现更高精度的用户行为识别, 而且不受背景流量的影响.

4 结 论

本文提出了面向加密流量的社交软件用户行为识别方法, 并以 WhatsApp 为例进行了验证. 该方法综合服务频次和服务传输特征从 WhatsApp 流量数据中提取控制流, 通过分析控制流分组负载长度序列的特征, 设计了 LS-CNN 和 LS-LSTM 这 2 种神经网络模型, 识别 WhatsApp 用户行为达到精准率 97.7%、召回率 97.6%、*F*1-*score* 97.6%. 综合与其他方法比较的结果, 本文方法不仅识别的精准度高, 而且需要分析的流量数据少.

本文方法结合应用软件的服务架构特点对数据进行了提取, 然后使用深度学习进行了分类, 使得方法具有 2 个优点:

1) 网络环境的波动对识别结果的影响较小, 同

时识别精准率较高；

2)只需少量的控制流数据分组就能实现较高的精准率，可以区分主动、被动行为，具有细粒度的行为识别能力.

作者贡献声明：吴桦提出了方案设计、写作思路，并修订论文；王磊负责实验并撰写了论文；黄瑞琪负责文献调研和全文修订；程光提供了实验指导意见；胡晓艳提供了论文修改意见.

参 考 文 献

[1] Xu Peng, Liu Qiong, Lin Sen. An improved transport layer identification of peer to peer traffic[J]. Journal of Computer Research and Development, 2008, 45(5): 794−802 (in Chinese)
(徐鹏，刘琼，林森. 改进的对等网络流量传输层识别方法[J]. 计算机研究与发展，2008，45(5)：794−802)

[2] Yoon S H, Park J W, Park J S, et al. Internet application traffic classification using fixed IP-port[C]//Proc of the 12th Asia-Pacific Network Operations and Management Symp. Berlin: Springer, 2009: 21−30

[3] Dharmapurikar S, Krishnamurthy P, Sproull T S, et al. Deep packet inspection using parallel Bloom filters[C]//Proc of the 11th Symp on High Performance Interconnects. Piscataway, NJ: IEEE, 2003: 44−51

[4] Najam M, Younis U, Rasool R U. Speculative parallel pattern matching using stride-k DFA for deep packet inspection[J]. Journal of Network and Computer Applications, 2015, 54: 78−87

[5] Altschaffel R, Clausing R, Kraetzer C, et al. Statistical pattern recognition based content analysis on encrypted network: Traffic for the teamviewer application[C]//Proc of the 7th Int Conf on IT Security Incident Management and IT Forensics. Piscataway, NJ: IEEE, 2013: 113−121

[6] Velan P, Cermak M, Celeda P, et al. A survey of methods for encrypted traffic classification and analysis [J]. International Journal of Network Management, 2015, 25(5): 355−374

[7] Kong Lingjing, Huang Guowei, Wu Keke. Identification of abnormal network traffic using support vector machine[C]//Proc of the 18th Int Conf on Parallel and Distributed Computing. Piscataway, NJ: IEEE, 2017: 288−292

[8] Li Rui, Xiao Xi, Ni Shiguang, et al. Byte segment neural network for network traffic classification[C/OL]//Proc of the 26th IEEE/ACM Int Symp on Quality of Service (IWQoS). Piscataway, NJ: IEEE, 2018 [2023-09-26].https://ieeexplore.ieee.org/document/8624128

[9] Liu Chang, He Longtao, Xiong Gang, et al. FS-Net: A flow sequence network for encrypted traffic classification[C]//Proc of the 38th IEEE Int Conf on Computer Communications. Piscataway, NJ: IEEE, 2019: 1171−1179

[10] Lotfollahi M, Siavoshani M J, Zade R S H, et al. Deep packet: A novel approach for encrypted traffic classification using deep learning[J]. Soft Computing, 2020, 24(3): 1999−2012

[11] Chen Shuo, Wang Rui, Wang Xiaofeng, et al. Side-channel leaks in web applications: A reality today, a challenge tomorrow[C]//Proc of the 31st IEEE Symp on Security and Privacy. Los Alamitos, CA: IEEE Computer Society, 2010: 191−206

[12] Tseng C M, Huang G T, Liu T J. P2P traffic classification using clustering technology[C]//Proc of the 11th IEEE/SICE Int Symp on System Integration (SII). Piscataway, NJ: IEEE, 2016: 174−179

[13] Tang Bo, Kay S, He Haibo. Toward optimal feature selection in naive Bayes for text categorization[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(9): 2508−2521

[14] Dong Shi. Multi class SVM algorithm with active learning for network traffic classification[J]. Expert Systems with Applications, 2021, 176: 114885

[15] Wang Chao, Xu Tongge, Qin Xi. Network traffic classification with improved random forest[C]//Proc of the 11th Int Conf on Computational Intelligenec and Security (CIS). Piscataway, NJ: IEEE, 2015: 78−81

[16] Wang Qinglong, Yahyavi A, Kemme B, et al. I know what you did on your smartphone: Inferring app usage over encrypted data traffic[C]//Proc of the 3rd IEEE Conf on Communications and Network Security (CNS). Piscataway, NJ: IEEE, 2015: 433−441

[17] Saltaformaggio B, Choi H, Johnson K, et al. Eavesdropping on fine-grained user activities within smartphone Apps over encrypted network traffic[C]//Proc of the 10th USENIX Workshop on Offensive Technologies (WOOT 16). Berkeley, CA: USENIX Association, 2016: 69−78

[18] Alan H F, Kaur J. Can Android applications be identified using only TCP/IP headers of their launch time traffic[C]//Proc the 9th of ACM Conf on Security and Privacy in Wireless and Mobile Networks. New York: ACM, 2016: 61−66

[19] Shen Meng, Wei Mingwei, Zhu Liehuang, et al. Classification of encrypted traffic with second-order Markov chains and application attribute bigrams[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(8): 1830−1843

[20] Sawabe A, Iwai T, Satoda K. Identification of smartphone applications by encrypted traffic analysis[C/OL]//Proc of the 16th IEEE Annual Consumer Communications and Networking Conf (CCNC). Piscataway, NJ: IEEE, 2019 [2023-09-26].https://ieeexplore.ieee.org/abstract/document/8651821

[21] Conti M, Mancini L V, Spolaor R, et al. Can't you hear me knocking: Identification of user actions on Android Apps via traffic analysis[C]//Proc of the 5th ACM Conf on Data and Application Security and Privacy. New York: ACM, 2015: 297−304

[22] Fu Yanjie, Xiong Hui, Lu Xinjiang, et al. Service usage classification with encrypted Internet traffic in mobile messaging apps[J]. IEEE Transactions on Mobile Computing, 2016, 15(11): 2851−2864

[23] He Gaofeng, Xu Bingfeng, Zhu Haiting. Identifying mobile applications for encrypted network traffic[C]//Proc of the 5th Int Conf

on Advanced Cloud and Big Data (CBD). Piscataway, NJ: IEEE, 2017: 279−284

[24] Yan Feipeng, Xu Ming, Qiao Tong, et al. Identifying wechat red packets and fund transfers via analyzing encrypted network traffic[C]//Proc of the 17th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications and 12th IEEE Int Conf on Big Data Science and Engineering (IEEE BigDataSE). Piscataway, NJ: IEEE, 2018: 1426−1432

[25] Hou Chengshang, Shi Junzheng, Kang Cuicui, et al. Classifying user activities in the encrypted Wechat traffic[C/OL]//Proc of the 37th Int Performance Computing and Communications Conf (IPCCC). Piscataway, NJ: IEEE, 2018 [2023-09-26].https://ieeexplore.ieee.org/abstract/document/8711267

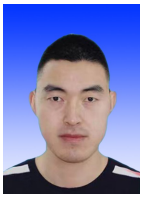
[26] Wu Hua, Wu Qiuyan, Cheng Guang, et al. Instagram user behavior identification based on multidimensional features[C]//Proc of the 39th IEEE Int Conf on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway, NJ: IEEE, 2020: 1111−1116

[27] Li Changbing, Dong Chao, Niu Kai, et al. Mobile service traffic classification based on joint deep learning with attention mechanism[J]. IEEE Access, 2021, 9: 74729−74738



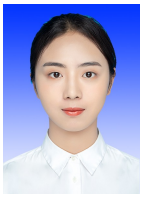
Wu Hua, born in 1973. PhD, associate professor. Member of CCF. Her main research interests include cyberspace security, network measurement, and traffic analysis.

吴　桦, 1973 年生. 博士, 副教授. CCF 会员. 主要研究方向为网络空间安全、网络测量、流量分析.



Wang Lei, born in 1986. Master. His main research interests include cyberspace security and traffic analysis.

王　磊, 1986 年生. 硕士. 主要研究方向为网络空间安全、流量分析.



Huang Ruiqi, born in 2000. Master candidate. Her main research interests include cyberspace security and traffic analysis.

黄瑞琪, 2000 年生. 硕士研究生. 主要研究方向为网络空间安全、流量分析.



Cheng Guang, born in 1973. PhD, professor. Member of CCF. His main research interests include network architecture, cyberspace security, and traffic analysis.

程　光, 1973 年生. 博士, 教授. CCF 会员. 主要研究方向为网络架构、网络空间安全、流量分析.



Hu Xiaoyan, born in 1985. PhD, associate professor. Member of CCF. Her main research interests include machine learning, future network architecture, and computer network security.

胡晓艳, 1985 年生. 博士, 副教授. CCF 会员. 主要研究方向为机器学习、未来网络体系结构、计算机网络安全.