第 13 卷 第 5 期
2024 年 9 月
集 成 技 术
JOURNAL OF INTEGRATION TECHNOLOGY
Vol. 13 No. 5
Sep. 2024

引文格式：
张莉, 谭静文, 苘大鹏, 等. 单向加密流量的移动应用程序分类技术研究 [J]. 集成技术, 2024, 13(5): 40-52.
Zhang L, Tan JW, Man DP, et al. Research on mobile application classification technology for unidirectional encrypted traffic [J]. Journal of Integration Technology, 2024, 13(5): 40-52.

单向加密流量的移动应用程序分类技术研究

张 莉 谭静文 苘大鹏 韩 帅 马书磊*

（哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001）

摘 要 在加密移动应用程序流量分类领域，传统方法均基于双向流量的特征对流量进行分类，但在实际场景中，非对称路由会导致远程网络管理员仅能获得单向流量，使得传统方法分类准确率下降。因此设计了一种仅使用单向流量特征的加密移动应用程序流量分类方法。由于下行流量包含的信息多于上行流量，因此选择对下行流量的有效负载进行分析。同时，由于移动应用程序流量具有时间、空间相关性，因此提出利用双向长短期记忆网络捕获数据流的时序相关性，并利用卷积神经网络学习特征的空间相关性，通过引入注意力层关注重要特征，进一步提高分类准确率。该方法比之前方法的使用范围广，可用于单向流量和双向流量场景，并可通过更少的特征获取更高的准确率。

关键词 加密流量；移动应用程序；单向流量；非对称路由
中图分类号 TP309.2 **文献标志码** A doi: 10.12146/j.issn.2095-3135.20240128003

Research on Mobile Application Classification Technology for Unidirectional Encrypted Traffic

ZHANG Li TAN Jingwen MAN Dapeng HAN Shuai MA Shulei*

(*College of Computer Science and Technology*, *Harbin Engineering University*, *Harbin* 150001, *China*)

**Corresponding Author*: 13810834662@126.com

Abstract In the field of encrypted mobile application traffic classification, traditional methods classify traffic based on the characteristics of bidirectional traffic. However, in actual scenarios, asymmetric routing will cause remote network administrators to only obtain unidirectional traffic, which will reduce the accuracy of traditional methods. Therefore, an encrypted mobile application traffic classification method using only one-way traffic characteristics is designed. Since downlink traffic contains more information than uplink traffic, the payload of downlink traffic is chosen for analysis. Due to the temporal and spatial correlation

收稿日期：2024-01-28 **修回日期**：2024-06-19
基金项目：国家重点研发计划项目（2021YFB3101401）
作者简介：张莉，硕士研究生，研究方向为加密流量识别；谭静文，博士研究生，研究方向为加密流量识别；苘大鹏，教授，研究方向为无线传感器网络和移动计算；韩帅，讲师，研究方向为大数据查询；马书磊（通讯作者），博士研究生，研究方向为加密流量识别，E-mail：13810834662@126.com。

of mobile application traffic, a bidirectional long short-term memory network is proposed to capture the temporal correlation of data streams, a convolutional neural network is used to learn the spatial correlation of features, and an attention layer is introduced to focus on important features to further improve the recognition accuracy. Compared with the previous methods, this method has a wider range of use, can be applied to both unidirectional and bidirectional traffic scenarios, and uses fewer features to obtain higher accuracy.

Keywords　encrypted traffic; mobile applications; unidirectional traffic; asymmetric routing

Funding　This work is supported by National Key Research and Development Program of China (2021YFB3101401)

1　引　言

随着智能手机的大面积应用，人们对移动互联网使用的需求不断增加，各类移动应用与服务也不断涌现，这对网络的管理能力提出更高要求。网络流量识别作为提高网络控制能力的必要技术之一，能通过识别用户使用的应用信息或用户在应用程序上的行为对用户进行刻画，帮助网络运营商和移动服务提供商更好地了解用户的需求和行为特征。

移动应用程序流量分类是流量识别的一个分支，目的是将收集到的流量分为预定义的类别，如按照正常或恶意流量、应用程序类型(视频应用、在线聊天应用、游戏类应用、社交应用等)、应用程序名称(微信、QQ、YouTube 等)或用户在应用程序上的行为(注册、登录、浏览等)分类。对移动应用程序流量的分析有助于确定加密应用程序流量所属的类别，并有助于故障排除、网络安全保障和服务质量管理。其中，故障排除的主要目标是定位故障网络设备、设备/软件的错误配置、丢包点、网络错误等；网络安全保障的目标是避免恶意软件的使用或防止侵犯个人隐私；服务质量管理则是为了保证终端用户感知的应用程序或服务的整体可用性。

本文将移动应用程序流量问题中试图获取并分析用户与服务器之间流量的人或系统称为攻击者。目前关于移动应用程序流量分类的相关研究均假设攻击者位于用户的本地网络中，称为本地攻击者。本地攻击者的身份可以是校园网络管理员或本地 Internet 服务提供商，能够获取本地网络网关的上行链路和下行链路流量。

但在现实场景中，可能存在一种远程攻击者，在用户与应用服务器路径上的某个关键点上，利用流量分类的结果进行网络审查，如大型自治系统或区域 Internet 服务提供商的关键路由器。由于受害者的上行链路和下行链路流量通常遵循不同的路由，因此远程攻击者仅能获得用户通信的某一单向链路流量。与完整的双向流量相比，单向流量缺乏用户与应用服务器之间的交互次数、频率及内容大小等相关信息，导致远程攻击者的性能下降。

为此，本文提出一种面向远程攻击者的单向加密移动应用程序流量识别方法，利用双向长短期记忆网络和卷积神经网络提取单向流的时间维度特征和上下文特征，进一步提高非对称路由场景下远程攻击者基于单向流量分类应用程序的准确率。

2　移动应用程序分类相关工作

目前，加密移动应用程序分类领域已取得一些研究成果。早期的加密移动应用程序流量分

类方法包括基于端口的方法和基于深度数据包检测的方法。随着加密技术和通信技术的发展，来自加密移动应用程序的流量大幅增加，导致传统的基于端口与深度数据包检测的分类方法不再适用，新的基于机器学习和深度学习的流量分类方法利用流量的时间、方向、交互特征等实现流量分类。

2.1 基于机器学习的流量分类方法

基于机器学习的流量分类方法先利用统计方法提取流量特征，再利用计算和专家经验优化特征，最后利用机器学习方法进行流量分类。Taylor 等[1]提出一种模块化的框架应用分类器 AppScanner，利用传输控制协议(Transmission Control Protocol，TCP)流中包大小的统计特征训练支持向量机和随机森林分类器，进而识别已知的应用程序。Van 等[2]提出 FlowPrint，利用数据包头部信息，如目的 IP 地址、TlS 证书等，先对所有流的目的地进行建模，再自动学习网络流量中目的地特征之间的时间相关性，最后利用随机森林分类器对流量进行预测。Al-Naami 等[3]提出 Bind，根据 TCP 流的统计特征创建受监督的应用程序指纹，利用时间特性更好地捕捉应用程序的行为。同时，Al-Naami 等[3]注意到系统的性能会随着时间的推移而衰减，因此建议在实际应用过程中定期重新训练系统。Liu 等[4]提出一种基于多属性马尔可夫概率指纹的加密流量分类方法，可将长度序列输入隐马尔可夫模型进行特征学习，主要用于解决因区分特征不足而无法获得令人满意的分类精度的问题。Petagna 等[5]证实单个应用程序也可通过 Tor 的匿名流量识别出来。Petagna 等[5]提出的有监督方法利用 TCP 流的时间、大小、数据包方向和突发特性，完成了对应用程序流量的有效分类。Panchenko 等[6]提出一种新的流量识别方法 CUMUL，具有识别精确度高、计算复杂度优于之前方法的优点。同时，Panchenko 等[6]指出在互联网规模上实现网页指纹识别是不可行的，而网站指纹识别有可能成功。Stöber 等[7]提出一种通过智能手机独有的流量模式识别智能手机型号的方法，利用数据包长度信息，并通过监督学习方法进行流量分类。Stöber 等[7]指出，在移动设备产生的流量中，约 70% 来自移动设备的后台活动，利用这些流量可进行设备指纹提取。Yang 等[8]提出针对视频流量识别的马尔可夫概率指纹，利用时序信息设计了一种加密的视频流标题识别算法，证明外部攻击者可通过分析加密视频流量的片段序列识别视频标题，并发现高阶马尔可夫链、更大的训练集及更详细的流量片段分割有助于加密视频流量识别。Zhuo 等[9]提出一种基于剖面隐马尔可夫模型的流量分类方法，通过先提取出数据包长度序列，再应用生物信息学的序列分析技术对长度序列进行建模，可实现两种可能的序列表示方法形式(流和数据包序列)及一种符号化方案。实验结果表明，该方法在网站分类方面比现有方法具有更高的准确性，且优于传统的个人网页识别技术。Conti 等[10]设计了一个利用先进的机器学习技术识别用户在移动应用程序上行为的系统，通过分析加密的网络流量，推断用户在应用程序上执行的操作。同时，Conti 等[10]尝试利用之前提出的较为有效的防御方式来防范流量分类行为，然而即便在这些防御措施下，提出的方法依旧能够获得相当高的准确率。Li 等[11]指出，当用户进行一些与特定用户界面组件密切相关的敏感操作时，往往会发生严重的隐私泄露，但现有的流量分类技术过于粗粒度，无法获取如此细粒度的敏感信息。为此，Li 等[11]提出了一种在开放世界环境中识别 Android 应用程序的方案，从而有效识别用户的操作行为。Li 等[11]还提出一种名为结构相似性的新指标，以自适应地过滤掉与监控的应用程序无关的流量段。该方法提取 123 维的统计特征，通过多种机器学习技术相结合识别用户行为。

2.2　基于深度学习的流量分类方法

基于深度学习的应用程序流量分类方法主要通过将原始流量输入深度学习模型来自动学习并提取流量特征。Wang 等[12]提出一个端到端的混合神经网络 App-Net，从原始流量中学习有效特征，用于移动应用程序识别。Zhang 等[13]提出一种自主学习框架，在主动操作期间有效地更新基于深度学习（Deep Learning，DL）的流量分类模型，该模型具有对活动网络中未知类报文进行分类的能力。上述框架由 1 个基于 DL 的分类器、1 个自学习的鉴别器和 1 个自主的自标记模型组成。其中，鉴别器和自标记过程可在主动操作时生成新的数据集，以支持分类器更新。Guarino 等[14]提出新的深度学习模型，用于实现 5 个应用程序和 3 种用户行为的分类。该分类模型针对输入参数的维度进行调整，通过提取携带有效载荷的数据包方向、有效载荷长度、有效载荷内容、TCP 窗口大小和到达时间分析流量。Telikani 等[15]提出一种新的对代价敏感的深度学习方法，用以提高深度学习分类器在流量分类中不平衡类问题的鲁棒性。Telikani 等[15]首先将数据集划分为不同的分区，并根据数据分布为每个分区创建成本矩阵，然后将代价应用到代价函数层，对分类错误进行惩罚。在 Telikani 等[15]提出的方法中，每种错误分类的代价是不同的。Liu 等[16]提出流序列网络 FS-Net，利用循环神经网络（Recurrent Neural Network，RNN）解决流量分类问题。FS-Net 是一个端到端的分类模型，可直接从原始流量中学习有效特征，进而使用学习到的特征对流量进行分类。此外，FS-Net 还采用一种多层编解码结构，利用多层双格鲁编码器学习流序列的表示，并利用多层双格鲁译码器重构原始序列，可显著提升 FS-Net 的性能。Yao 等[17]利用 RNN 对时间序列网络流量进行建模，提取流量的时间特征，并将其以矩阵的形式存储。在此基础上，Yao 等[17]通过引入注意辅助双向长短期记忆网络（BiLSTM）和分层注意网络两种模型辅助网络流量分类。Hayes 等[18]利用随机森林提取每个流量实例的指纹，从 10 万个未监控网页中准确识别出客户端正在访问的 30 个监控隐藏服务。Hayes 等[18]提出一种特征提取方式，根据森林的输出定义两个迹线之间的距离度量：给定一个特征向量，森林中的每棵树都将其与叶子标识符相关联，形成项目的叶子标识符向量。Hayes 等[18]还指出某些网站的监控隐藏服务比其他网站更容易被识别。Aceto 等[19]提出一个多模态深度学习框架（MIMETIC），可识别不同应用程序。该框架可通过学习通道内和通道间的依赖关系实现异构数据的利用与整合，克服了现有基于模态数据传输的单通道数据传输方案的局限性。Tian 等[20]提出一种基于多视图特征的应用识别（Application Identification Based on Multi-view Features，AIBMF）方法，利用卷积神经网络（Convolutional Neural Network，CNN）、RNN 等对超文本传输安全协议（Hyper Text Transfer Protocol over SecureSocket Layer，HTTPS）流量进行分类。AIBMF 的核心思想是将有效载荷卷积特征、数据包大小序列和数据包内容类型序列 3 种特征结合起来，基于不同的视图特征构建深度学习模型。Aceto 等[21]通过对标准流量分类中现有机器学习工作的分析提出一种基于深度学习体系结构的加密移动流量分类方案，为移动应用程序流量的综合评价和比较提供了一个框架。Lin 等[22]提出一种新的流量分类模型，被命名为 ET-BERT，利用 Transformer 实现加密流量双向编码。ET-BERT 从大规模未标记数据中预训练深度上下文化数据包级模型，同时，预训练模型可在少量特定任务的标记数据上进行微调。Shen 等[23]提出一种细粒度网页流量识别方法 BurNet，可同时适用于本地和远程攻击场景。BurNet 利用 CNN 构建了一个强大的具有复杂架构的分类器，旨在提高分类准确性，同时降低训练时间的复杂

性。然而，BurNet 是为网页流量识别设计的，致力于捕捉网页组成的固有模式，难以表达应用程序流量的复杂性、持续性和随机性。

3 面向单向流量的移动应用程序分类方法

本地攻击者和远程攻击者的场景如图 1 所示。由于非对称路由，远程攻击者仅能获得下行流量，使得用于流量识别的部分信息缺失，进而影响识别性能。因此，本文提出一种面向单向流量的移动应用程序分类方法。由于单向流量具有极强的时间连续性，且距离较远的数据包之间也具有一定的相关性，因此选取长短期记忆网络(Long Short-Term Memory，LSTM)对单向流量进行特征提取。此外，为更好地学习数据的上下文关系，并处理具有长依赖性的时序特征数据，本文最终采用双向长短期记忆网络(BiLSTM)作为特征提取器。本文提出的 CB 方法(CNN+BiLSTM)结合了 BiLSTM 和 CNN，利用 BiLSTM 与 CNN 模型对 TLS 流量数据不同方面的表示能力，学习流量的联合特征。CB 系统的整体框架如图 2 所示。提取原始应用

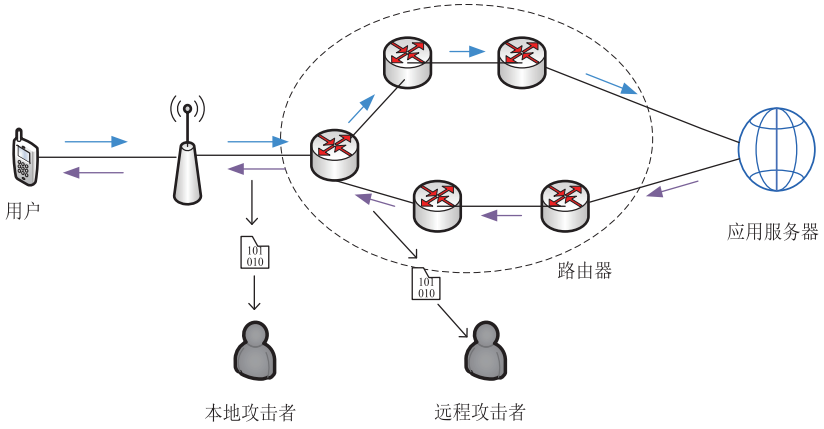


图 1 应用程序流量分类场景

Fig. 1 Application traffic classification scenario

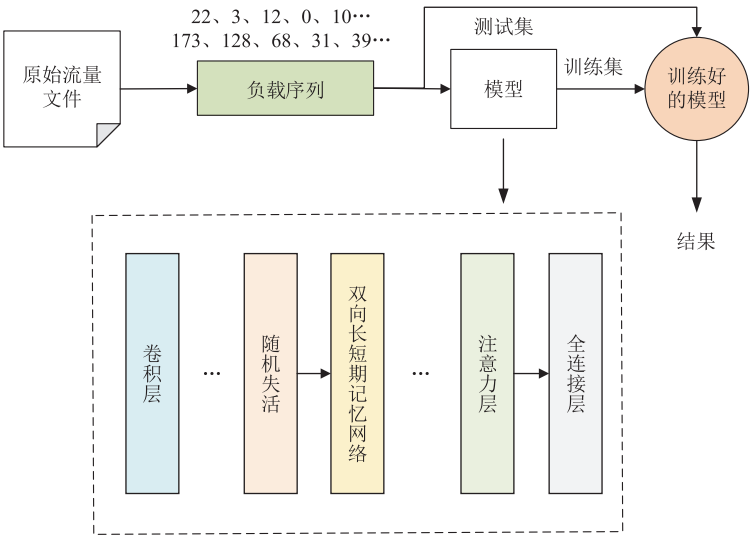


图 2 整体框架

Fig. 2 Overall framework

程序流量数据的负载信息，作为原始数据输入 CNN+BiLSTM，并运用注意力机制获取特征重要性，最后通过 Softmax 激活函数进行分类。

3.1　问题描述

将 TLS 单向流量作为分类对象，是本文提出的模型可以分类的最大粒度。本研究的任务是执行监督多类分类，以找到将 TLS 单向流映射到生成它的移动应用程序的函数。假设由应用程序 a 创建的单向流量 f 表示为 (f,a)，给定一组远程攻击者收集到的单向流量：$S=\{(f_i,a_i):f_i\in F,a_i\in A\}$，其中，$F$ 为未标记单向流的集合；A 为生成它们的所有可能应用程序的集合。本研究的目的是找到一个函数 $g:F\to A$，使得每个未标记的流 $f_i\in F$ 都可以映射到第 j 个应用 $a_j\in A$，尽可能地满足 $(f_i,a_j)\in S$。

3.2　单向流量特征提取

本文提出的特征提取模型中的 1D-CNN 由一维卷积、随机失活层组成。每层均有指定数量和指定大小的过滤器，且层上的每个过滤器都会扫描整个数据集，以提取局部特征。由于神经网络通常使用紧凑区间[－1,1]中的实数进行数学运算，因此需要先将输入数据的每个整数值元素映射到相应的实数向量，以便 LSTM 可以摄取和处理它们。

在训练阶段，对于给定的时间步长 t 来说，小批量输入是 $\boldsymbol{L}_t\in\mathbb{R}^{n\times d}$，其中，$\boldsymbol{L}$ 为输入的单向流 $\boldsymbol{S}$ 的特征序列矩阵，由数据包大小构成；d 为单个特征序列的维数；n 为特征序列示例的数量。在 BiLSTM 架构中，假设时间步长为 t 时的前向和后向隐藏状态分别为 $\overrightarrow{\boldsymbol{H}}_t\in\mathbb{R}^{n\times h}$ 和 $\overleftarrow{\boldsymbol{H}}_t\in\mathbb{R}^{n\times h}$。其中，$h$ 为隐藏单元的数量。前向和后向隐藏状态更新如式(1)～(2)所示。

$$\overrightarrow{\boldsymbol{H}}_t=\phi\left(\boldsymbol{L}_t\overrightarrow{\boldsymbol{W}}_{lh}+\overrightarrow{\boldsymbol{H}}_{t-1}\overrightarrow{\boldsymbol{W}}_{hh}+\overrightarrow{\boldsymbol{b}}_h\right) \tag{1}$$

$$\overleftarrow{\boldsymbol{H}}_t=\phi\left(\boldsymbol{L}_t\overleftarrow{\boldsymbol{W}}_{lh}+\overleftarrow{\boldsymbol{H}}_{t-1}\overleftarrow{\boldsymbol{W}}_{hh}+\overleftarrow{\boldsymbol{b}}_h\right) \tag{2}$$

其中，ϕ 为隐藏层激活函数；$\overrightarrow{\boldsymbol{W}}_{lh}$、$\overrightarrow{\boldsymbol{W}}_{hh}$、$\overleftarrow{\boldsymbol{W}}_{lh}$、$\overleftarrow{\boldsymbol{W}}_{hh}$ 为权重参数；lh 用于标识输入，$\boldsymbol{W}_{lh}$ 表示输入权重；hh 用于标识训练，$\boldsymbol{W}_{hh}$ 表示循环权重；$\overrightarrow{\boldsymbol{b}}_h$、$\overleftarrow{\boldsymbol{b}}_h$ 为偏置参数。然后，将前向和后向隐藏状态 $\overrightarrow{\boldsymbol{H}}_t$ 和 $\overleftarrow{\boldsymbol{H}}_t$ 连接起来，形成隐藏状态 $\boldsymbol{H}_t\in\mathbb{R}^{n\times 2d}$。

为进一步提高 LSTM 的学习能力，本文模型堆叠了两层 BiLSTM。第一个双向层的隐藏状态 $\boldsymbol{H}_t$ 作为输入传递给第二个双向层。最后，输出层用第二层的 $\boldsymbol{H}_t$ 计算输出 $\boldsymbol{O}_t$，如式(3)所示。

$$\boldsymbol{O}_t=\boldsymbol{H}_t\boldsymbol{W}_{hq}+\boldsymbol{b}_q \tag{3}$$

其中，$\boldsymbol{W}_{hq}$ 为输出层的权重参数；$\boldsymbol{b}_q$ 为输出层的偏置参数；q 为输出个数。由于本文的 BiLSTM 架构是一个序列到向量的 RNN 模型，因此使用最终输出向量 $\boldsymbol{O}_t$ 作为特征提取层的输出。

3.3　基于注意力机制的关键信息识别

由于移动应用程序的流量是一个按到达时间排序的数据包序列，不同数据包生成的序列在重要性和包含的信息方面会存在差异，因此使用注意力机制对数据包进行处理，使得模型能将重点放在某些应用特有的特征中，从而忽略相同的特征序列。

注意力层的计算如式(4)所示。其中，$\boldsymbol{W}_w$ 和 $\boldsymbol{b}_w$ 分别表示非线性变换层的权重和偏置；$\boldsymbol{O}_{t,i}$ 为第 i 个数据包于注意力层的输入，即 BiLSTM 层的输出，将其进行非线性变换，可得到权重 $v_{t,i}$。然后，利用 Softmax 函数生成目标注意力权重 $\boldsymbol{E}_{t,i}$，实现注意力权重概率变化的过程，如式(5)所示。

$$\boldsymbol{v}_{t,i}=\tanh\left(\boldsymbol{W}_w\boldsymbol{O}_{t,i}+\boldsymbol{b}_w\right) \tag{4}$$

$$\boldsymbol{E}_{t,i}=\frac{\mathrm{e}^{v_{t,i}}}{\sum_{j=0}^{K}\mathrm{e}^{v_{t,j}}} \tag{5}$$

其中，K 是数据包的总个数。

计算 $\boldsymbol{E}_{t,i}$ 与 $\boldsymbol{O}_{t,i}$ 的加权和，得到最终权重 $\boldsymbol{G}$，如式(6)所示。

$$\boldsymbol{G}=\sum_{i=0}^{K}\boldsymbol{E}_{t,i}\boldsymbol{O}_{t,i} \tag{6}$$

3.4 分类结果输出

由于移动加密流量分类是一个多分类问题，因此，最终的全连接层使用 Softmax 激活函数实现分类。该函数将上一层网络的向量进行归一化操作，转换为(0,1)之间的数值。

在训练阶段，Softmax 函数提供一个向量，该向量可被解释为一个概率分布，是给定输入的每个类的估计条件概率。通过检查实际类别的概率，并将估计概率与现实进行比较，可计算得到损失函数，该损失函数实际上是一个分类熵，如式(7)所示。

$$CE=\frac{1}{n}\sum_{i=1}^{N}p_i\lg p_i \tag{7}$$

其中，p_i 为预测的返回概率；N 为监控类别总数。对分类决策有信心的分类器为每个预测类别提供高概率，从而导致熵最小化。

4 实验结果与分析

4.1 数据集

本文用到的 CrossPlatform 数据集[24]由 215 个 Android 和 196 个 IOS 应用程序生成的数据组成，收集了 2017 年 9—11 月的应用程序流量，使用 3 个不同的移动设备。其中，IOS 应用是从美国、中国和印度应用商店中排名前 100 的应用中挑选出来的。而 Android 应用程序来自美国和印度 Google Play 商店的前 100 名应用程序，以及腾讯 MyApps 和 360 移动助手商店的前 100 名应用程序。每个应用程序的流量由研究人员手动操作收集得到，每个应用程序在接收真实用户输入的同时执行 3～10 min。

对于封闭世界实验(即假设用户仅访问攻击者监控的应用程序)而言，本文假设攻击者位于国内，关注的是我国常用的应用程序，因此选取 CrossPlatform 中来自中国的 IOS 数据集，数据集包含的应用程序名及流量占比如图 3 所示。

开放世界实验(即假设用户不仅访问攻击者监控的应用程序，还访问攻击者未监控的应用程序)选择 Andribus[25]作为测试集。该数据集包含来自 Google Play 商店和 15 个其他应用商店的共 103 万个 Android 应用程序的标记数据。在数据收集过程中通过 Android application exerciser monkey 模拟用户交互，并在模拟 Android 设备的沙盒环境中运行应用程序，每个应用程序模拟运行 4 min。

4.2 评价指标

在深度学习的分类任务中，经常使用以下评价指标对模型的分类性能进行评估：精确度(*precision*)、F1 值(*F*1-*score*)、准确率(*accuracy*)、召回率(*recall*)，本文也利用上述 4 个指标进行模型效果评价。这 4 个指标由真阳性(true positive，TP)、假阴性(false negative，FN)、假阳性(false positive，FP)、真阴性(true negative，TN)组成。*TP* 为实际为正样本，模型预测也为正样本的数量；*FN* 为实际为正样本，模型预测为负样本的数量；*FP* 为实际为负样本，模型预测为正样本的数量；*TN* 为实际为负样本，模型预测为负样本的数量。

$$precision=\frac{TP}{TP+FP} \tag{8}$$

$$accuracy=\frac{TP+TN}{TP+FP+FN+TN} \tag{9}$$

$$recall=\frac{TP}{TP+FN} \tag{10}$$

$$F1\text{-}score=\frac{2\times precision\times recall}{precision+recall} \tag{11}$$

由于加密应用程序流量分类问题为多分类问题，因此，*precision*、*recall*、*F*1-*score* 使用宏平均(macro)计算，即分别对每个类别求出对应的 *precision*、*recall*、*F*1-*score*，随后通过平均操作求出可用于体现多分类性能的最终指标。

在开放世界实验中，利用 $\overline{TPR}$(所有正确分

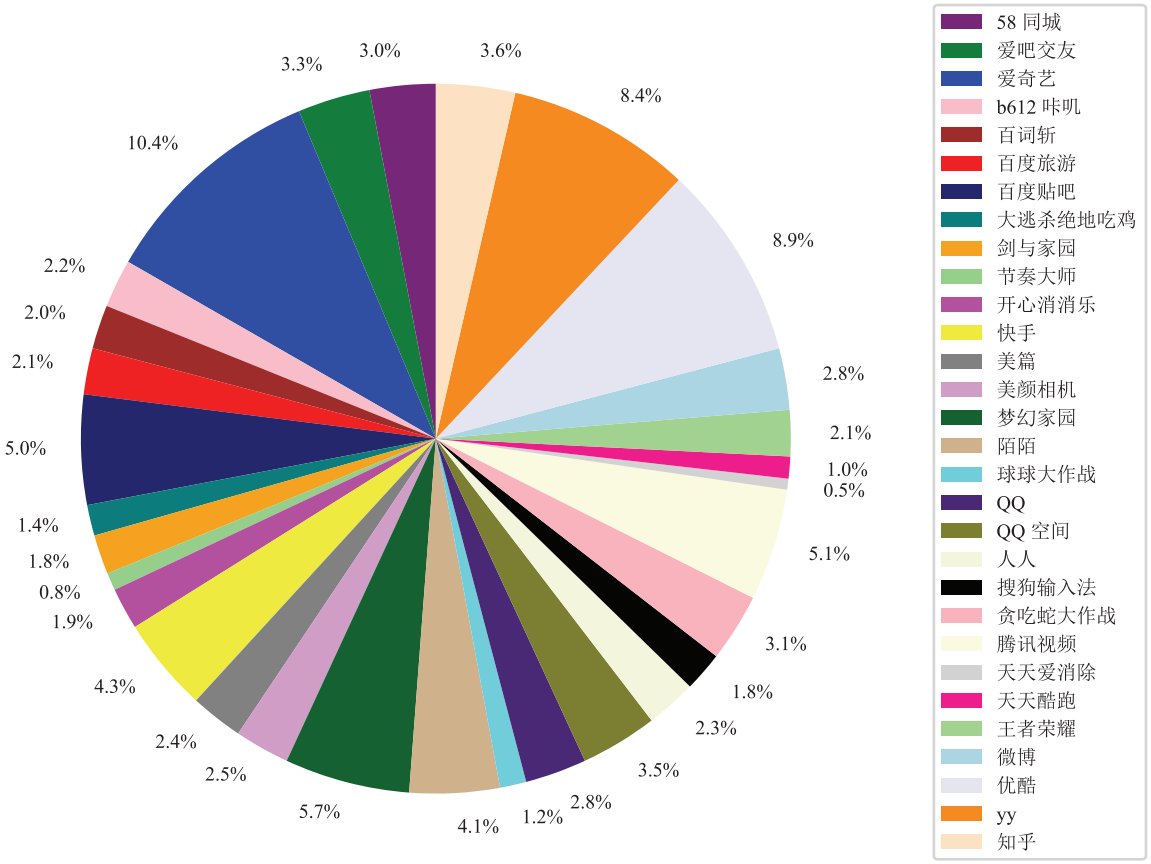


图 3　CrossPlatform IOS 数据集

Fig. 3　CrossPlatform IOS dataset

类的流量与总流量之间的比率)、$\overline{FPR}$(所有错误分类的流量与总流量之间的比率)衡量模型分类的整体性能，具体公式如式(12)～(13)所示。

$$\overline{TPR}=\frac{1}{N}\sum_{i=0}^{N}TP_i\times FlN_i \tag{12}$$

$$\overline{FPR}=\frac{1}{N}\sum_{i=0}^{N}FP_i\times FlN_i \tag{13}$$

其中，N 为监控类别总数；FlN_i 为第 i 个类别的样本总数。上述两个指标可反映模型对监控类和非监控类流量的识别能力。

4.3　实验结果与分析

4.3.1　实验设置

(1)对比实验选择

为证明本文所提 CB 方法的有效性，将 CB 方法与经典的机器学习方法 AppScanner[1]和 FlowPrint[2]，以及最经典的深度学习方法 AppNet[3]进行对比。其中，AppScanner 可提取双向流量的数据包长度、时间戳、数据包间隔时间，并可利用支持向量机和随机森林实现分类。FlowPrint 可提取数据包中的明文信息，并可基于半监督学习方法对网络流量的目的地进行建模，同样使用随机森林分类器对流量进行分类。AppNet 可提取 SSL/TLS 握手阶段交换的第一个数据包(client hello)的有效负载，并可利用 LSTM 和 CNN 学习深度特征。

(2)模型设置

模型由 2 层 CNN 层、1 层随机失活层、2 层 BiLSTM 层、1 层注意力层及 1 层全连接层组成。每层网络使用 ReLU 作为激活函数，神经元的大小分别为 128、64、64、32、30。训练过程

使用 Adamx 作为优化器，学习率设为 0.001，训练轮次为 100 轮。

(3) 参数设置

首先，由于数据包之间存在交互信息，会存在时序特征，因此需要获取多个数据包的信息，但是输入的数据包越多，时间序列越长，需要增加的模型隐藏层就越多。因此，本文针对数据包数目对分类性能的影响进行实验，如图 4(a) 所示，当选取前 6 个数据包时，会获得较好的准确率，之后再增加数据包个数，准确率提升幅度不大，因此本文选择提取 6 个数据包。实际上，前 6 个数据包中包含了有利于分类的有效信息，如数据加密时使用的加密算法、TLS 证书等信息。

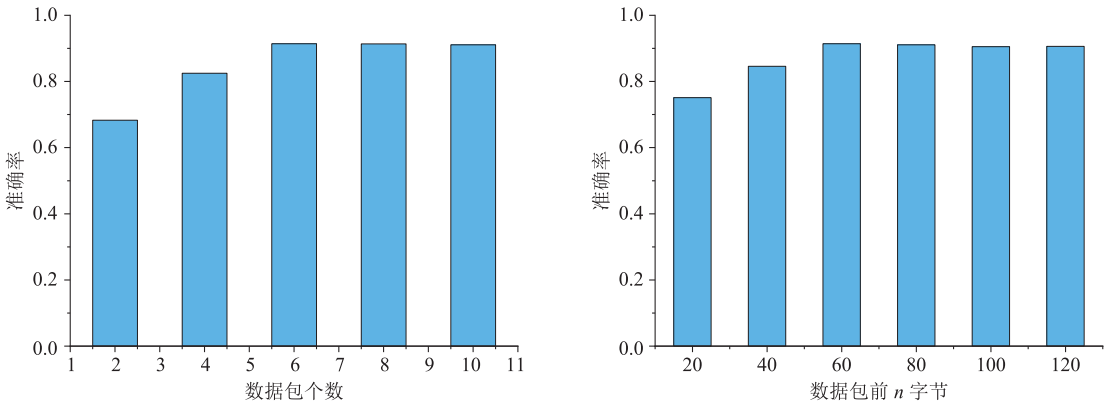
然后，由于每个数据包的长度范围为几十到上千，因此需要归一化所有数据包的长度。然而，若截取的字节数过短，则会导致分类器获取的信息不足，不能较好地进行分类；若截取的字节数较长，则又会导致计算复杂度升高。为此，本文针对数据包字节数目对分类器性能的影响进行实验，如图 4(b) 所示，在进行实验中发现，仅提取 20 个字节即可获得接近 80% 的准确率，当提取的字节数达到 60 时，会得到较高的准确率，当字节长度再次增加时，准确率提升不高，且模型训练时间增长，因此，本文将 60 字节作为统一长度。

后续所有实验遵循以下设置：每个流量样本提取 6 个数据包，每个数据包截取其有效负载的前 60 字节，将每个字节转换为 0～255 之间的数字，以进行训练和测试。此外，从数据集中选择 80% 的数据作为训练集，10% 的数据作为验证集，10% 的数据作为测试集。

4.3.2 封闭世界实验

由于目前没有实验证明由双向流量变为单向流量会对实验造成影响，因此，为确定本文提出方法具有现实意义，选择在实验中表现最好的 AppScanner 方法展示只使用单向流量特征对其性能造成的影响。

AppScanner 中使用统计特征(数据包长度、时间、传入数据包、传出数据包和双向数据包的最大值、最小值等)进行分类，使用的分类器是随机森林。在进行单向流量实验时，过滤掉传出数据包，计算流量的统计特征，将其输入随机森林分类器进行训练和预测。由表 1 可知，当仅使用单向流量的特征时，模型的 4 项指标均有所下降，准确率和召回率的下降幅度最大。当使用双向流量特征进行分类时，准确率为 89.53%，而仅使用单向流量特征进行分类时，准确率为 80.7%，下降了 8.83%。原因是 AppScanner 按照时间提取流



(a) 用于训练的数据包个数对准确率的影响　(b) 用于训练的字节数对准确率的影响

图 4　参数选择实验

Fig. 4　Parameter selection experiments

表 1　不同情况下的 AppScanner 性能

Table 1　AppScanner performance in different scenarios

输入	精确度 (%)	召回率 (%)	F1 值 (%)	准确率 (%)
双向流量	91.73	89.53	88.66	89.53
单向流量	83.95	80.70	81.38	80.70

量的数据包大小，形成突发序列(即在一定时间内的一系列数据包)，然后将流量按照上行流量与下行流量进行划分，并分别计算上下行流量的统计数据。由于 AppScanner 依赖于流量的交互信息，因此当仅使用单向流量时，AppScanner 的性能会有所下降。由此表明：现有方法在仅有单向流量的情况下，性能会大幅下降。

AppScanner、FlowPrint、App-Net 及本文提出的方法在 CrossPlatform 上的精确度、召回率、F1 值、准确率见表 2。其中，仅本文的 CB 方法使用单向流量，其他方法均使用双向流量进行实现。目的是证明本文方法在仅有单向流量的情况下，依旧能获得和其他使用双向流量的方法相同，甚至更好的性能。

由表 2 可知，CB 仅使用单向流量的特征即可以较高的准确率将网络流量正确分类到相应的应用程序，在 CrossPlatform Android 数据集上，精确度可达 90.19%，和使用双向流量的 AppScanner 方法相当，比 FlowPrint 和 App-Net 分别高出 9.32% 和 23.5%。

本文及对比实验方法在 CrossPlatform IOS 上的混淆矩阵如图 5 所示。由图 5(a)可知，AppAcanner 的总体分类性能较好，但会存在其余类被错误分类为第 7、30 类的问题，部分类别的准确率不高。由图 5(b)可知，FlowPrint 方法的总体分类性能虽然较高，但是会存在部分类别的准确率很差的情况，如将第 8、24 类的大部分示例都进行了错误分类，它们的准确率均低于 20%。而 App-Net 仅为一小部分应用程序提供了较高的准确率。本文方法在对所有类型的应用程序进行分类方面表现最好，总体分类性能最高。

4.3.3 开放世界实验

开放世界场景中的分类问题需要确定测试流量是属于“监控”还是“未监控”的应用程序。本文重新训练了一个多分类模型，在监控类别的基础上增加了一个未知类别。如果分类器的预测类别为未知类或预测概率的最大值小于阈值，则认为输入的样本属于未监控网站。实验所用数据集包含监控类和未监控类，监控类来自 CrossPlatfronm Android 数据集，未监控类来自

表 2　对比实验

Table 2　Comparison experiments

数据集	模型	精确度 (%)	召回率 (%)	F1 值 (%)	准确率 (%)
CrossPlatform Android	AppScanner	91.73	89.53	88.66	89.53
	FlowPrint	80.87	77.66	73.42	77.66
	App-Net	66.69	68.54	63.54	68.54
	CB	90.19	91.35	90.66	91.35
CrossPlatform IOS	AppScanner	92.23	84.00	83.96	84.00
	FlowPrint	88.57	89.87	84.49	89.87
	App-Net	70.39	67.69	67.78	67.69
	CB	89.36	90.48	89.41	90.48

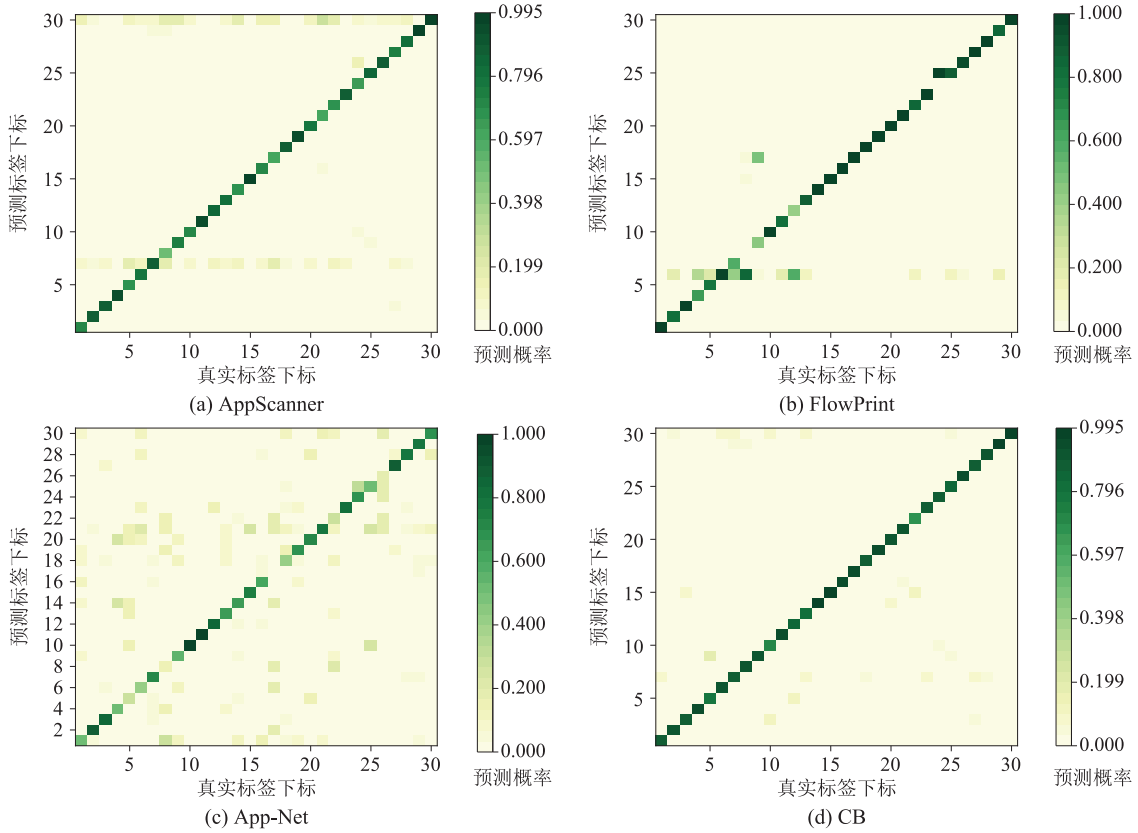


图 5 混淆矩阵

Fig. 5 Confusion matrixes

Andribus 数据集，在模型训练过程中将未知类的示例由 2 000 增至 50 000（每个应用一个样本），测试开放世界大小对模型性能的影响。

世界大小对 $\overline{TPR}$ 的影响如图 6(a)所示，世界大小对 $\overline{FPR}$ 的影响如图 6(b)所示。由图 6(a)可知，随着世界大小的增大，所有模型的 $\overline{TPR}$ 都明显下降，但在任何情况下，CB 的性能始终最好。其中，App-Net 的下降幅度最大，为 16%；

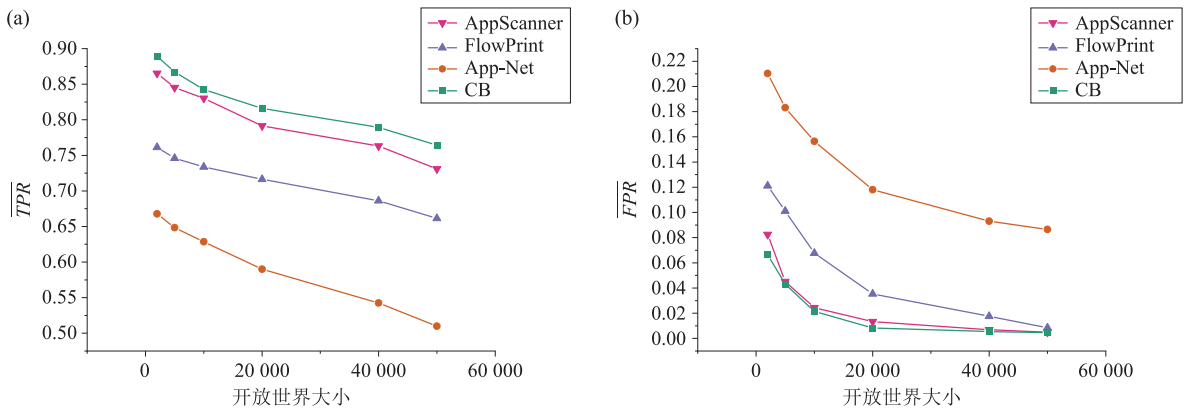


图 6 开放世界大小对模型性能的影响

Fig. 6 Impact of open world size on model performance

FlowPrint 的下降幅度最小，为 10%。

由于受监控的数据集和未受监控的数据集中的数据总量严重不平衡，仅使用 $\overline{TPR}$ 和 $\overline{FPR}$ 无法准确反映模型对正类和负类的识别能力。因此，还利用精度-召回曲线评估模型在开放世界场景中的性能。本实验将未受监控的网站数量控制在 20 000，精度-召回曲线如图 7 所示。本文的 CB 方法优于其他方法。在 CrossPlatform Android 数据集中，CB 方法对大多数阈值都比较有效。由图 7 可知，AppScanner 方法的精度较高，召回率的范围较广，而 FlowPrint 和 App-Net 的召回率和精确度都比本文方法低。

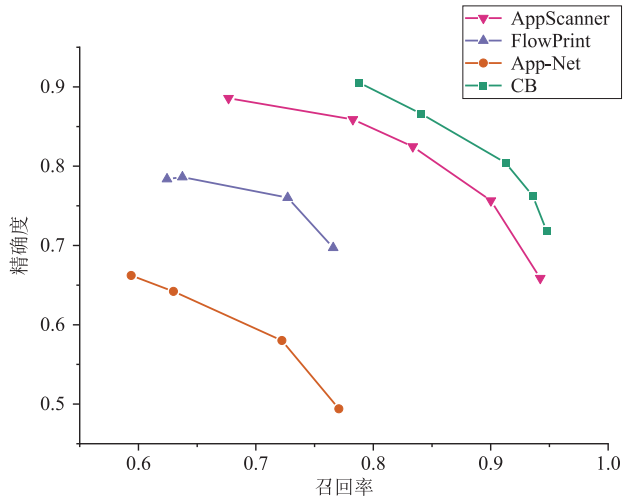


图 7　精度-召回曲线

Fig. 7　Precision-Recall Curves

5　结　论

针对远程攻击者仅能获取单向流量，导致传统的基于双向流量特征的方法准确率下降的问题，本文提出面向单向流量的移动应用程序分类方法。该方法将卷积神经网络、双向长短期记忆网络、注意力机制进行结合，利用卷积神经网络提取网络流量的空间特征，利用双向长短期记忆网络获取流量的时序信息，利用注意力机制进行特征选择。该方法比传统方法使用更少的特征，获得了更高的性能，在 CrossPlatform Android 数据集上的准确率比 AppScanner 方法高 1.82%，在 CrossPlatform IOS 数据集上的准确率比 AppScanner 方法高 6.48%。在开放世界场景中，本文提出的方法比传统方法的准确率高 2%。

参 考 文 献

[1] Taylor VF, Spolaor R, Conti M, et al. AppScanner: automatic fingerprinting of smartphone apps from encrypted network traffic [C] // Proceedings of the 2016 IEEE European Symposium on Security and Privacy, 2016: 439-454.

[2] Van Ede T, Bortolameotti R, Continella A, et al. FlowPrint: semi-supervised mobile-app fingerprinting on encrypted network traffic [C] // Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2020, 2020: 1-18.

[3] Al-Naami K, Chandra S, Mustafa A, et al. Adaptive encrypted traffic fingerprinting with bi-directional dependence [C] // Proceedings of the 32nd Annual Conference on Computer Security Applications, 2016: 177-188.

[4] Liu C, Cao ZG, Xiong G, et al. MaMPF: encrypted traffic classification based on multi-attribute Markov probability fingerprints [C] // Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service, 2018: 1-10.

[5] Petagna E, Laurenza G, Ciccotelli C, et al. Peel the onion: recognition of Android apps behind the Tor network [C] // Proceedings of the 15th International Conference on Information Security Practice and Experience, 2019: 95-112.

[6] Panchenko A, Lanze F, Zinnen A, et al. Website fingerprinting at Internet scale [C] // Proceedings of the 23rd Internet Society (ISOC) Network and Distributed System Security Symposium (NDSS 2016), 2016: 21-24.

[7] Stöber T, Frank M, Schmitt J, et al. Who do you Sync you are? Smartphone fingerprinting via application behaviour [C] // Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2013: 7-12.

[8] Yang LM, Fu SJ, Luo YC, et al. Markov probability fingerprints: a method for identifying encrypted video traffic [C] // Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN), 2020: 283-290.

[9] Zhuo ZL, Zhang Y, Zhang ZL, et al. Website fingerprinting attack on anonymity networks based on profile hidden Markov model [J]. IEEE Transactions on Information Forensics and Security, 2017, 13(5): 1081-1095.

[10] Conti M, Mancini LV, Spolaor R, et al. Can't you hear me knocking: identification of user actions on Android apps via traffic analysis [C] // Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, 2015: 297-304.

[11] Li JF, Zhou H, Wu SH, et al. FOAP: fine-grained open-world Android app fingerprinting [C] // Proceedings of the 31st USENIX Security Symposium, 2022: 1579-1596.

[12] Wang X, Chen SH, Su JS. App-Net: a hybrid neural network for encrypted mobile traffic classification [C] // Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020: 424-429.

[13] Zhang JL, Li FH, Ye F, et al. Autonomous unknown-application filtering and labeling for DL-based traffic classifier update [C] // Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, 2020: 397-405.

[14] Guarino I, Aceto G, Ciuonzo D, et al. Classification of communication and collaboration apps via advanced deep-learning approaches [C] // Proceedings of the 2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, 2021: 1-6.

[15] Telikani A, Gandomi AH, Choo KKR, et al. A cost-sensitive deep learning-based approach for network traffic classification [J]. IEEE Transactions on Network and Service Management, 2022, 19(1): 661-670.

[16] Liu C, He LT, Xiong G, et al. FS-Net: a flow sequence network for encrypted traffic classification [C] // Proceedings of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019: 1171-1179.

[17] Yao HP, Liu C, Zhang PY, et al. Identification of encrypted traffic through attention mechanism based long short term memory [J]. IEEE Transactions on Big Data, 2022, 8(1): 241-252.

[18] Hayes J, Danezis G. k-fingerprinting: a robust scalable website fingerprinting technique [C] // Proceedings of the 25th USENIX Security Symposium, 2016: 1187-1203.

[19] Aceto G, Ciuonzo D, Montieri A, et al. MIMETIC: mobile encrypted traffic classification using multimodal deep learning [J]. Computer Networks, 2019, 165: 106944.

[20] Tian M, Chang P, Sang YF, et al. Mobile application identification over HTTPS traffic based on multi-view features [C] // Proceedings of the 2019 26th International Conference on Telecommunications, 2019: 73-79.

[21] Aceto G, Ciuonzo D, Montieri A, et al. Mobile encrypted traffic classification using deep learning [C] // Proceedings of the 2018 Network Traffic Measurement and Analysis Conference (TMA), 2018: 1-8.

[22] Lin XJ, Xiong G, Gou GP, et al. ET-BERT: a contextualized datagram representation with pre-training transformers for encrypted traffic classification [C] // Proceedings of the ACM Web Conference 2022, 2022: 633-642.

[23] Shen M, Gao ZB, Zhu LH, et al. Efficient fine-grained website fingerprinting via encrypted traffic analysis with deep learning [C] // Proceedings of the 2021 IEEE/ACM 29th International Symposium on Quality of Service, 2021: 1-10.

[24] Alan HF, Kaur J. Can Android applications be identified using only TCP/IP headers of their launch time traffic? [C] // Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2016: 61-66.

[25] Lindorfer M, Neugschwandtner M, Weichselbaum L, et al. ANDRUBIS -- 1,000,000 Apps Later: a view on current Android malware behaviors [C] // Proceedings of the 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, 2014: 3-17.