面向单向加密流量的移动应用程序分类技术研究

张莉[1]，谭静文[1]，苘大鹏[1]，韩帅[1]，马书磊[1]

[1]（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，传统方法都是基于双向流量的特征对流量进行分类，但在实际场景中，不对称路由会导致远程监控者只能获得单向流量，使得传统方法分类准确率下降。因此本文设计一种只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，本文选择对下行流量的有效负载进行分析。由于移动应用程序流量具有时间和空间的相关性，提出使用双向长短期记忆网络捕获数据流的时序相关性、使用卷积神经网络学习特征的空间相关性，并引入注意力层关注重要特征来进一步提高分类准确率。该方法相较于之前的方法，它的使用范围更广，能够同时适用于单向流量和双向流量场景，使用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[1], TAN Jingwen[1], MAN Dapeng[1], HAN Shuai[1], MA Shulei[1]

[1] (Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China)

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 monitors to only obtain unidirectional traffic, which will reduce the accuracy of traditional methods. Therefore, this paper designs an encrypted mobile application traffic classification method using only one-way traffic characteristics. Since downlink traffic contains more information than uplink traffic, this paper chooses to analyze the payload of downlink traffic. Due to the temporal and spatial correlation 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.

Key Words: Encrypted traffic；Mobile applications；Unidirectional Traffic；Asymmetric routing

来稿日期：2024-01-28 修回日期：2024-06-19
基金项目：国家重点研发计划子课题（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。

Funding: This project is supported by the National Key R&D 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了更高的要求。网络流量识别作为提高网络控制能力的必要技术之一，能够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：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）。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从而处理某些事件，如：（1）故障排除：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；（2）安全：避免恶意软件的使用或防止侵犯个人隐私；（3）服务质量（Quality of Service，Qos）管理：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。此类攻击者可以是校园网络管理员或本地 Internet 服务提供商（Internet Service Provider，ISP），他们可以获得通过本地网络网关的上行链路和下行链路流量。

然而在现实场景中，可能存在一种远程攻击者，在用户和相应应用服务器之间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，例如大型自治系统（Autonomous System，AS）或区域 ISP 的关键路由器。这类远程攻击者由于非对称路由，即受害者的上行链路和下行链路流量通常遵循不同的路由，只能获得用户通信的某一单向链路的流量。本地攻击者和远程攻击者的场景示意图如图 1 所示。相比于完整的双向流量，单向流量缺乏了用户与应用服务器之间的交互次数、频率以及内容大小等相关信息，导致远程攻击者的性能下降。

为此，本文提出了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来提取单向流的时间维度特征和上下文特征，进一步提高不对称路由场景下，远程攻击者基于单向流量分类应用程序的准确率。

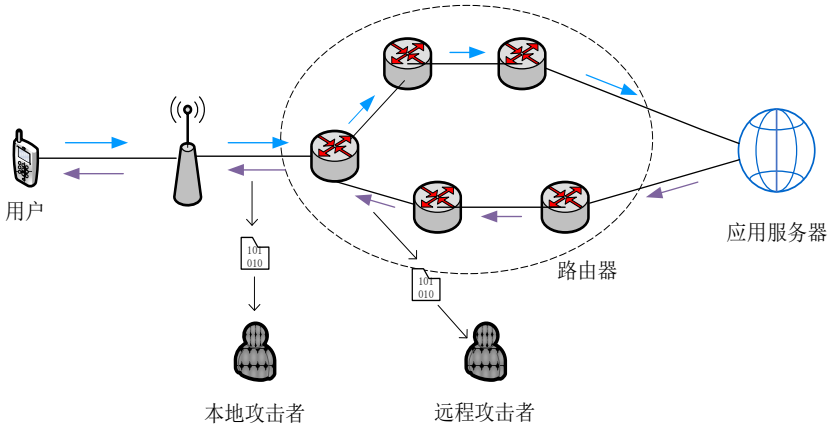

图 1 应用程序流量分类场景图

2 移动应用程序分类相关工作

目前在加密移动应用程序分类领域，已经取得了一些研究成果。早期的加密移动应用程

序流量分类方法有基于端口的方法和基于 DPI 的方法。而后续随着加密技术和通信技术的发展，来自加密移动应用程序流量的大幅增加，导致传统的基于端口与 DPI 的分类方法已不再适用，出现了基于机器学习和深度学习的方法，它们使用流量的时间、方向、交互特征等来实现流量分类。

2.1 基于机器学习的流量分类方法

基于机器学习的方法会使用统计学和数学知识提取出流量特征，并通过计算和专家经验优化特征，再使用机器学习方法进行流量分类。文献[1]提出了一种模块化的框架应用分类器 AppScanner。其使用 TCP 流中包大小的统计特征来训练支持向量机和随机森林分类器来识别已知的应用程序。文献[2]提出了 FlowPrint，它使用数据包头部信息，如：目的 IP 地址，TlS 证书等先对所有流的目的地进行建模，自动学习网络流量中目的地特征之间的时间相关性，并使用随机森林分类器对流量进行预测。文献[4]提出了 Bind，Bind 根据 TCP 流的统计特征创建受监督的应用程序指纹，使用时间特性来更好地捕捉应用程序的行为。同时作者注意到系统的性能会随着时间的推移而衰减，并建议在实际应用过程中应定期重新训练系统。文献[5]提出了一种基于多属性马尔可夫概率指纹的加密流量分类方法，将长度序列输入隐马尔可夫模型进行特征学习。它主要用于解决由于区分特征不足，而无法获得令人满意的分类精度的问题。文献[6]证实了单个应用程序也可以通过 Tor 的匿名流量识别出来。这个监督方法利用了 TCP 流的时间、大小、数据包方向和突发特性。文献[7]提出了一种新的流量识别方法 CUMUL，该方法识别精确度高并且计算复杂度优于之前的方法。同时文献指出在互联网规模上实现网页指纹是不可行的，而 WF 有机会实现。文献[8]提出了一种通过智能手机独有的流量模式来识别智能手机型号的方法，它利用数据包长度信息，使用监督学习方法进行流量分类。文献指出移动设备产生的流量中，有大约 70%都来自移动设备的后台活动，利用这些流量可以进行设备指纹的提取。文献[9]提出了针对视频流量进行识别的马尔可夫概率指纹。该文献利用时序信息，设计了一种加密的视频流标题识别算法。文献证明了外部攻击者可以通过分析加密视频流量的片段序列来识别视频标题。同时发现高阶马尔可夫链、更大的训练集以及更详细的流量片段分割，有助于加密视频流量的识别。文献[10]提出了一种基于 PHMM 的流量分类方法，该方法提取出数据包长度序列，然后应用生物信息学的序列分析技术对长度序列进行建模。文献实现了两种可能的序列表示方法形式（流和数据包序列）以及一种符号化方案。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，并且优于传统的个人网页识别技术。文献[11]设计了一个利用先进的机器学习技术来识别用户在移动应用程序上行为的系统。文献提出了一个框架来分析加密的网络流量，并推断出用户在安装在手机上的应用程序上执行了哪些特定的操作。同时文献采用了之前提出的比较有效的防御方式来防范流量分类行为，但是提出的方法依旧能够得到相当的准确率，因此得出结论目前没有针对流量分类的较为有效的防御手段。文献[12]提出了一种在开放世界环境中识别 Android 应用程序的方法级细粒度用户操作的方案，称为 FOAP。文献指出当用户进行一些与特定 UI 组件密切相关的敏感操作时，往往会发生严重的隐私泄露，但是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。FOAP 提出了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，该方法提取了 123 维的统计特征，使用多种机器学习技术相结合用来识别用户行为。

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，主要将原始流量输入深度学习模型，自动学习流量特征。文献[3]提出了一个端到端混合神经网络 App-Net，从原始流量中学习有效特征，用于移动应用程序识别。文献[13]提出了一种自主学习框架，在主动操作期间有效地更新基于 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。该框架由一

个基于 DL 的分类器、一个自学习的鉴别器和一个自主的自标记模型组成。其中鉴别器和自标记过程可以在主动操作时生成新的数据集，以支持分类器更新。文献[14]提出了新的模型用于实现五个应用程序和三种用户行为的分类，模型针对输入参数的维度进行调整，提取出（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来分析流量。文献[15]提出了一种新的对代价敏感的深度学习方法，以提高深度学习分类器对流量分类中不平衡类问题的鲁棒性。文献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。在被提出的方法中，每种类型错误分类的代价是不同的。同时文献在堆叠自编码器（Stacked Auto-Encoder，SAE）和 CNN 中测试了其效果。文献[16]使用循环神经网络（Recurrent Neural Networks，RNN）来解决流量分类问题，提出了流序列网络 FS-Net。FS-Net 是一个端到端的分类模型，可以直接从原始流量中学习有效特征，然后再使用学习到的特征对其进行分类。此外，FS-Net 还采用了一种多层编解码结构，使用多层双格鲁编码器学习流序列的表示，并用多层双格鲁译码器重构原始序列。通过对 18 个应用的实际数据集进行综合实验，结果表明 FS-Net 具有很好的性能。文献[17]使用循环神经网络对时间序列网络流量进行建模，文献提取出流量的时间特征，并将其以矩阵的形式存储。在此基础上，引入了注意辅助 BiLSTM 和分层注意网络（HAN）两种模型来辅助网络流量分类。文献[18]使用随机森林来提取每个流量实例的指纹，从 10 万个网站中识别出 30 个监控的网站。文献提出了一种特征提取方式，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，文献同时指出某些网站比其他网站更容易被识别。文献[19]提出了一个多模态深度学习框架（MITETIC），用来识别不同的应用程序，该框架能够通过学习通道内和通道间的依赖关系来实现数据异构性的资本化，克服了现有基于模态数据传输的单通道数据传输方案的性能局限性。文献[20]提出了 AIBMF，使用 CNN 和 RNN 等对 HTTPS 流量进行分类的细粒度方法。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三种特征结合起来。基于不同的视图特征，构建深度学习模型。文献[21]通过对标准流量分类中现有的机器学习工作的分析，提出了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。文献[22]提出了一种新的流量分类模型，被命名为 ET-BERT，使用 Transformer 实现加密流量双向编码器，它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，同时预训练模型可以在少量特定于任务的标记数据上进行微调。文献[25]提出了一种细粒度网页流量识别方法 BurNet，能够同适用于本地和远程攻击场景。BurNet 利用 CNN 构建了一个强大的分类器，其复杂的架构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、持续性和随机性特点。

3 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，因而选取长短期记忆网络（LSTM）来对其进行特征提取。此外，为了更好地学习到数据的上下文关系，适合处理具有长依赖性的时序特征数据，最终利用双向长短期记忆网络（BiLSTM）最终的特征提取器。本文提出的方法 CB 结合了 BiLSTM 和卷积神经网络（CNN），利用它们对 TLS 流量数据不同方面的表示能力，学习其联合特征，系统整体框架如图 2 所示。对于原始应用程序流量数据，提取其负载信息作为原始数据，输入 CNN+BiLSTM 网络，并且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。

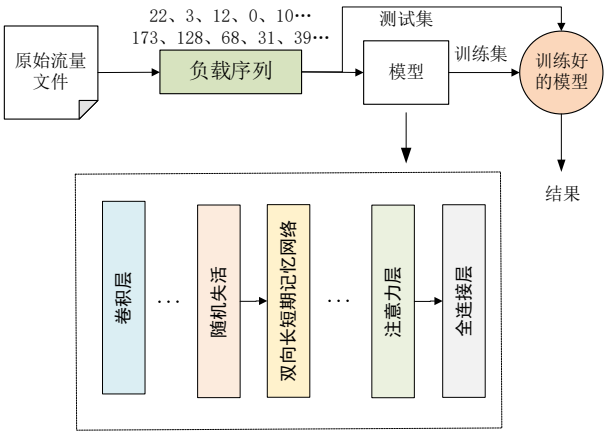

图 2 整体框架图

3.1 问题描述

在本文中，将 TLS 单向流量作为分类对象，同时它也是本文提出的模型可以分类的最大粒度。本文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$F$是未标记单向流的集合，$A$是生成它们的所有可能应用程序的集合，本文的目的是找到一个函数$g:F\to A$，使得每个未标记的流 $f_s\in F$ 都可以映射到一个应用$a_t\in A$，尽可能地满足$(f_s,a_t)\in S$。

3.2 单向流量特征提取

本文的特征提取模型中的 1D-CNN 由一维卷积、随机失活层组成。每个层都有指定数量的指定过滤器大小的过滤器，并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，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 网络可以摄取和处理它们。

在训练阶段，对于给定的时间步长t，小批量输入是$L_t\in\mathbb{R}^{n\times d}$，其中，$L$ 为输入的单向流 S 的特征序列矩阵，由数据包大小构成。d 表示单个特征序列的维数，n 是特征序列示例的数量。在 BiLSTM 架构中，假设此时间步的前向和后向隐藏状态分别为$\overrightarrow{H}_t\in\mathbb{R}^{n\times h}$和$\overleftarrow{H}_t\in\mathbb{R}^{n\times h}$。其中，$h$ 表示隐藏单元的数量。计算前向和后向隐藏状态更新的公式如（1.1）和（1.2）。

$$\overrightarrow{H}_t=\phi(L_t\overrightarrow{W}_{lh}+\overrightarrow{H}_{t-1}\overrightarrow{W}_{hh}+\overrightarrow{b}_h) \tag{0.0}$$

$$\overleftarrow{H}_t=\phi(L_t\overleftarrow{W}_{lh}+\overleftarrow{H}_{t-1}\overleftarrow{W}_{hh}+\overleftarrow{b}_h) \tag{0.0}$$

其中，ϕ 是隐藏层激活函数，模型参数包括权重参数$\overrightarrow{W}_{lh}$、$\overrightarrow{W}_{hh}$、$\overleftarrow{W}_{lh}$、$\overleftarrow{W}_{hh}$和偏置参数$\overrightarrow{b}_h$、$\overleftarrow{b}_h$。然后，将前向和后向隐藏状态$\overrightarrow{H}_t$和$\overleftarrow{H}_t$连接起来形成隐藏状态$H_t\in\mathbb{R}^{n\times 2d}$。

为了进一步提高 LSTM 网络的学习能力，在本文的模型中堆叠了两层 BiLSTM。第一个双向层的隐藏状态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H_t计算输出O_t，如公式（1.3）。

$$O_t=H_tW_{hq}+b_q \tag{0.0}$$

其中权重参数W_{hq}和偏置参数b_q为输出层的模型参数，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，使用最终输出向量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

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，不同的数据包其生成的序列在重要性和包含的信息方面会存在差异，所以使用注意力机制对其进行处理，使得模型能够将重点放在某些应用特有的特征中，而忽略相同的特征序列。

注意力层的计算公式如（4）所示。其中O_t是注意力层的输入，即 BiLSTM 层的输出，将其通过非线性变换得到权重v_t。然后 Softmax 函数生成目标注意力权重E_t，实现注意力权重概率变化的过程，如公式（1.5）。

$$v_t = \tanh(W_w O_t + b_w) \tag{0.0}$$

$$E_t = \frac{e^{v_t}}{\sum_{K-1}^{T}\sum e^{v_t}} \tag{0.0}$$

最后计算E_t与O_t的加权和得到最终权重S，如公式（1.6）。

$$S = \sum E_t O_t \tag{0.0}$$

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，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 函数为本文提供了一个向量，该向量可以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，可以通过计算得到损失函数，它实际上是一个分类熵，公式如（1.7）。

$$E = \frac{1}{n}\sum_{i=1}^{N} p_i \log p_i \tag{0.0}$$

其中，p_i是预测的返回概率、N是监控的类别的数。对其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

本文用到的是 CrossPlatform[23]数据集，该数据集由 215 个 Android 和 196 个 IOS 应用程序生成的数据组成，该数据集收集了 2017 年 9 月到 11 月的应用程序流量，使用三个不同的移动设备。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谷歌 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 到 10 分钟。

对于封闭世界实验（即假设用户只会访问攻击者监控的应用程序），我们假设攻击者位于国内，关注的是中国常用的应用程序，因此我们选取了 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。

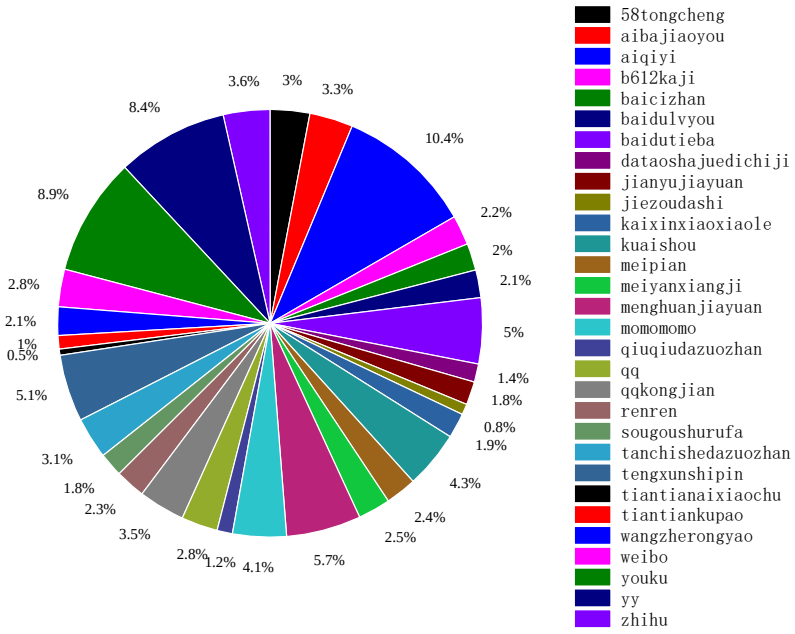

图 3 CrossPlatform IOS 数据集

对于开放世界实验（即假设用户不光会访问攻击者监控的应用程序，还会访问其他攻击者未监控的应用程序），选择 Andribus[24]作为开放世界的测试集。该数据集包含来自 Google Play 商店和 15 个其它应用商店的共 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四分钟。

4.2 评价指标

在深度学习的分类任务中，经常使用以下的评价指标对模型的分类性能进行评估，如：精确度（Precision）、F 值（F1-Score）、准确率（Accuracy）、召回率（Recall），本文也使用这四个指标进行模型效果的评价。这四个指标由真阳性（True Positive，TP）、假阴性（False Negative，FN）、假阳性（False Positive，FP）、真阴性（True Negative，TN）组成。TP 表示的是实际为正样本，模型预测也为正样本的数量；FN 表示的是实际为正样本，模型预测为负样本的数量；FP 表示实际为负样本，模型预测为正样本的数量；TN 表示实际为负样本，模型预测为负样本的数量。

$$precision = \frac{TP}{TP + FP} \tag{0.0}$$

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{0.0}$$

$$recall = \frac{TP}{TP + FN} \tag{0.0}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{0.0}$$

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

在开放世界实验中使用 TPR_{AVE}（所有正确分类的流量与总流量之间的比率）、FPR_{AVE}（所有错误分类的流量与总流量之间的比率）来衡量整体性能，公式如（1.12）和（1.13）

所示

$$TPR_{AVE} = \frac{1}{N}\sum_{i=0}^{N} TP_i * FlN_i \tag{0.0}$$

$$FPR_{AVE} = \frac{1}{N}\sum_{i=0}^{N} FP_i * FlN_i \tag{0.0}$$

其中 N 为监控类别总数，FlN_i 为第 i 个类别的样本总数。这两个指标能反映出模型对监控类和非监控类的识别能力。

4.3 实验结果与分析

4.3.1 实验设置

（1）对比实验选择：

为证明所提出 CB 方法的有效性，选择最经典的机器学习方法 AppScannerp[1]和 FlowPrint[2]，最经典的深度学习方法 App-Net[3]进行对比实验。其中 AppScanner 提取了双向流量的数据包的长度、时间戳、数据包间隔时间，并利用支持向量机和随机森林实现分类。FlowPrint 提取数据包中的明文信息，基于半监督学习的方法对网络流量的目的地进行建模实现流量识别模，同样使用随机森林分类器对流量进行分类。App-Net 该方法提取出 SSL/TLS 握手阶段交换的第一个数据包（Client Hello）的有效负载荷，利用 LSTM 和 CNN 分别学习深度特征。

（2）模型设置：

模型由两层 CNN 层、一层随机失活层、两层 BiLSTM 层、一层注意力层以及一层全连接层组成。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 和 30。训练过程使用 Adamx 作为优化器，学习率设置为 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，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，需要增加的模型的隐藏层就越多。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，所以本文中选择提取 6 个数据包。实际上，前六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。

其次，每个数据包的长度在几十到上千不等，需要归一化所有数据包的长度。然而，如果截取的字节数过短会导致分类器获取的信息不足，不能较好地进行分类；如果截取的字节数较长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了实验，如图 4（b）所示，在进行实验中发现只提取 20 个字节就能获得接近 80%的准确，当提取的字节数达到 80 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，且模型训练时间会增长，由此本文决定选 60 字节作为统一的长度。

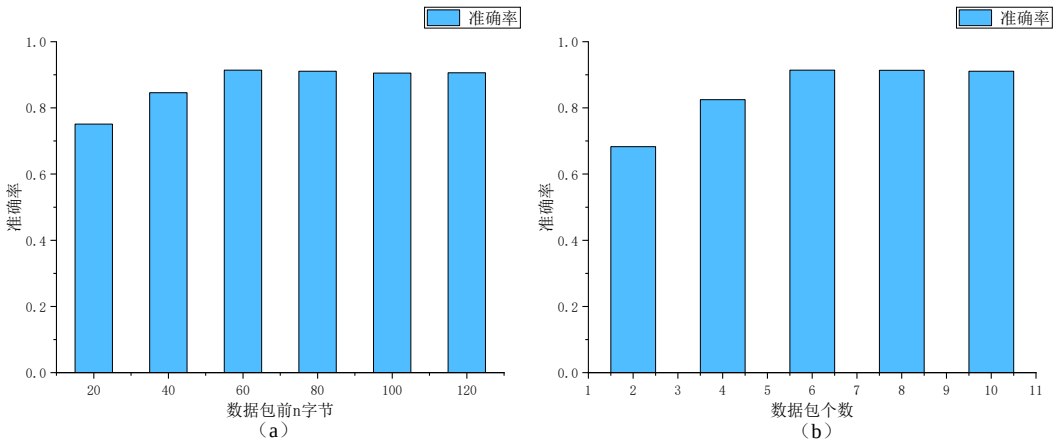

图 4 参数选择实验

后续的所有实验将遵循以下设置：对于每个流量样本，提取 6 个数据包，每个数据包截取其有效负载的前 60 字节，将其转换为 0-255 之间的数字，以进行训练和测试。此外，选取 80%作为训练集，10%作为验证集，10%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的影响，为确定本文提出的方法具有现实意义，在本节中选择了在实验中表现最好的 AppScanner 方法来展示，只使用单向流量特征对传统方法性能造成的影响。

AppScanner 中使用了统计特征来进行分类，使用的分类器是随机森林。AppScanner 中提取出数据包长度、时间，然后分别计算传入数据包、传出数据包和双向数据包的最大值、最小值等来进行流量分类。在进行单向流量实验时，本文过滤掉传出数据包，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。在表 1 中可以看出，当只使用单向流量的特征时，模型的四项指标都有所下降，准确率下降的幅度最大。当使用双向流量特征进行分类时准确率为 89.53%，而只使用单向流量特征进行分类其准确率为 80.70%，下降了 8.83%。这是因为 AppScanner 按照时间将流量划分为突发（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划分并分别计算其统计数据，由于其涉及到流量的交互信息，当只使用单向流量时，AppScanner 的性能会下降。这证明了现有方法在只有单向流量的情况下，性能会大幅度下降。

表 1 不同情况下 AppScanner 性能

输入	精确度（%）	召回率（%）	F 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

表 2 展示了 AppScanner、FlowPrint、App-Net 还有本文提出的方法在 CrossPlatform 上的准确率、召回率、精确度等指标。其中，只有本文的 CB 方法使用了单向流量，其他方法使用均使用双向流量进行实现。这是为了证明本文的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方法相同甚至更好的性能。

从表 2 中的数据中可以看出，CB 只使用单向流量的特征就能够以较高的准确率将网络流量正确分类到相应的应用程序，精准率可达 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%。

表 2 对比实验

数据集	模型	精确度（%）	召回率（%）	F 值（%）	准确率（%）

	AppScanner	91.73	89.53	88.66	89.53
CrossPlatfrom	FlowPrint	80.87	77.66	73.42	77.66
Android	App-Net	66.69	68.54	63.54	68.54
	CB	90.19	91.35	90.66	91.35
	AppScanner	92.23	84.00	83.96	84.00
CrossPlatform	FlowPrint	88.57	89.87	84.49	89.87
IOS	App-Net	70.39	67.69	67.78	67.69
	CB	89.36	90.48	89.41	90.48

图 5 展示了本文及对比实验方法在 CrossPlatform IOS 上的混淆矩阵，其中（a）、（b）、（c）、（d）依次为 AppScanner、FlowPrint、App-Net 和本文方法。从（a）中可以看出 AppAcanner 的总体分类性能较好，但是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。从（b）中可以看出 FlowPrint 方法虽然总体分类性能较高，但是会存在对部分类别准确率很差的情况，如将第 8、24 类的大部分示例都进行了错误的分类，它们的准确率都低于 20%。而 App-Net 的仅为一小部分应用程序提供了较高的准确率。而本文的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。

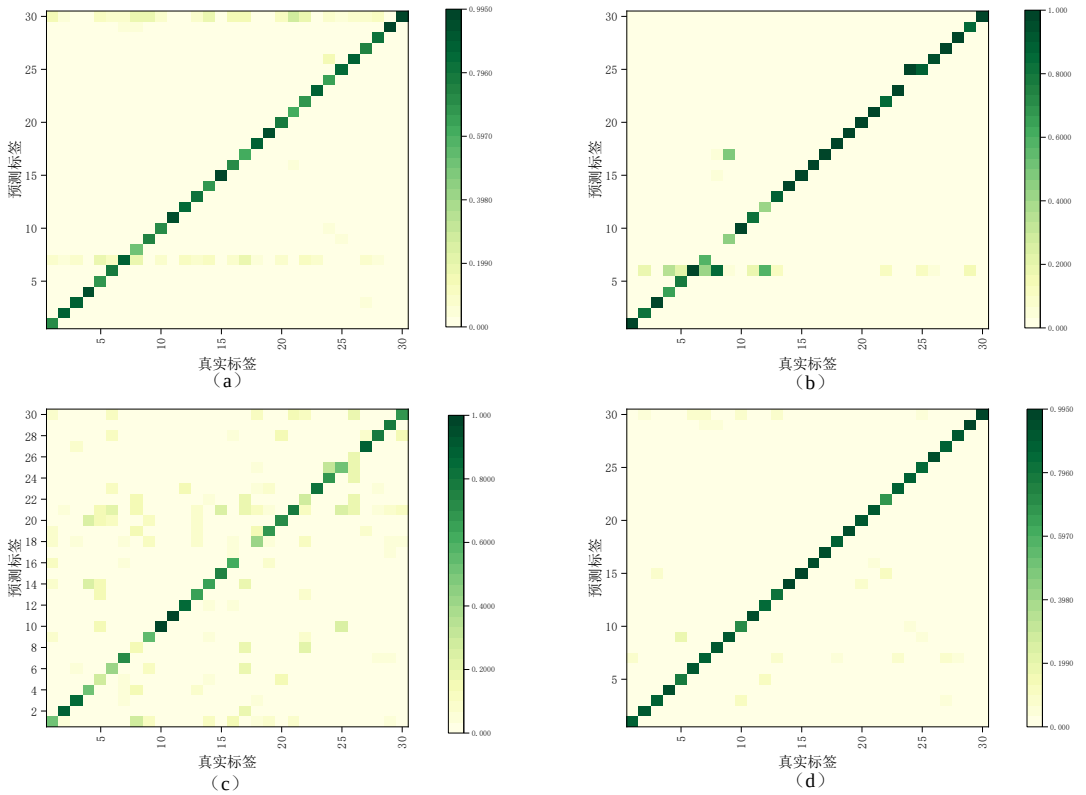

图 5 混淆矩阵

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，监控类别来自 CrossPlatfronm Android 数据集，未监控类别来自 Andribus 数据集，在模型训练过程中将未知类的示例由 2000 增加到 50000（每

个应用一个示例），测试开放世界大小对模型性能的影响。

如图 6 为世界大小对 TPRAVE 的影响，图 7 为世界大小对 FPRAVE 的影响。从图 6 可以看出随着世界大小的增大，所有模型的 TPR_{AVE} 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，下降了 16%。FlowPrint 的下降幅度最小，下降了 10%。

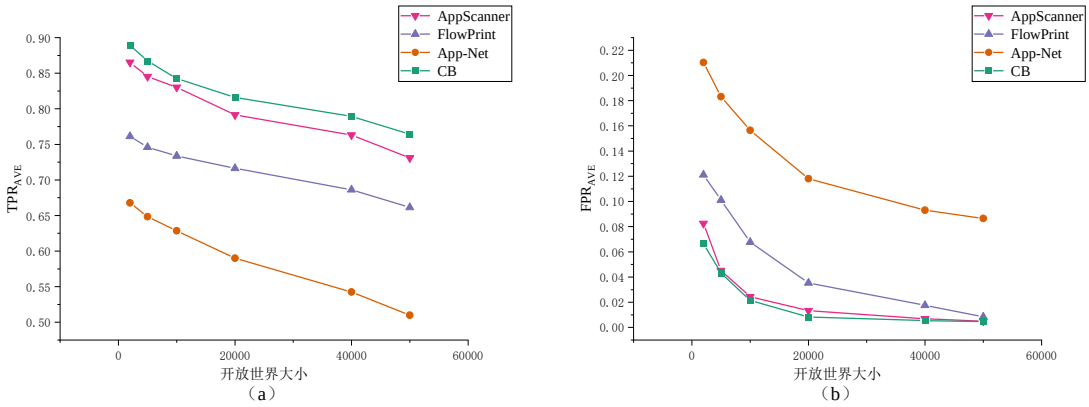

图 6 开放世界大小对模型性能影响

由于监控的数据集和未受监控数据严重不平衡，只使用 TPR 和 FPR 可能会由于基本速率谬误而导致出现错误的解释，所以在本文中还使用精度－召回曲线来评估模型在开放世界场景中的性能。在实验中将未监控的网站数量控制在 20000，如图 7 为精度－召回曲线。本文的方法 CB 优于其他所有方法。在 CrossPlatform Android 数据集中，它对大多数阈值都比较有效。从图中可以看出 AppScanner 方法具有很高的精度，但是召回率的范围比较广，而 FlowPrint 和 App-Net 召回率和精确度都比本文的方法低。

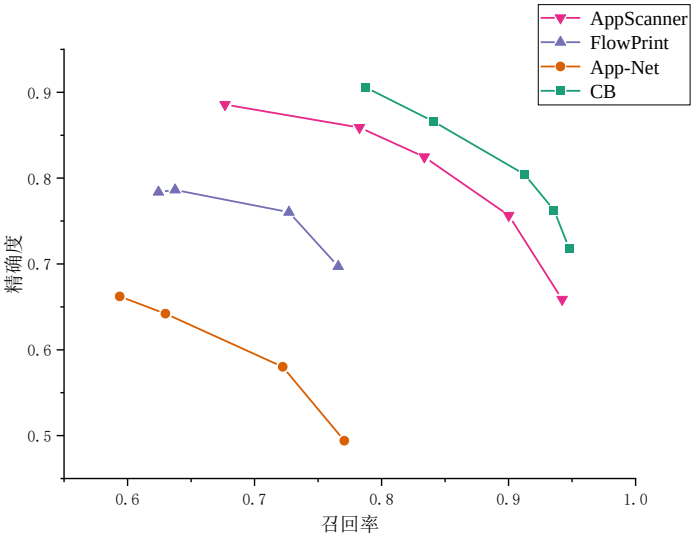

图 7 精度-召回曲线

5 结论

针对远程攻击者只能获取单向流量，导致传统的基于双向流量特征方法准确率下降问题，提出了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、注意力机制进行了结合，使用卷积神经网络提取网络流量的空间特征、使用双向长短期记忆网络来获取流量的时序信息、利用注意力模块进行特征选择。该方法相比于传统方法使用了更少的特征，获得了更高的性能，本文提出的方法在 CrossPlatform Android 数据集上准确率比 AppScanner 方法高了 0.8%，在 CrossPlatform IOS 数据集准确率高了 0.5%。在开放世界场景中，本文提出的方法在开放世界上比传统方法准确率高 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 2020 Annual Network and Distributed System Security Symposium, 2020:1-18.

[3] Wang X, Chen S H, Su J S, et al. App-Net: A hybrid neural network for encrypted mobile traffic classification[C] // Proceedings of the 2020 IEEE International Conference on Computer Communications, 2020:424-429.

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

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

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

[7] Panchenko A, Lanze F, Zinnen A, et al. Website fingerprinting at Internet scale[C] // Proceedings of the 2016 Annual Network and Distributed System Security Symposium, 2016:40-55.

[8] Stöber T, Frank M, Schmitt J, et al. Who do you sync you are? smartphone fingerprinting via application behaviour[C] // Proceedings of the 2013 ACM conference on Security and privacy in wireless and mobile networks, 2013:7–12.

[9] Yang L M, Fu S J, Luo Y C A, et al. Markov probability fingerprints: a method for identifying encrypted video traffic[C] // Proceedings of the 2020 IEEE International Conference on Mobility, Sensing and Networking, 2020:283-290.

[10] Zhuo Z, Zhang Y, Zhang Z-l, et al. Website fingerprinting attack on anonymity networks based on profile hidden Markov model[J] // IEEE Transactions on Information Forensics and Security, 2017, PP:1-1.

[11] Conti M, Mancini L V, Spolaor R, et al. Can't you hear me knocking[C] // Proceedings of the 2015 ACM Conference on Data and Application Security and Privacy, 2015:297-304.

[12] Li J F, Zhou H, Wu S H, et al. FOAP: Fine-grained open-world android app fingerprinting[C] // Proceedings of the 2022 USENIX Security Symposium, 2022:1579-1596.

[13] Zhang J L, Li F H, Ye F, et al. Autonomous unknown-application filtering and labeling for DL-based traffic classifier update[C] // Proceedings of the 2020 IEEE International 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 International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, 2021:1-6.

[15] Telikani A, Gandomi A H, Choo K K R, 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 L T, Xiong G, et al. FS-Net: A how sequence network for encrypted traffic classification[C] // Proceedings of the 2019 IEEE Conference on Computer Communications, 2019:1171-1179.

[17] Yao H P, Liu C, Zhang P Y, 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, Assoc U. k-fingerprinting: A robust scalable website fingerprinting technique[C] // Proceedings of the 2016 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(24):106944.

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

[21] Aceto G, Ciuonzo D, Montieri A, et al. Mobile encrypted traffic classification using deep learning[C] // IEEE Transactions on Network and Service Management, 2019:445-458.
[22] Lin X, Xiong G, Gou G, et al. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification[C] // Proceedings of the 2020 ACM Web Conference, 2022:633–642.
[23] Alan H F, Kaur J. Can android applications be identified using only TCP/IP headers of their launch time traffic?[C] // Proceedings of the 2016 ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2016:61–66.
[24] Lindorfer M, Neugschwandtner M, Weichselbaum L, et al. Apps later: A view on current android malware behaviors[C] // Proceedings of the 2014 International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, 2014:3-17.
[25] Shen M, Gao Z, Zhu L, et al. Efficient fine-grained website fingerprinting via encrypted traffic analysis with deep learning[C] // Proceedings of the 2021 IEEE/ACM International Symposium on Quality of Service. IEEE, 2021: 1-10.