中图分类号: 学校代码: 10055
UDC: 密级: 公开

南開大學

硕 士 学 位 论 文

面向大规模网络的 VoIP 流量实时识别研究

A real-time identification system for VoIP traffic in large-scale networks

论文作者 王荣康 指导教师 张建忠教授
申请学位 工学硕士 培养单位 计算机与控制工程学院
学科专业 计算机科学与技术 研究方向 网络与信息安全
答辩委员会主席 评 阅 人

南开大学研究生院
二〇一九年五月

南开大学学位论文使用授权书

根据《南开大学关于研究生学位论文收藏和利用管理办法》，我校的博士、硕士学位获得者均须向南开大学提交本人的学位论文纸质本及相应电子版。

本人完全了解南开大学有关研究生学位论文收藏和利用的管理规定。南开大学拥有在《著作权法》规定范围内的学位论文使用权，即：(1) 学位获得者必须按规定提交学位论文(包括纸质印刷本及电子版)，学校可以采用影印、缩印或其他复制手段保存研究生学位论文，并编入《南开大学博硕士学位论文全文数据库》；(2) 为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆等场所提供校内师生阅读，在校园网上提供论文目录检索、文摘以及论文全文浏览、下载等免费信息服务；(3) 根据教育部有关规定，南开大学向教育部指定单位提交公开的学位论文；(4) 学位论文作者授权学校向中国科技信息研究所和中国学术期刊 (光盘) 电子出版社提交规定范围的学位论文及其电子版并收入相应学位论文数据库，通过其相关网站对外进行信息服务。同时本人保留在其他媒体发表论文的权利。

非公开学位论文，保密期限内不向外提交和提供服务，解密后提交和服务同公开论文。

论文电子版提交至校图书馆网站：http://202.113.20.161:8001/index.htm。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人同意遵守上述规定。本授权书签署一式两份，由研究生院和图书馆留存。

作者暨授权人签字：_____

20　　年　　月　　日

南开大学研究生学位论文作者信息

论 文 题 目	面向大规模网络的 VoIP 流量实时识别研究				
姓　　名	王荣康	学号	2120160423	答辩日期	
论 文 类 别	博士　学历硕士　硕士专业学位　高校教师　同等学力硕士				
院 / 系 / 所			专　业		
联 系 电 话			Email		
通讯地址 (邮编)：					
备注：					

注:本授权书适用我校授予的所有博士、硕士的学位论文。由作者填写 (一式两份) 签字后交校图书馆，非公开学位论文须附《南开大学研究生申请非公开学位论文审批表》。

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：_____ 年 月 日

非公开学位论文标注说明

根据南开大学有关规定，非公开学位论文须经指导教师同意、作者本人申请和相关部门批准方能标注。未经批准的均为公开学位论文，公开学位论文本说明为空白。

论文题目			
申请密级	限制（2 年） 秘密（10 年） 机密（20 年）		
保密期限	20 年 月 日至 20 年 月 日		
审批表编号		**批准日期**	20 年 月 日

南开大学学位办公室盖章（有效）

限制 2 年（最长 2 年，可少于 2 年）

秘密 10 年（最长 5 年，可少于 5 年）

机密 20 年（最长10年，可少于10年）

摘要

近年来，VoIP 应用凭借其丰富的服务和低廉的价格已经成为了重要的通信服务手段。不幸的是，VoIP 服务为人类带来便利性的同时，也造成了诸多的社会问题。诈骗分子利用 VoIP 所提供的便利的服务实施犯罪活动，对我国诈骗案件的侦破造成了极大的困难。我国反 VoIP 作案面临严峻的局势，为了使 VoIP 进一步服务于人类，对恶意 VoIP 的监管是必要的。

网络流量识别是一个较热门的研究领域，主流的流量分类方法包括基于端口的识别方法，基于字节、字符等的模式识别方法，基于行为的分析方法和基于机器学习的识别方法等。由于 VoIP 应用具有加密和 P2P 特性，以上分类方法大部分很难单独应用于 VoIP 流量识别，目前较为成熟的 VoIP 流量识别方法普遍采用多种方法结合的策略。另一方面，VoIP 流量识别要求较高的实时性，一些基于流特征进行识别的方法不能被应用于实时识别。此外，对 VoIP 流量建立特征集的任务也是极其繁琐的。

本文针对上述问题进行研究，第一，系统地归纳了各类 VoIP 识别方法，包括非实时的识别和实时的识别方法，并且从不同的识别层面上作出比较；第二，本文围绕 VoIP 应用类型采用 CLNN（卷积神经网络，长短期记忆网络）模型提取特征，提取特征的过程不但不需要人为参与，并且提取的特征具有更高的精确度。第三，基于本文提出的方法，我们设计并实现了一个 VoIP 实时识别系统，该系统采用 Apahce Storm 作为流处理计算引擎，支持在大规模的网络中采集并识别 VoIP 流量。实验结果表明，该系统可以实时的准确识别 VoIP 流量。

关键词： VoIP；流量识别；实时；卷积神经网络；长短期记忆网络

Abstract

Recently, owing to the service and price advantage, VoIP has become an important communication technology. Unfortunately, while providing convenience services, it also causes some social tragedies. Swindlers handle VoIP services provided by VoIP to carry out criminal activities, which causes great difficulties in solving fraud cases. We are facing severe situation to against VoIP frauds, in order to make VoIP applications serve humans better, it is important to keep VoIP applications under supervision.

Traffic identification has been an active research topic in the past decade, several methods have been given, which can be generalized as port-based, pattern recognition, behavioral analysis, machine learning methods. Due to the encryption and P2P characteristics of VoIP applications, the above identification methods are difficult to be applied to VoIP traffic identification alone, researchers generally adopt a combination of several methods. On the other hand, VoIP identification has strict real-time requirements, the methods based on flow features are not applicable to real-time identification. It is troublesome to extract features for VoIP traffic.

Firstly, various VoIP identification methods are summarized and we compare them from different identification levels; Secondly, we adopt CNN(Convolutional Neural Networks) to extract features for accurate VoIP application identification. These extracted features are not only more reliable than the features extracted by humans, but also greatly improve the identification efficiency; Thirdly, we design a real-time identification system with Apache Storm, it can capture VoIP traffic in a large-scale network and identify their application types with the features we trained. The evaluation results verify that our system can identify VoIP traffic timely and accurately.

Key Words: VoIP; application identification; real-time; CNN; LSTM

目录

第一章 引言

第一节 研究背景

近年来，VoIP（Voice Over Internet Protocol）技术发展迅速。VoIP 是一种语音通信技术，可提供网际协议作为媒介进行语音传输的环境。包括智能手机、计算机和 VoIP 电话在内的各类终端可在蜂窝网络、Wi-Fi 和以太网等各种网络环境中进行语音通信。与传统的公共交换电话网络（Public Switched Telephone Network，PSTN）相比，VoIP 服务具有低通话成本、低建设成本、高扩展性、服务丰富等优势，因此，VoIP 很快获取了人们的青睐。此外，公共交换电话网络可以通过 FXO 语音网关接入 VoIP 网络，从而实现两种网络之间的上下车应用，各种各样的 VoIP 应用也应运而生。

VoIP 服务在创造大量社会价值的同时，也造成了诸多的社会问题。来自通信欺诈管制协会的报告显示，电信欺诈所造成的损失以每年 29% 的速度增长。其发布的一份报告显示，2017 年仅与 VoIP PBX（专用交换机）相关的黑客攻击行为所造成的价值损失就高达 19.4 亿美元[1]。据估算，每年因 VoIP 欺诈事件所造成的全球总损失达 300-500 亿美元。在国内，VoIP 欺诈案件普遍表现为金融诈骗，诈骗分子利用 VoIP 所提供的改号服务冒充证券银行相关工作人员实时金融诈骗。诈骗分子通过将 VoIP 呼叫系统架设于境外代理服务器，以匿名接入的方式对中国境内电话呼叫实施诈骗。VoIP 所提供的服务允许诈骗分子修改主叫号码，操作路由跳板，对我国诈骗案件的侦破造成了极大的困难。由中国信息通信研究院发布的《信息通信行业防范打击通讯信息诈骗白皮书》显示，2016 年全国共立通讯信息诈骗案件 63 万起，上半年造成经济损失 80.4 亿元[2]。《2017 年中国反通讯网络诈骗报告》显示，每 16 个用户中就超过一个收到了诈骗电话[3]。另一方面，VoIP 因其加密性而难以追踪经常被恐怖组织所利用。早在 2005 年，时代周刊就指出 Skype 为大规模技术盲点。2008 年，英国每日邮报报道，阿富汗境内的塔利班恐怖武装使用 Skype 逃避军情六处的侦查[4]。2012 年，印度时报报道，伊斯兰恐怖组织虔诚军已经开发了自己的 VoIP 网络 Ibotel 用以躲避侦查[5]。

为应对以上问题，各国都对 VoIP 技术有不同程度的监管。在我国，对 IP 业务牌照审核有严格的要求，实行适度开放 VoIP 业务的政策。这也意味着未经授权的 VoIP 业务是非法的，并且所有合法的 VoIP 业务也必须处于监管之下以应对任何程度的紧急情况。但是随着技术的进步，越来越多的 VoIP 应用开始采用加密和对等网络的方式，甚至有一些大型厂商采用自己的 VoIP 协议，这些都使得 VoIP 识别任务愈发艰难。大数据时代的到来为 VoIP 识别提供了新的思路，VoIP 数据规模的增加使得越来越多的信息可供挖掘和分析，通过所挖掘的信息识别 VoIP 流量。本文将结合深度学习和大数据处理技术，对大规模网络中的 VoIP 流量进行实时识别，以支撑网络管理员对非法 VoIP 流量进行监管。

第二节 研究现状

在过去的十几年内，网络流量分类一直是一个研究热点。同时因为 VoIP 技术的发展，针对 VoIP 的网络流量分类吸引了很多研究者，许多有价值的工作也随之产生。

同其他流行的应用一样，早期的 VoIP 应用也使用由 IANA（The Internet Assigned Numbers Authority，互联网数字分配机构）所指定的端口号。但是随着 P2P 和 NAT 技术的发展，许多厂商开始在自己的 VoIP 软件中使用这两项技术，VoIP 应用不再局限于使用固定端口号，而是采取动态端口映射的方案。这就使得传统的基于端口号的识别方法不再有效。为了克服这种困难，一些研究人员[6,7] 通过识别某些特定协议如 SIP，RTP 协议所携带的特征来进行 VoIP 流量识别。一些研究[8] 则从应用级别的角度对 VoIP 应用采取 DPI（Deep Packet Inspection，深度包检测）的方式提取相关特征进行 VoIP 识别。这些识别方法仅仅可以识别分析过的 VoIP 应用，不可能做到对更多 VoIP 应用的识别，并且这种方式对于 Skype 这种私有协议是无效的。其次，这些方法不能识别采用了加密技术的 VoIP 应用。

基于统计分析的方法是另一种经常被用于检测 VoIP 流量的方法。诸如数据流的包长度、包到达间隔、持续时间和分组交换率等统计特征被分析用于检测 VoIP 流量[9]。一些研究方法[10] 结合流的统计特征分析和主机行为估计方法，通过设置参数 D 用于过滤非 VoIP 的流量，之后通过 EL（large inter-packet time，最大包到达间隔）和 ES（small inter-packet time，最小包到达间隔）计算出的参数 R 再次准确的识别 VoIP 流量。与此同时，一些最新的研究[11,12] 仍然在

表 1.1 IANA 指定的 VoIP 默认端口

VoIP 协议	端口	传输层协议
SIP	5060	TCP/UDP
SIPS	5061	TCP/UDP
SIP Audio	2326-2485	UDP
H.323	1300, 1718-1720	TCP/UDP
H.323 Audio	2326-2485	UDP
MGCP	2427	TCP/UDP
MEGACO/H.248	2944, 2945	TCP/UDP

采用基于规则和统计分析的方法检测 VoIP 流量。这些方法需要根据对应的规则设置很多的阈值参数，越来越多的 VoIP 应用会导致这些阈值参数有很大的出入。随着机器学习技术的发展，基于统计分析的方法也被赋予了新的生命力。一些研究[13] 提出了使用机器学习的方法识别加密的 VoIP 流量。他们建立了基于流的特征集并且使用了 3 中机器学习的方法来训练数据集，3 中机器学习方法有不错的识别率，其 C5.0 获得了最优的效果。机器学习的方式减轻了研究人员在设定阈值方面的负担，但是终归还是基于统计分析的方法，并且对于更新速度越来越快的 VoIP 应用来说，每种 VoIP 应用都可能在短时间内更新自己的算法，这样就需要认为的重新进行统计分析来适应 VoIP 的更新。尤其中作者提出的方法是基于整个 VoIP 数据流训练数据集，训练所得的分类器并不适用于实时的 VoIP 流量识别。从流量识别级别的角度来说，上述方法大多应用于检测 VoIP 流量，也就是说，仅仅针对 VoIP 流量与非 VoIP 流量进行识别，不能从应用级别的角度进行识别。

VoIP 流量识别最大的难点是要求更高的实时性，一些研究也从实时识别的角度进行 VoIP 识别的研究工作。为了实时的检测 VoIP 流量，数据包长度在 60-150 字节内的 UDP 数据包将会被判断成 VoIP 数据包[14]。在这篇文章中，作者仅仅致力于提高 VoIP 的服务质量，该方法不追求较高的识别率所以不能被应用于准确的 VoIP 实时识别。一种基于 VoIP 会话熵变和解编码器类型的方法被提出[15]，文章考虑了几种流行的 VoIP 应用并研究了它们的解编码器类型，通过解编码器类型可以初步的识别一些 VoIP 应用。为了更加准确的识别 VoIP 应用，他们研究了 N 个连续数据包之间的长度关系，并且用熵来衡量它们之间的异质性，通过解编码器类型和熵的异质性来识别 VoIP 流量。从实时识别的角度，他们采取了具有 N 个长度的滑窗计算这 N 个数据包的异质性，通过识

别 N 个数据包就可以识别整个流的 VoIP 应用类型。相同的实时识别策略也被文章[16] 采纳，作者通过提取整个双向流的前几个数据包的 PSD（Packet Size Distribution，包长分布）用于识别 VoIP 应用，实验结果显示这种方式可以在双向流建立之后立即识别出 VoIP 应用类型。目前为止所有的 VoIP 实时识别方案都是基于识别子流的方式从而得知整个数据流的应用类型，这种方式的识别精度与所采用的滑窗大小有直接关系，采用的滑窗太小会导致识别精度太低，滑窗太大会降低实时性。尤其因为识别所采用的特征太少，文章[15] 采用两个特征识别数据流，文章[16] 采用一个特征识别数据流，因此识别精度不是很高。

第三节 本文主要工作

本文主要针对 VoIP 流量识别两个难点展开研究，研究的识别方法通用与加密和非加密 VoIP 应用。两个研究难点，其一是识别精度的问题，提高识别精度的根本在于寻找更加精确的特征集；其二是保证识别的实时性，提高实时性要求减少识别所需的数据包长度。本文研究了现有的 VoIP 流量识别的方法，并对卷积神经网络和长短期记忆网络进行了阐述。利用卷积神经网络在频域方面的建模能力和长短期记忆网络在时域方面的建模能力提取精度更高的特征集。本文针对多种 VoIP 应用建立数据集用于有监督的训练。综合研究成果，本文将训练所得分类器应用于实时识别系统，该系统可支撑在大规模网络中实时识别 VoIP 应用。

1）VoIP 流量识别综述。本文对几种常用的流量识别方法进行了归纳和总结，并将其与 VoIP 流量识别做了比较，总结了 VoIP 流量识别的难点。对现有的几种 VoIP 流量识别方法进行了详细介绍，分析了其优缺点和适用性。

2）基于 CLNN 模型的 VoIP 识别方法。本文研究了 CNN 和 LSTM 的适用性，结合了卷积神经网络在频域方面和长短期记忆网络在时域方面的建模能力，通过研究多种网络模型设计了适用于 VoIP 流量的网络模型。该模型可以提取最优的特征集，避免了人工提取特征集的繁杂工作，并且具有良好的识别性能。

3）VoIP 实时识别系统的实现。对 CLNN 模型训练所的的特征集，本文使用了 4 中机器学习方法重新训练分类器，这些分类器较 CLNN 模型具有更好的实时性。为处置大规模网络中的流量，本文采用了 Apache Storm 作为流式计算引擎，拓扑结构中包含多种分类器 Bolt，网络管理员可以根据紧急程度采用不同的识别结果。

第四节 本文的组织结构

本文的结构安排如下：

第一章为引言，主要介绍了 VoIP 流量识别的研究背景和研究现状，阐述了 VoIP 流量识别的必要性和目前的识别方法所存在的不足，最后介绍了本文的主要研究内容。

第二章为相关研究，介绍了 VoIP 相关技术并比较了传统的流量识别与 VoIP 流量识别的异同，详细的分析了现有研究方法的适用范围和优缺点。最后，介绍了本文所采用的卷积神经网络和长短期记忆网络的基本原理。

第三章为基于 CLNN 网络的特征提取相关研究。该章首先分析了 VoIP 流量所携带的可支持识别的特征，其次介绍了本文所采用的用以支撑实时识别的方案。最后介绍了如何构建 VoIP 流量数据集并使用 CLNN 模型训练数据集提取特征集，详细介绍了 CLNN 模型的设计。

第四章为 VoIP 流量实时识别系统的实现。该章介绍了实时系统的总体框架。之后从离线训练和在线识别两个方面介绍了系统的实现。

第五章为实现结果以及性能评估。评估了 CLNN 模型提取的特征集，并对 4 种机器学习训练所得的分类器进行了评估。

第六章总结了全文的工作，分析了本文研究工作的不足之处以及进一步的研究方向。

第二章 相关研究

VoIP 流量识别是从各种网络协议流量中识别出 VoIP 相关流量，并对其进一步分类的过程。VoIP 流量识别主要用于保障网络安全，可以依 VoIP 应用类型对流量作出不同的处置。本章对多种传统流量识别方法进行了总结分析，并比较了其与 VoIP 流量识别的异同，介绍了 VoIP 识别的难点。最后详细阐述了卷积神经网络与长短期记忆网络的基本原理。

第一节 VoIP 简介

VoIP（Voice over Internet Protocol，基于 IP 的语音传输协议）是一种网络通话技术，经由网际协议来达成语音传输，也就是在互联网上进行语音通信。早期的 VoIP 技术是在个人电脑上通过软件的方式实现的，当时的 VoIP 软件只能支持电脑与电脑之间进行通信。2000 年，VoIP 技术在 IP PBX 的加持下，开始向电信领域渗透。传统的公共交换电话网络可以通过 IP PBX 接入 FXO 网关连接到互联网，从而实现传统公共交换电话与网络电话之间的通信，这也导致了 VoIP 技术的飞速发展。VoIP 技术发展至今，作为宽带电话已经从语音质量、价格等各个方面超越了传统的固定电话。

VoIP 协议包括用于信令控制和媒体传输两方面的相关协议。主流的 VoIP 协议包括 H.323，SIP，MGCP，MEGACO 和 RTP/RTCP 等。其中 H.323 和 SIP 作为被广泛使用的信令控制协议主要用于呼叫建立、呼叫保持和呼叫终止，它们由不同的标准化组织制定，SIP 主要用于分组网络，H.323 同时用于分组网络和电路网络。MGCP 和 MEGACO 协议在概念和结构上有很多重叠，两种协议都可以说是对 H.323 协议簇的进一步发展，其中 MGCP 定义了网关分离的思想，大幅度提高了媒体网关的可用性；MEGACO 即 H.248 现属于 H.323 协议簇的一部分，建立在 MGCP 基础之上，其功能强大、扩展性良好，允许在呼叫控制层建立多个分区网关，已开始逐步取代 MGCP。实时传输协议即 RTP 为语音提供端到端的传输服务，其包括 RTP 和 RTCP 两个子协议，是最主流的 VoIP 媒体传输协议。因为本文的 VoIP 识别方法主要基于对 RTP/RTCP 流量进行识别，本节将对信令控制协议和实时传输协议作简单。

2.1.1 信令控制协议

本节将对 H.323 协议簇和 SIP 两种信令控制协议作简单介绍，并分析信令控制协议在 VoIP 流量识别方面的优缺点，并解释为什么本文没有采用分析信令控制协议的方案识别 VoIP 流量。

2.1.1.1 H.323

H.323 协议簇是于 1996 年由 ITU-T 制定针对多媒体通信的协议，它包括呼叫控制、媒体编码、管理控制、网络安全等一系列协议，是目前使用最为广泛的 VoIP 通信协议。本节我们主要介绍 H.323 协议簇的核心内容，包括 H.323 结构相关、H.225 和 H.245 协议。

H.323 结构中定义了多个提供多媒体通信的网络元素，这些元素包括终端、多点控制单元、网关和网守等。H.323 系统中所定义的终端包括支持语音视频编解码、远程信息处理和用户界面的设备，需要支持 H.323 协议簇所支持的一系列协议堆栈；多点控制单元由多点控制器和多点处理器组成，用于管理多点回忆，可以进行音频混合、视频切换等操作；网关是用于分组交换网络和其他电路交换网络，如 PSTN 和 ISDN，进行数据转换的元素；网守在 H.323 网络中是可选的，用于提供包括最重要的地址解析在内的多种服务，其主要通过 H.225 中的 RAS 协议实现与网关的信令传输。

H.225 协议主要包括两部分的内容，RAS 和 Q.931。RAS 协议是用于网关和网守之间的通信协议，其独立于呼叫建立和媒体传输信道，RAS 一般使用 1718 和 1719 端口进行信令传输。Q.931 协议用于呼叫的建立和拆除，它使用 TCP 协议进行传输，一般使用端口 1720 建立呼叫控制信道。H.225 协议中固定端口的使用是很多研究人员分析 H.323 协议进行 VoIP 流量识别的基础。

H.245 协议用于建立音视频和数据传输所用的媒体传输信道，即为 VoIP 通话建立 RTP/RTCP 传输信道。它包括请求、响应、命令和指示四种类型的消息，用于协商信道使用率和功能。

2.1.1.2 SIP

SIP 被设计用于为 VoIP 提供一系列扩展集调用处理功能，通过该扩展集，可以实现 VoIP 拨号、振铃、回铃和忙音等和传统的电话网相同的功能。SIP 的设计简单、灵活、扩展性高，并且 SIP 最初是由 IETF 组织进行标准化和管理，因此它吸引了众多植根于 IP 标准的团体。

SIP 从设计上采用了一些与 HTTP 相同的原则，SIP 报文是人类可读的，同样采取了请求和应答的基本流程。SIP 借助了很多 HTTP 状态码来响应请求，指示呼叫状态。SIP 网络中包括多种逻辑实体，包括用户代理、代理服务器、注册服务器和重定向服务器等。其中用户代理和代理服务器用于发送请求和请求转发，注册服务器和重定向服务器用于位置注册和位置查询。图片2.1展示了 SIP 会话的基本呼叫流程，通话邀请请求会经过代理服务器进行转发，呼叫终止请求由用户代理之间直接发送。

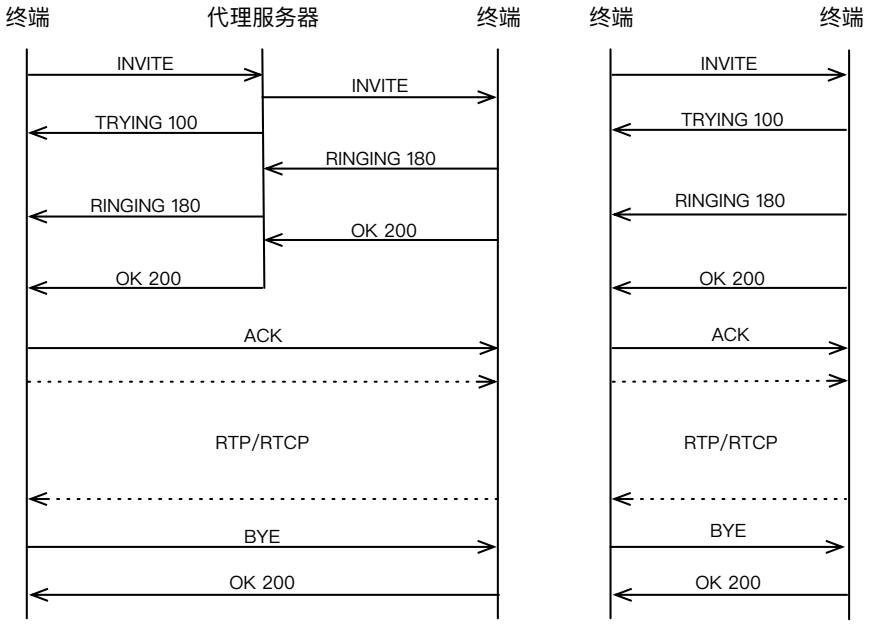

图 2.1 SIP 会话连接建立基本流程

2.1.1.3 信令控制协议之于 VoIP 流量识别

在 VoIP 发展的初期，许多研究人员通过研究信令控制协议进行 VoIP 流量识别。通过研究信令控制协议进行 VoIP 流量识别的方案仍然具有很大的价值，技术方面也具有很大的可行性。因为信令控制阶段位于媒体传输通道建立之前，所以这种方案比研究实时传输协议具有更高的实时性和完善性。

由于信令控制协议的种类纷繁复杂，研究人员都针对某一特定信令控制协议进行分析。比如对 SIP 进行深度包检测的研究，对 H.323 协议的端口进行分析，对私有 VoIP 协议 Skype 协议的握手规则进行分析。因此，对信令控制协议的研究不利于设计通用的 VoIP 流量识别系统。

H.323 协议簇作为最复杂的开源 VoIP 通信协议，除上述介绍的几个核心协议之外，还包括 H.235 在内的安全加密协议。多样的 H.323 协议簇和加密协议的存在会对我们分析 H.323 流量造成很大的障碍。另外，如今的 H.323 协议簇可以采用 SSL 和 TLS 等安全传输层协议进一步加密，从而使得 H.323 流量的分析更加困难。SIP 最初作为最简单的信令控制协议在 VoIP 通信中获得了广泛的应用。但是随之 SIP 技术的发展，其已经演变的和 H.323 一样复杂了，分析 SIP 请求的行为也变的越来越困难。从 SIP 加密的角度，SIPS 为 SIP 使用 TLS 提供传输安全所制定的协议，已经足够成熟被应用到各种 VoIP 应用。协议逐渐变得复杂而灵活以及加密技术的成熟使得对信令控制协议的研究愈发困难。

各种 NAT（Network Address Translation，网络地址转换）技术和 NAT 穿透技术的发展也对信令控制协议的分析造成了一定的难度。SIP 的 NAT 穿透技术包括 ALG、MidCom、STUN、SBC 和 report 等，其中以架设 SBC（Session Border Controller，会话边界控制器）最为常用。该技术成功解决了媒体传输通道在 NAT 技术中采用随机端口的问题，图2.2展示了 SBC 的工作过程，其中 NAT 服务器和 SIP 服务器只进行转发，不会修改 SIP 数据包内容，而 SBC 会修改 SIP 数据包内容并保存各地址和端口的映射表关系。

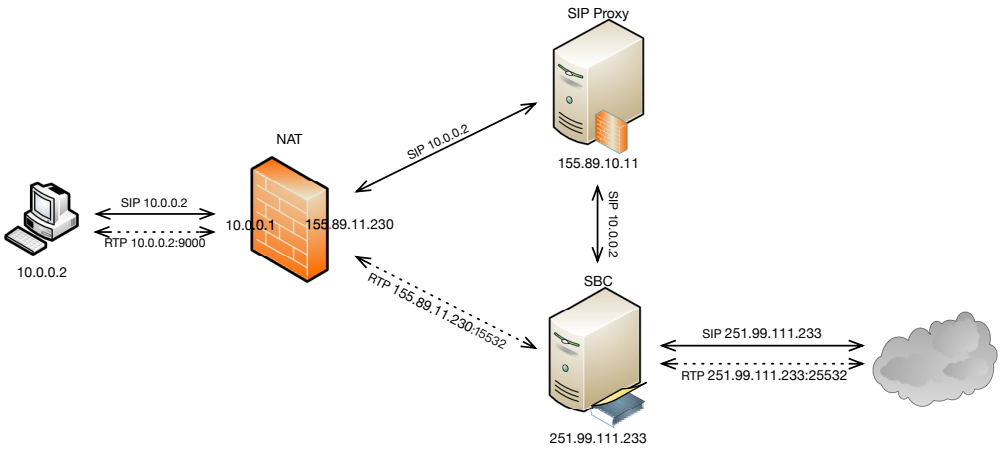

图 2.2 SBC 在 SIP 穿透过程中的工作图示

各种软交换技术即 IP PBX 的发展，将传统的交换设备部件化，加之 MGCP 和 MEGACO 协议的不断完善，成功达到了呼叫控制与媒体分离的效果。这种技术在追求高性能的同时，也成功解决了授权问题、NAT 穿透等诸多问题。图片??展示软交换技术实现呼叫控制与媒体传输分离的基本原理。这使得研究人

员即使成功的分析了信令控制协议，仍需进一步分析软交换过程确定对应的媒体传输链路。

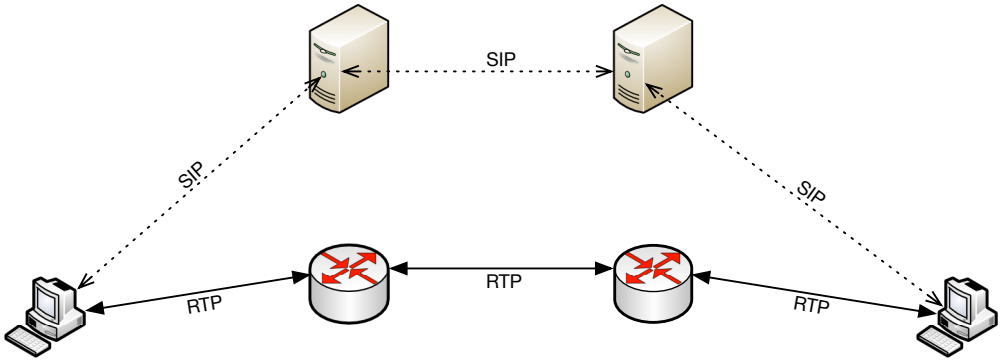

图 2.3 软交换环境中的呼叫控制与媒体传输分离

本文致力于找到一种方法可以应用于通用的 VoIP 流量识别。综上所述，由于信令控制协议的复杂、加密等的特性，对于信令控制协议的研究不适用于通用的 VoIP 流量识别方法。同时也由于 NAT 穿透技术和软交换技术的发展，造成了对信令控制协议分析的难度进一步加大，所以本文不采用信令控制协议产生的流量进行分析的方法，而是致力于使用广泛被 VoIP 应用采用的实时传输协议进行 VoIP 流量识别。

2.1.2 实时传输协议

实时传输协议（Real-time Tarnsport Protocol，RTP）普遍用于在 IP 网络中传输音频或视频数据。通常实时传输协议与实时传输控制协议（RTP Control Protocol，RTCP）结合使用，被广泛的应用于 VoIP 应用中。另外，还有用于控制流媒体服务器的实时流媒体协议（Real Time Streaming Protocol，RTSP）也经常与 RTP 和 RTCP 协议结合使用，由于 RTSP 本身不对流数据进行传输，主要用于控制流数据的传输，本文对 RTSP 不作介绍。RTP 和 RTCP 通常采用 UDP 进行装载，提供端到端的实时性传输。

RTP 用于装载音频和视频数据，包含了装载数据的标识符、负载类型、序列计数和时间戳等信息。图2.3展示了 RTP 的报头格式。我们对其中带有明显识别特征的 4 个占位符做简单介绍。其中 M 作为标记位，占 1 个位，对于音频负载，其标记会话的开始，对于视频负载，其标记一帧的结束。PT 位标记有效负载类型，其用于通知接收该数据包的终端负载所使用的解编码器类型。如果当

前终端在某种条件下需要对该数据包所包含的有效负载进行重新解编码的工作，可以按照 PT 位选择解编码器。由于音频负载和视频负载使用不同的解编码器，根据该位可以很容易识别出当前负载属于音频数据还是视频数据。序列号占 16 位，用于标识发送者所发送的 RTP 报文的序列号。时间戳占 32 位，其表示对当前数据包打包的时间。序列号和时间戳结合可以提取 VoIP 应用在时序方面的特征。

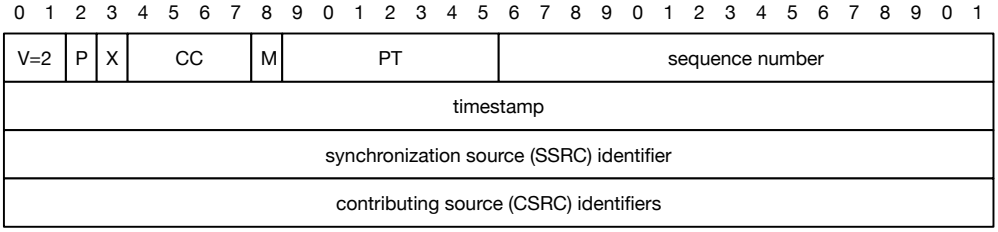

图 2.4 RTP 报头格式

RTCP 作为 RTP 的控制协议其本身并不传输数据，通常与 RTP 一起协作将多媒体数据打包和发送。RTCP 为 RTP 提供质量保证服务，通过周期性的向参与者发送统计信息提供媒体分发中的服务质量的反馈信息，统计信息包括传输字节数、传输分组数、丢包数、抖动和单双向抖动延时等等。RTCP 数据包具有多种不同的类型，包括 RR（接收者报告包）、SR（源报告包）、SEDS（源描述包）、BYE（离开申明）和 APP（特殊应用包）五类，分别用于传送不同的信息。RTCP 通常使用 RTP 所采用的偶数端口号的下一个奇数端口号。

通过分析实时传输协议进行 VoIP 流量识别在识别实时性上有一定的弊端，不能像分析信令控制协议那样在建立媒体传输通道之前就可以知道通话双方即将采用的地址及端口号，并对其作出相应的措施。分析实时传输协议也有其优势，VoIP 应用普遍采用实时传输协议，我们可以根据这种特点设计一套通用的 VoIP 识别方法。另外，VoIP 应用在媒体传输阶段所产生的流量要远远多于信令控制阶段产生的流量，这一条件决定了媒体传输阶段产生的流量要有更强的识别特征。虽然这种方式在实时性上有一定的不足，但是如果在媒体传输通道建立之后的几秒钟甚至几毫秒中对其进行识别，那么在实时性方面的牺牲也是微小的。

第二节 传统流量识别与 VoIP 流量识别方法

第三节 神经网络

为了尽最大可能的提取 VoIP 语音流所携带的可识别特征，本文采用了两种神经网络结构构件深度神经网络模型 CLNN 用于提取特征。长短期记忆网络用于提取语音流在时序方面的特征，卷积神经网络用于提取语音流在频域方面的特征。本节将对两种神经网络结构展开进行介绍。

2.3.1 长短期记忆网络

长短期记忆网络（LSTM）是一种用于提取时间序列相关关系的循环神经网络，适于处理时间序列中间隔较长的重要事件。LSTM 现已被广泛的应用于各种领域，被用于执行图像识别、语音识别和自然语言处理等多种任务。本节将对 LSTM 从发展过程和基本原理两个方面进行介绍。

2.3.1.1 发展过程

循环神经网络（RNN）在训练过程中会出现梯度消失和梯度爆炸的现象，为了解决这一问题，S Hochreiter 和 J Schmidhuber 于 1997 年提出了 LSTM 模型[17]。LSTM 引入了 CEC(Constant Error Carrousel) 单元用于解决梯度消失和爆炸的问题，同时引入了输入输出门用于选择性的忘记和屏蔽某些输入输出，“门”的概念成功的解决了输入输出在更新权重上具有冲突的问题。以上最原始的 LSTM 模型具有记忆长时序列特征的特性，但是其在记忆过长的时间序列特征时，会出现 LSTM 细胞状态过于饱和的现象，即细胞状态对当前输入过于依赖导致失去了泛化的能力。因此，FA Gers 等人在 1999 年针对该问题提出了遗忘门的概念[18]。遗忘门可以使 LSTM 网络选择合适的位置进行记忆的重置，通过引入变量替换了 CEC 单元中的常量，解决了 LSTM 网络对于长时序列过于饱和的问题。FA Gers 在 2001 年又针对 LSTM 细胞遗忘过多历史信息的问题引入了窥视孔连接（Peephole Connections）的概念[19]。这种连接是 CEC 单元与输入门、输出门和输出门之间的关系是双向的，增强了 LSTM 模型在时序上建模的能力。在 LSTM 不断发展的过程中，针对不同的训练任务出现了不同的版本，K Cho 等人于 2014 年提出的 GRU（Gated Recurrent Unit）是其中一个重要的版本，其在原始 LSTM 基础上增加了更新门，从而减小了梯度弥散的风险。如今的 LSTM 广泛的被谷歌、微软等公司采用执行各种任务，LSTM 的发展无疑是人工智能事业发展的一大步。

2.3.1.2 基本原理

LSTM 的实际结构是由单个细胞循环所组成的，细胞通过挖掘上一个时序上的输出 h_{t-1} 和当前时序上的输入 x_t 之间的关系将细胞状态由 C_{t-1} 更新为 C_t。细胞内部由三个门控结构组成，分别为输入门、输出门和遗忘门。我们将介绍这三个门控结构的基本原理。

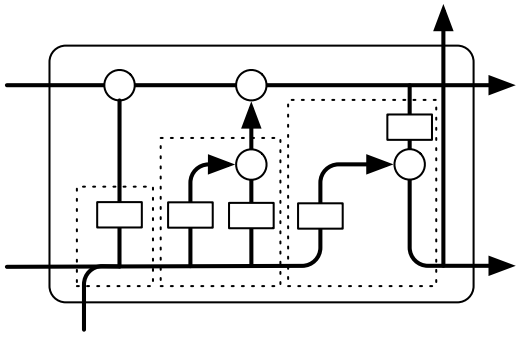

图 2.5 LSTM 细胞结构

图2.4中虚线部分①为遗忘门部分，遗忘门通过计算上一序列的隐藏状态 h_{t-1} 和当前序列的输入数据 x_t 之间的关系获得对上一个细胞状态 C_{t-1} 的遗忘概率 f_t。数学函数表示为：

$$f_t = \delta(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.1}$$

图2.4中虚线部分②为输入门部分，输入门部分包含两个步骤，第一步是使用 Sigmoid 激活函数计算上一序列的隐藏状态 h_{t-1} 和当前序列的输入数据 x_t 获得本序列要更新的信息 i_t，第二步是使用 tanh 激活函数计算出候选的细胞状态 $\tilde{C}_t$。这两部分所得结果相乘即为当前序列对细胞状态更新的贡献值，再加上经过遗忘门遗忘后的细胞状态就可以获得当前序列的细胞状态 C_t。数学函数表示为：

$$i_t = \delta(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{2.4}$$

图2.4中虚线部分③为输出门部分，输出门要决定在当前序列上要输出的隐藏层信息 h_t。输出门仍然是通过使用一个 Sigmoid 激活函数计算上一序列的隐藏状态 h_{t-1} 和当前序列的输入数据 x_t 获得基于本序列数据的输出状态 o_t，之后通过 tanh 激活函数对当前的细胞状态 C_t 处理后与输出状态 o_t 相乘，从而得到基于当前细胞状态的隐藏层信息。数学函数表示为：

$$o_t = \delta(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.5}$$

$$h_t = o_t \times tanh(C_t) \tag{2.6}$$

经过以上三层门后，所得到的当前细胞状态 C_t 和隐藏层信息 h_t 将被用于计算下一个时序上的数据 x_{t+1}，通过不断的更新细胞状态，细胞将会保存每个时序上最重要的信息，从而提取出时序序列在时间维度上的特征。

2.3.2 卷积神经网络

卷积神经网络是一种典型的前馈神经网络，其神经元结构可以覆盖一定范围内的数据单元，有很强的在频域方面的建模能力。卷积神经网络对于二维数据有优秀的处理能力，它的网络结构所依靠的参数量也少于其他类型的网络，其被广泛的应用于视频分析、图像识别等领域。本节将对卷积神经网络的发展过程和基本原理进行介绍。

2.3.2.1 发展过程

早在 1943 年，WS McCulloch 等人就将生物神经系统归纳为 M-P 神经元模型，他们的论文被认为是神经网络的开山之作[20]。K Fukushima 等人在 1984 年提出了基于感受野的神经认知机，这是生物感受野的概念首次被应用于人工神经网络领域[21]。由于早期的神经网络在各个方面存在一定的局限性，神经网络的发展一直处于停滞阶段。1986 年，DE Rumelhart 以及 GE Hinton 等人提出了著名的反向传播算法[22]。在该算法受到重视的同时，神经网络的发展也迎来了第二次兴起。90 年代，Y LeCun 等人使用反向传播算法进行了手写数字的识别研究[lecun1990handwritten]。随后，他们又设计了一种多层的神经网络结构，取名为 LeNet-5[23]。LeNet 被成功应用于手写数字的分类工作，确立了 CNN 网络的现代结构。此时的神经网络仍受限于计算机的计算能力，在很多复杂问题上的处理结果并不理想。2012 年，A Krizhevsky 等人提出了一个经典卷积神经网

络结构，命名为 AlexNet[24]。它在 2012 年举办的 ImageNet 竞赛中获得了冠军，成功的将图片分类问题的 Top5 错误率降低至 15%。AlexNet 在图像识别领域的成功，掀起了卷积神经网络研究的热潮。随后，K Simonyan，C Szegedy 和 K He 等人分别设计出了 VGG[25]，GoogleNet[26] 和 ResNet[27] 网络模型，它们都在各种大赛中获得了优异的成绩。值得一提的是，在各种网络通过加深神经网络层数提高识别性能时，GoogleNet 模型在网络设计上做了大胆的尝试，通过引入的 Inception 结构在保持网络结构的稀疏性的同时也利用了密集矩阵的高计算性能。如今的卷积神经网络仍然在随着计算能力的提高而不断发展中，也已经成功被应用于医疗、金融等领域。

2.3.2.2 基本原理

卷积神经网络由输入层、隐藏层和输出层三部分组成。其中的隐藏层是卷积神经网络的核心部分，一般由卷积层、池化层和全连接层搭建而成。卷积神经网络接收多维数据作为输入，隐层中的神经元通过激活函数建立合适的权值映射到下一层中，输出即为样本对应的分类类别。本节我们对卷积神经网络的隐层部分作详细介绍。

相比于其他神经网络结构，卷积神经网络之间链接所需的参数更少。这要归功于卷积神经网络中的卷积层，其具有局部感知和权值共享两大特性。局部感知受启发于生物学里面的视觉系统结构，视觉神经元只响应某些特定区域的刺激局部的接收信息。卷积层中的神经元也没有必要对全局图像进行感知，每个神经元只需对局部状态进行感知，然后到更高层再将其感知到的信息进行综合，这样高层的神经元就具有表达全局图像状态的能力。图片2.5展示了神经元全局感知与局部感知的区别。另一方面，参数共享的特性要求图像在某一部分的特征如果和其他部分相同，在该部分学习到的特征也可以应用于其他部分上。比如我们使用 3×3 的卷积核在一个部分学习到了特征，我们可以将该卷积核作为探测器应用到图像的任意其他部分，判断可不可以作为其他部分的特征的过程在反向传播中进行。

在卷积层的后面一般跟随着池化层，池化层的主要目的是进行下采样操作，下采样可以减少权值数量，在训练的过程中可以提升学习效率。常用的池化操作包括最大池化和平均池化，最大池化操作一般用于保留图像的纹理信息，平均池化操作用于保留图像的背景信息。最大池化操作对临域内的单元取最大，平均池化操作对临域内的单元取平均。如图2.6所示，池化单元大小为 2×2，步

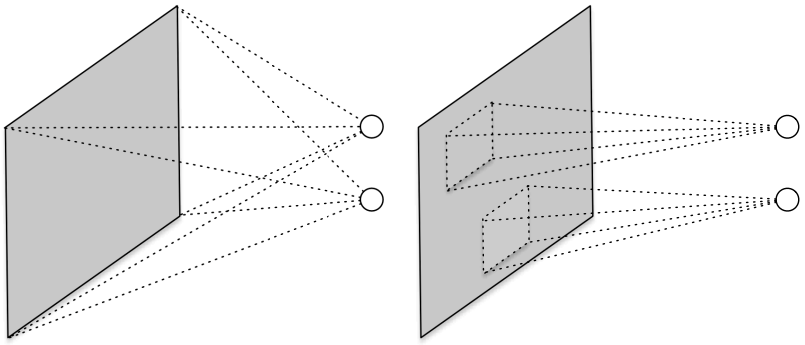

图 2.6 全局感知与局部感知

长为 1。

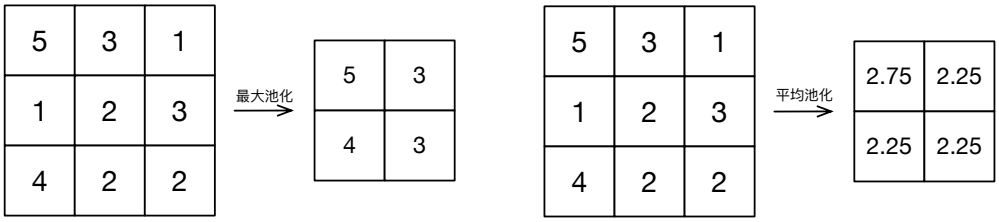

图 2.7 最大池化与平均池化

神经网络模型的输出结果一般是一维向量的形式，所以在卷积层与池化层的后方一般会进行降维操作，将多维矩阵转换为一维向量的形式输入到全连接层。全连接层实际上就是传统前馈神经网络的隐含层。通过激活函数最终映射到长度为类别种类数量的一维向量。

2.3.3 长短期记忆网络与卷积神经网络

第四节 本章小结

第三章 基于 CLNN 模型的 VoIP 流量实时识别研究

在第二章我们提到，本文针对在媒体传输阶段所产生的流量进行特征提取。第一，由于加密技术的发展，SSL/TLS，WEP 和 WAP/WAP2 等加密技术被广泛的应用于信令控制阶段。但是由于媒体传输阶段要求更好的服务质量，考虑到加密技术需要耗费更多的计算资源，大多数 VoIP 应用并没有在其媒体传输阶段使用加密技术；第二，对于少量在媒体传输阶段使用机密技术的 VoIP 应用，因为媒体传输阶段所产生的流量要远远超过信令控制阶段，在大数据技术的支持下，媒体传输阶段的流量相比信令控制阶段可以提取更多的特征用于 VoIP 流量识别；第三，VoIP 在信令控制阶段所采用的应用协议种类繁多，下层协议也包括 UDP 和 TCP，而在媒体传输阶段普遍采用 UDP 进行传输，上层协议也普遍采用 RTP 进行传输，对媒体传输阶段的流量进行识别可以设计更加通用的识别方案。

为了提取有效的特征进行 VoIP 流量识别，本章将从分析 VoIP 媒体流量所携带的特征出发，分析使用 CLNN 模型提取特征的可行性，进而介绍本文进行实时识别所采用的方案以及设计的 CLNN 网络模型。最后将会介绍本文如何使用提取的特征集进行在线实时识别。

第一节 VoIP 特征分析

3.1.1 基本网络特征

RTP/RTCP 是 UDP 的上层协议，其报头可以作为识别 VoIP 流量的基本网络特征。由于本文所研究的 VoIP 流量主要针对于语音流量。RTP 中的 M 和 PT 标记位都可以作为特征用于区分视频和语音流量。RTCP 在对 RTP 媒体传输质量提供反馈信息的同时，会根据与会者的数量调整自己的发送速率。RTP 和 RTCP 数据包普遍采用临近端口号可以作为 VoIP 流量识别的另一特征。

负载类型（PT）是 VoIP 流量最主要的网络特征，用于指明发送端对有效负载采用的解编码器类型。由于语音在不同的通信网络中由不同的解编码器解编码，如果发送端在会话或者广播的中途决定改变编码方式，可以通过设置 RTP 报头中的负载类型位来通知接收端，接收端也可根据负载类型位进行解编码的

工作。RTP 报头中的负载类型预留位为 7 位，意味着 RTP 可以支持 128 种不同的负载类型。VoIP 应用会在不同的网络中基于语音质量和带宽要求等相关条件选择合适的语音解编码器，每种 VoIP 应用所支持的语音解编码器不同。针对本文捕获的 VoIP 流量，表格3.1展示了我们分析所得的 VoIP 应用所使用的部分解编码器类型。RTP 所支持的更多的负载类型可以在 IANA 文档[28] 进行查看。从解编码器的另一角度来看，不同的解编码器会使用不同的比特率传输语音，其与 VoIP 的各项统计特征直接相关，我们将在章节3.1.2进行详细介绍。

表 3.1 VoIP 应用使用的解编码器

应用类型	解编码器类型
Skype	dynamic (96-127), SILK, G.729, PCM A/U, GSM
UUCall	G.723, G.728
KCCall	PCM A/U, ILBC, GSM, G.729
Jumblo	PCM A/U, dynamic (118)
Zoiper	PCM A/U, GSM
Xlite	PCM A/U, iLBC, GSM, Speex
Eyebeam	PCM A/U
ExpressTalk	PCM A/U, Speex
Bria	PCM A/U, Speex, GSM

3.1.2 统计特征

本节所涉及的各项统计特征都与 VoIP 应用所使用的解编码器类型直接相关。许多解编码器是专门为 VoIP 设计的，PCM A/U 解编码器作为最常用的 VoIP 语音解编码器，这两种编码器都生成以 8kHz 采样的 64kbps 的恒定比特率（Constant Bit Rate，CBR）语音流，与其他解编码器相比可以提供较好的语音质量，但同时也较其他具有低比特率的解编码器占用更多的带宽资源。G.723 和 G.729 作为最流行的语音解编码器，占用的带宽比较低，其中 G.723 支持在 5.3kbps 和 6.3kbps 两种码率下工作，G.729 支持在 8kbps 的码率下工作。G.728 也是一种使用恒定比特率的解编码器，它使用 16kbps 的恒定比特率运行。iLBC 是对丢包具有处理能力的专为包交换网络通信设计的解编码器，优于目前流行的 G.729 和 G.723，它可以以 13.33kbps 和 15.20kbps 的比特率工作。GSM 和 Speex 是两种可变比特率（Variable Bit Rate，VBR）解编码器，支持宽范围的比特率，也被广泛的应用于 VoIP 通话技术。多数解编码器都采用 20ms 的打包周期，G.729 采用 10ms 的打包周期，iLBC 在 13.33kbps 下采用 30ms 的打包周

期。

各种解编码器比特率和打包周期的不同，直接造成了 VoIP 流量所占带宽和包到达间隔的不同，同时也间接造成了数据包大小的不同。

3.1.2.1 数据包大小分布

数据包大小分布（Distribution of Packet Size，PSD）是进行 VoIP 流量识别最常用的统计特征。相关工作中介绍的研究方法[16] 直接采用 PSD 作为 VoIP 流量识别的标准，研究方法[15] 提到的数据包长度的熵变化也是数据包大小分布的表现形式。

3.1.2.2 包到达间隔分布

第二节 基于 CLNN 模型的特征提取研究

3.2.1 实时识别基础

为了进行实时识别以达到流量控制的目的，可以通过在 VoIP 会话建立的初期识别出该 VoIP 会话。如果一个 VoIP 会话在可以接受的时间内被识别出，我们认为该识别满足了实时性的要求。假设，在一个组织内部是禁止使用某种 VoIP 应用的，如果我们可以在几秒钟之内准确的识别出一个 VoIP 流，之后在网关处拦截属于该 VoIP 流的全部流量，这就达到了我们实时识别的目的。

根据 VoIP 选择的解编码器的不同，构造并发送数据包的时间在 10ms-30ms 之间，这意味着 VoIP 应用在 1 秒中可以发送 33-100 个数据包供我们进行识别。本文将通过为 VoIP 子流构建数据集并提取特征集的方式，通过所提取的特征集识别 VoIP 子流从而确定整个流所属的应用类型。

3.2.2 VoIP 子流数据集构造方法

本文采用有监督的学习方法训练数据集获取特征集，构造和标记数据集对于很多研究人员来说是一项繁琐的工作。本节我们将介绍构造适用于实时识别的带标签的数据集的方法。

3.2.2.1 数据集构造

数据集构造过程由收集原始数据、标记数据和拆分子流三个步骤构成。

为了获取收集原始 VoIP 流量，我们在真实网络环境中部署了多台电脑用于收集实验数据集。标记数据集的过程实际上是随着收集原始数据完成的，我们在 Windows 操作系统下采用了基于进程的流量收集工具 QPA，在 Linux 系

统下通过确定进程标识符，并按照进程标识符进行流量收集的方式。收集原始数据的过程实际上已经将原始数据按照进程名称或者进程标识符进行标记。

拆分子流的过程是将 VoIP 完整流拆分成多个子流的过程，我们将拆分后的单个子流叫做子流 ^{k}，k 代表该子流中所包含的数据包个数。为了保证子流数据集的随机性，我们设计了基于滑窗的采样方法。从索引为 0 的数据包开始，我们采用大小为 k 的滑窗选择子流 ^{k} 进入我们的数据集，滑窗的步长是在 [1, k] 中随机选取的。这种方式不但保证了数据集的随机性，同时也节省了大量的训练所需时间。图片3.1展示了滑窗大小为 10 时选择 4 个子流 ^{10} 的过程。

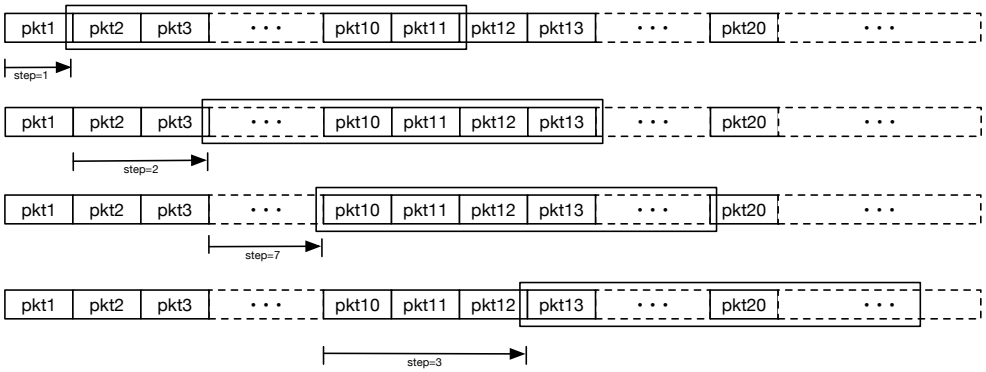

图 3.1 滑窗大小为 10 时选择 4 个子流 ^{10}

3.2.2.2 数据预处理

数据预处理的过程是将数据集处理成可供 CLNN 模型学习的训练数据的过程。本文提出的方法致力于识别 RTP/RTCP 数据流所携带的特征进而识别出 VoIP 流量。由于网络地址转换（Network Address Translation，NAT）技术的发展，对于我们来说，IP 报头所携带的信息是没有识别价值的。UDP 报头所携带的端口号信息对于我们识别 RTP/RTCP 数据包有一定的价值，因此我们保留 UDP 端口号信息。

数据预处理的过程主要包括五个步骤：1）移除网络层和传输层头部；2）为数据包添加传输层端口号；3）添加 2 个时间戳，一个为距离首个数据包的时间差，另一个为距离上个收据包的时间差；4）按照 ASCII 码将子流 ^{k} 转换为矩阵的格式；5）对所得的矩阵进行归一化操作。根据我们的经验，RTP/RTCP 数据包的长度普遍在 50-210 之间，因此我们将矩阵的列设为 256。矩阵的行为 k，即一个子流 ^{k} 所包含的数据包个数。

3.2.3 CLNN 模型研究

基于以上特征分析和构建的子流数据集，本节将介绍本文所使用的 CLNN 模型以及有监督学习的过程。

3.2.3.1 CLNN 模型结构

我们研究了章节2.3.2中所提到的神经网络模型，同时吸收 CLDNN 模型[29]的设计思想设计了本文所使用的 8 层 CLNN 模型。正如大多数的深度学习模型，本文所使用的 CLNN 模型的大多数想法来自于 AlexNet。唯一的不同就是我们将 LSTM 模型加入到卷积神经网络模型中以为语音流提供更强的在时序方面的建模能力。CLNN 模型由 2 层 LSTM，3 层卷积层和 3 层全连接层构成。

在我们的实验中，输入由大小为 k×l 的矩阵组成，k 代表子流 ^{k} 中包含的数据包个数，l 代表矩阵的列数，本文所采取的默认值为 256。我们将 CNN 层的过滤核大小调整为 5×5，调整每层的填充类型为 SAME。结构中前 7 层的 DropOut 值都设置为 0.5。3 层全连接层的前两层使用 ReLU 作为激活函数，最后一层使用 Softmax 作为激活函数。全部 LSTM 层使用 Tanh 作为其激活函数，全部卷积层使用 ReLU 作为激活函数。在我们的 CLNN 模型中，我们去掉了多余的池化层，仅仅在第二层和第五层加入了窗口大小为 2×2 的池化层。

3.2.3.2 CLNN 网络模型的学习过程

对于 CLNN 模型的最后一层全连接层，我们使用 Softmax 作为其激活函数。在我们的实验中，一共有 11 中类型的流量，对于未知的 VoIP 流，我们通过计算其先验概率确定其应用类型。Softmax 函数表示如下：

$$\hat{P}(V_i) = softmax(V_i) = \frac{e^{V_i}}{\sum_{j=1}^{n} e^{V_j}} \tag{3.1}$$

其中 $(V_1, V_2, \cdots, V_i, \cdots, V_n)$ 代表第八层的输入矩阵，$(\hat{P}(V_1), \hat{P}(V_2), \cdots, \hat{P}(V_i), \cdots, \hat{P}(V_n))$ 代表第八层的输出结果，它代表一个未知子流 ^{k} 对于 11 中类型流量的先验概率分布，它是一个大小为 $1 \times n$ 的向量。在我们的实验中，n 为 11，意味着我们的 CLNN 模型可以识别 11 中类型的流量。

在学习的过程中，我们的目标是找出最优的权值以获得最大的似然估计。寻求最大似然估计就意味着需要将多分类的对数损失函数最小化，计算损失的

函数如下：

$$CCE(V_i) = -\sum_{i=1}^{n} P(V_i) \times \log(\hat{P}(V_i)) \tag{3.2}$$

其中 $P(V_i)$ 代表 V_i 的真实概率，它是通过训练数据集中所设置的标签所得的目标矩阵。如果 j 是该子流 ^{k} 所对应的应用类型，则 $P(V_j) = 1$。另外的，当 $i \neq j$ 时，$P(V_i) = 0$。

在本文中，我们使用随机梯度下降（Stochastic Gradient Descent，SGD）优化器来最小化损失函数，并且在每个迭代的过程中使用牛顿动量（Nesterov Momentum）来更新梯度值。梯度增量可通过如下公式计算：

$$\Delta X_t = \tau M_{t-1} - \eta \nabla f(X_{t-1} + \tau M_{t-1}) \tag{3.3}$$

其中，τ 用于表示动量因子，η 用于表示学习率。$g_t = \nabla f(X_{t-1} + \tau M_{t-1})$ 用于计算过度点的梯度，ΔX_t 表示实际下降位移，X_t 表示 t 时刻的位置，M_t 表示 t 时刻的动量。

在我们的实验中，学习率会在每轮计算后进行衰变，衰变规则如下：

$$\eta_i = \eta_{i-1} \times \frac{1}{1 + \rho \times i} \tag{3.4}$$

其中，ρ 表示衰减因子，i 表示当前正在进行第 i 轮计算。

3.2.3.3 特征集

获得更加精确的特征集用于识别 VoIP 流量是线下训练的目标。在我们的实验中，通过 CLNN 模型所提取的针对语音识别的特征集是不可读的。该特征集将会被用于使用 SVM，random forest，decision tree 和 naive bayes 等机器学习方法训练分类器。经过 4 中机器学习方法所训练的分类器均获得了较高的准确率，并且它们将会被用于我们的在线识别阶段。

第三节 在线实时识别方法研究

本节将会介绍如何使用训练所得的分类器在大规模网络中进行 VoIP 流量的实时识别。本文包括 3 个主要组件用于构建我们的实时识别系统，它们分别是流量捕获器、VoIP 流过滤器和分类器。流量捕获器用于捕获 UDP 数据包并按照 IP 地址和 UDP 端口号进行分流；VoIP 流过滤器用于过滤掉明显的非 VoIP 流以提高系统的实时性；分类器是通过我们提取的特征集经过 SVM 方法训练所得，用于识别 VoIP 流的应用类型。

3.3.1 流量捕获器

流量捕获器维护一个待识别流数据库并且具有将捕获的数据包按照 IP 地址和 UDP 端口号分发到对应的待识别流的能力。图片3.2展示了流量捕获器如何在真实网络中根据 IP 地址和 UDP 端口号分流的过程。如果刚到的数据包不能在数据库中找到对应的待识别流，流量捕获器则会使用该数据包的 IP 地址和 UDP 端口号作为键值创建一个待识别流；反之，如何可以找到对应的待识别流，则将该数据包处理后加入到该待识别流。当一个待识别流的数量达到 k 后，这个待识别流将会被送到下一个 VoIP 流过滤器组件中。

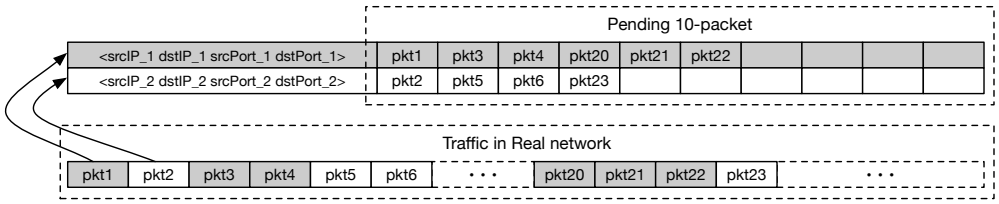

图 3.2 根据 IP 地址和 UDP 端口号进行分流

流量捕获器保证所捕获的数据包均为 UDP 数据包，并且它可以移除待识别流队列中的死流，死流是在有限时间内未达到 k 值的待识别流。

3.3.2 VoIP 流过滤器

VoIP 流过滤器作为实时识别系统的第二个重要组件，它是一个基于规则的过滤器。它接收由流量捕获器发送过来的待识别流，并对该待识别流作出初步的判断。如果初步判断的结果表明该待识别流为非 VoIP 流，因为非 VoIP 流对于我们没有任何意义，我们将不继续对该流做任何处理；如果该待识别流通过了我们设定的全部规则，则该待识别流将会被发送到分类器组件进行进一步识别。

我们从数据集中选择了一系列基准数据，并根据我们长期研究 VoIP 流量的经验设定了几个阈值参数用于过滤非 VoIP 流量。相关算法和参数分别在算法3.1和表格3.2中展示。

3.3.3 分类器

通过 VoIP 流过滤器的待识别流将会被输入第三个组件分类器中，分类器所使用的特征集是在离线训练过程中经过训练所得。我们使用了 4 中机器学习

表 3.2 算法3.1中使用的基准值和阈值参数

	符号	详细描述
1.	min_mpl	全部子流中的最小平均数据包长度
2.	max_mpl	全部子流中的最大平均数据包长度
3.	min_pl	最小数据包长度
4.	max_pl	最大数据包长度
5.	min_miat	全部子流中的最小平均包到达间隔
6.	max_miat	全部子流中的最大平均包到达间隔
7.	min_iat	最小包到达间隔
8.	max_iat	最大包到达间隔
9.	λ_1, λ_2	阈值分别为上限和下限
10.	ρ	比例阈值

的方法重新训练了分类器，我们将具有最好的性能的 SVM 分类器应用于实时识别系统。分类器将 k 维矩阵作为输入并输出一个大小为 $1 \times n$ 的向量，该向量代表此待识别流可能属于每种 VoIP 应用的概率，最大的概率所对应的 VoIP 类型将作为识别结果写入数据库。与此流具有相同键值的 VoIP 数据包将通过查询数据库获得识别结果，供网络管理员进行下一步操作。在识别过程中，一个 VoIP 流会在数据包个数达到不同 k 值的时候被送到对应的分类器，这些分类器都会给出它们的识别结果和它们判定该流属于该识别结果的概率。

第四节 本章小结

本章首先从基本网络特征和统计特征两个角度分析了 VoIP 流量所携带的可供识别的特征，证明了使用媒体传输阶段所产生流量进行 VoIP 流量识别的可行性。之后，本章分析了进行实时识别的基本条件，并介绍了针对 VoIP 实时识别构建其专用的子流数据集和数据预处理的基本方法。基于以上研究，本文针对 VoIP 语音流量设计了在频域和时域两个方面具有良好建模能力的 CLNN 模型，该模型可以自动提取 VoIP 流量的抽象特征，在 VoIP 识别方面具有良好的识别性能。本章最后介绍了在大规模网络中进行实时识别的方法，介绍了实时识别系统的三个重要组件及其对应特性。

算法 3.1: 用于识别 VoIP/非 VoIP 流的算法

Input: 子流 ^{k}
Output: VoIP/非 VoIP
if $\lambda_1$min_mpl < 子流 ^{k} 的平均数据包长度 < $\lambda_2$max_mpl and $\lambda_1$min_miat < 子流 ^{k} 的平均包到达间隔 < $\lambda_2$max_miat then
    for 数据包 in 子流 ^{k} do
        if (数据包长度 < $\lambda_1$min_pl) then
            $ct\_pl\_l \leftarrow ct\_pl\_l + 1$;
        if (数据包长度 > $\lambda_2$max_pl) then
            $ct\_pl\_u \leftarrow ct\_pl\_u + 1$;
        if (包到达间隔 < $\lambda_1$min_iat) then
            $ct\_iat\_l \leftarrow ct\_iat\_l + 1$;
        if (包到达间隔 > $\lambda_2$max_iat) then
            $ct\_iat\_u \leftarrow ct\_iat\_u + 1$;
    if ct_pl_l < ρk and ct_pl_u < ρk and ct_iat_l < ρk and ct_iat_u < ρk then
        return *VoIP*;
    else
        return 非 *VoIP*;
else
    return 非 *VoIP*;

第四章 VoIP 流量实时识别系统实现

本章根据第三章提出的基于 CLNN 模型进行 VoIP 流量识别的方法，设计并实现了一个可在大规模网络环境中进行 VoIP 流量识别的实时系统。本章首先对 VoIP 实时识别的整体框架做了介绍，之后对系统的离线训练和在线识别过程分别做了详细介绍。

第一节 框架设计

图片4.1展示了本文 VoIP 流量实时识别系统的总体框架。图中展示了两个阶段的具体步骤，离线训练阶段所获的的分类器作为实时识别阶段的第三个组件分类器，在实时识别阶段可以对 VoIP 流作出准确识别。

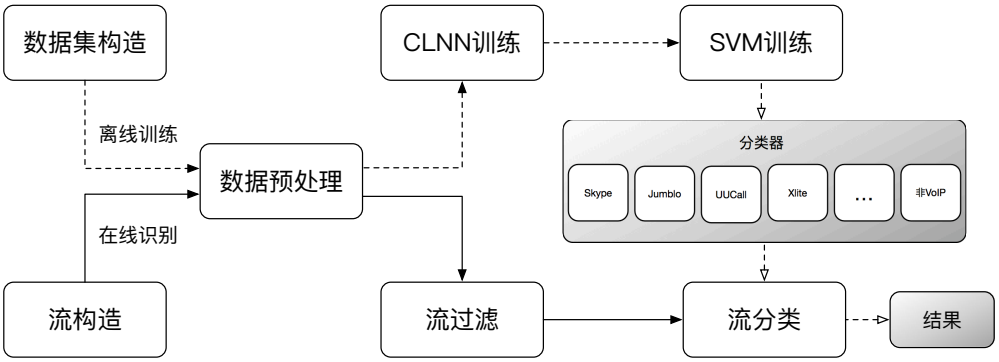

图 4.1 VoIP 流量实时识别框架

4.1.1 离线训练阶段

离线训练阶段包括数据集的构造、数据预处理、CLNN 模型训练和 SVM 训练等四个过程，其目的是通过深度学习提取 VoIP 子流的特征集用于训练适于实时识别的分类器。上文我们提到，构造数据集的过程是会将我们采集的原始数据按照应用类型进行标记并且将 VoIP 完整流按照不同的 k 值进行分流以适用于实时识别；之后，预处理过程会对数据集及其所对的标签进行预处理工作，数据集中的 VoIP 子流将会被处理成可输入 CLNN 模型的矩阵，其对应的标签将会被作为目标矩阵一同输入到 CLNN 模型进行训练；CLNN 模型接收数据集

执行训练过程，经过不断的迭代过程，得到具有最小损失的模型，通过该模型提取出的特征集将会用于 SVM 训练分类器；使用 SVM 训练所得的分类器的准确率没有 CLNN 模型高，但是其具有更良好的实时性。如图4.1中所示，分类器可以识别出各类型的 VoIP 应用流量，同时也可对非 VoIP 的媒体传输流进行识别。

4.1.2 在线识别阶段

在线识别阶段包括流构造、数据预处理、流过滤和流分类几个过程。章节第三节中我们提到的 3 个用于实时识别的组件分别用于执行流构造、流过滤和流分类几个过程。图片4.2展示了流量捕获器、VoIP 流过滤器和分类器工作的基本流程。其中待识别子流中键值为 key_3 的子流由于数据包数量达到了 k，它将会被送往 VoIP 流过滤器执行过滤操作。通过过滤器后将会由分类器作出识别结果存入数据库。图中展示了两个被流量捕获器捕获的数据包其键值分别为 key_1 和 key_2，由于 key_2 数据包未在数据库中查到其对应的识别结果，它将会被用于构造待识别子流等待识别。而 key_1 数据包在数据库中查到其识别结果为 Skype，该数据包会被判断为 Skype 数据包。

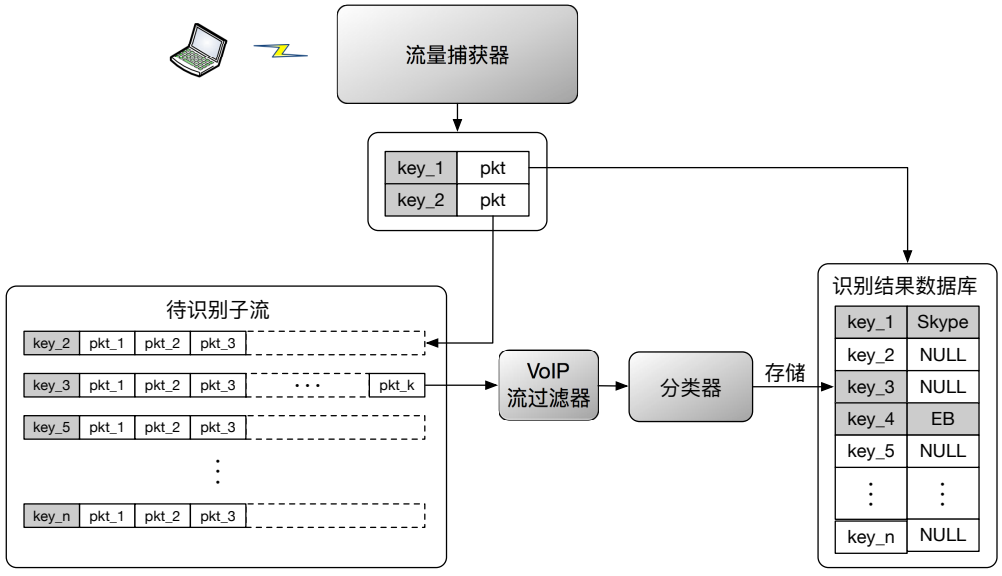

图 4.2 流量捕获器、VoIP 流过滤器和分类器工作流程示意图

第二节 数据集构造

深度学习是在海量数据的环境中提取有价值的信息，足够海量的数据是深度学习的基础。构造带标签的数据集对于研究人员来说是一项繁重的任务，另外由于 VoIP 流量的隐私性，国内外没有公开的海量数据集供我们使用。本文所涉及的落地应用主要针对于可以直拨 PSTN 的 VoIP 应用，因此采集了 10 种流行的 VoIP 应用流量和其他非 VoIP 流量用于训练深度学习模型。

4.2.1 原始数据收集

为了保证所捕获的 VoIP 原始流量的多样性，避免因流量过于单一造成所提取的特征集不具备足够的泛化能力，本文使用多台计算机在多种操作系统下进行流量的收集工作，并使用基于线程的流量捕获工具进行流量标记工作。

为了获得较为纯净的数据集，本文在原始数据收集阶段使用基于线程的方法进行捕获工作。在 Windows 操作系统下，我们使用 QPA 进行流量捕获。QPA 是一款开源的抓包、提供正则表达式识别引擎、按流量类型自动归类、能实时快速的分析软件，我们主要使用其基于进程的工作方式进行流量采集工作。由于 QPA 未在 Linux 环境下提供相应软件，我们使用 tcpdump 工具执行流量捕获工作。当 VoIP 应用开始通话后，我们首先分析其使用的端口号，针对该端口号使用命令 “tcpdump -i eth_name trans_protocol port port_id” 进行流量捕获。基于线程和端口的方法使得所捕获的流量可以很容易的按照应用类型进行标记用于深度学习。来自视频应用和游戏的流量都将被标记为非 VoIP 流量。

4.2.2 数据预处理

捕获的原始数据是以 pcap 文件的格式存储的，每个 pcap 文件存储的内容为一次 VoIP 通话所生成的流量。数据预处理就是要将原始 pcap 文件处理成可供 CLNN 模型学习的训练数据，一是要按照章节3.2.2.1中介绍的方法进行子流的划分用于提取适于实时识别的特征集，二是要将子流处理成可输入 CLNN 模型的矩阵格式。

为了采取多个数据集适应不同程度的实时识别，本文按照 k 值的不同训练了多个 CLNN 模型。k 值包括 2，4，6，8，10，20，40 和 100，以上多个 k 值所训练出的模型具有不同程度的实时性。其识别准确率与 k 值成正比，实时性与 k 值成反比。图片4.3展示了以 pcap 文件作为输入预处理后以子流作为输出的过程。输出的子流以文本文件的方式进行保存。

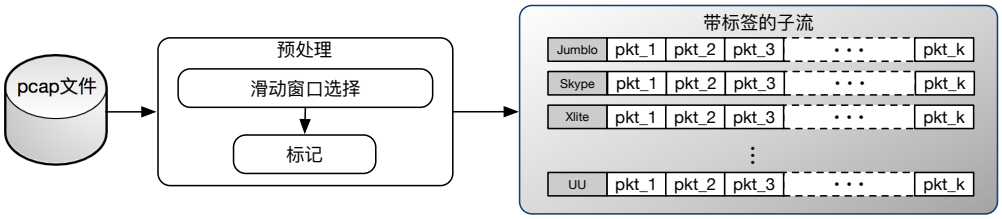

图 4.3 数据预处理详情

按照我们所提出的将 VoIP 子流转化成矩阵的方法，图片4.4展示了一个用于训练的 Skype 子流 ^{100} 图片格式的表现形式。

图 4.4 Skype 子流 ^{100} 的图片表现形式

第三节 CLNN 模型离线训练

基于 CLNN 模型的离线训练过程是进行 VoIP 流量识别的核心。训练过程中，CLNN 模型可以通过对训练数据的学习，将低维层次的特征转化成高维层次的特征，从而构造出用于 VoIP 流量识别的特征集。CLNN 模型通过训练数据集自学习过程中，会在正向传播的过程中计算损失，并通过随机梯度下降的方式逐层向前反馈进行反向传播，从而最小化损失。

本文使用 Keras 库搭建 CLNN 模型，Keras 是一个基于 Theano 和 TensorFlow 的深度学习库，其具有高度模块化和可扩充的特性，并且可以支持 CPU 和 GPU 的无缝切换。Keras 高度的支持 CNN 和 RNN 的结合，本文使用的 CNN 和 LSTM 的策略可以很简单的使用 Keras 实现。Keras 提供的函数式模型是用户定义多输出模型、非循环有向模型或具有共享层的模型等复杂模型的途径，本文使用函数式模型进行了 CNN 和 LSTM 的结合，从而创建了

具备更强的提取 VoIP 流量特征能力的模型。

CLNN 模型经过不断的迭代学习后，输出结果就是用于 VoIP 子流识别的一系列权重值。CLNN 模型使用这些权重值进行正向计算从而可以计算出 VoIP 子流的识别结果。

4.3.1 模型基本结构

CLNN 模型在学习过程中会生成大量的参数，大量的参数也是 CLNN 模型准确识别的基础。本节将以 k=100 对 CLNN 模型每层所使用的具体参数进行详细介绍。之后列出本文训练的 8 种模型生成的权值参数个数以及保存权值的 HDF5 文件大小。

表 4.1 CLNN 模型参数详情

<table>
<tr><th colspan="3">$CNN + Dense$</th><th>LSTM</th></tr>
<tr><td rowspan="3">Conv1</td><td>输入</td><td>$100 \times 256 \times 1$</td><td rowspan="24">输入：100×256
单元：100
权值个数：142800
Dropout：0.5</td></tr>
<tr><td>参数</td><td>卷积核：$5 \times 5 \times 48$；步长：2</td></tr>
<tr><td>输出</td><td>权值个数：1248；Dropout：0.5</td></tr>
<tr><td rowspan="3">Conv2</td><td>输入</td><td>$50 \times 128 \times 48$</td></tr>
<tr><td>参数</td><td>卷积核：$3 \times 3 \times 128$；步长：2</td></tr>
<tr><td>输出</td><td>权值个数：55424；Dropout：0.5</td></tr>
<tr><td rowspan="3">Conv3</td><td>输入</td><td>$25 \times 64 \times 128$</td></tr>
<tr><td>参数</td><td>卷积核：$3 \times 3 \times 192$；步长：1</td></tr>
<tr><td>输出</td><td>权值个数：221376；Dropout：0.5</td></tr>
<tr><td rowspan="3">Conv4</td><td>输入</td><td>$25 \times 64 \times 192$</td></tr>
<tr><td>参数</td><td>卷积核：$3 \times 3 \times 192$；步长：1</td></tr>
<tr><td>输出</td><td>权值个数：331968；Dropout：0.5</td></tr>
<tr><td rowspan="3">Conv5</td><td>输入</td><td>$25 \times 64 \times 192$</td></tr>
<tr><td>参数</td><td>卷积核：$3 \times 3 \times 128$；步长：1</td></tr>
<tr><td>输出</td><td>权值个数：221312；Dropout：0.5</td></tr>
<tr><td rowspan="3">Dense1</td><td>输入</td><td>204800</td></tr>
<tr><td>参数</td><td>单元：2048</td></tr>
<tr><td>输出</td><td>权值个数：419432448；Dropout：0.5</td></tr>
<tr><td rowspan="3">Dense2</td><td>输入</td><td>2048</td></tr>
<tr><td>参数</td><td>单元：100</td></tr>
<tr><td>输出</td><td>权值个数：204900；Dropout：0.5</td></tr>
<tr><td rowspan="3">Dense3</td><td>输入</td><td colspan="2">200</td></tr>
<tr><td>参数</td><td colspan="2">单元：11</td></tr>
<tr><td>输出</td><td colspan="2">权值个数：2211</td></tr>
</table>

表格4.1展示了 CLNN 模型每层所采用的参数详情。表格所示由两部分组

成，第一部分是由 CNN 和 Dense 组成的用于提取频域特征的模型，第二部分是 LSTM 层用于提取时域特征。两部分在 Dense3 也就最后一个全连接层进行组合，Dense3 将频域特征和时域特征进行组合最大患的提取 VoIP 流量携带的特征。以第一层 Conv 举例，输入为 $100\times256\times1$ 的矩阵，卷积核大小为 5×5，深度为 48。通过 $W=(r_{filter}\times c_{filter}\times d_{input}+b)\times d_{filter}$ 可以计算该层权值个数为 1248，其中每个神经元包含 $5\times5\times1$ 权值和一个偏差。以第一层 Dense 层举例，输入为 204800 维的向量，输出为 2048 维的向量，通过 $W=(dim_{input}+b)\times dim_{output}$ 可以计算其权值个数为 419432448。LSTM 层输入为 100×256 的矩阵，输出单元为 100，可以通过 $W=4\times dim_{output}+r_{input}+1\times dim_{output}$ 计算其权值个数，其中 4 代表每个 LSTM 单元包括 4 个激活函数。

4.3.2 Keras **搭建模型**

本文使用 Keras 提供的 Model 类搭建函数式模型，需要导入 Keras 提供的 models 包。使用

model=Model(input=subflows,output=[model])

创建模型。之后，向该模型中添加隐层，本文主要使用了 3 种隐层，卷积层、全连接层和 LSTM 层。依次可以使用

model = Conv2D(filters, kernel_size, strides, activation)(model)

model = Dense(units, activation)(model)

model = LSTM(units, activation)(model)

添加到 CLNN 模型中。另外，本文选择性使用了池化层（MaxPooling2D）和失活层（Dropout）加速训练的过程。在维度转换时，本文使用了平滑层（Flatten）和变形层（Reshape）提供的支持。

本文搭建的 CLNN 模型最主要的特性是针对频域特征和时域特征同时进行提取，最后使用全连接层综合两个类型的特征。以下代码段展示了如何将 CNN 和 LSTM 提取的特征进行组合。

```
input = Input(input_shape)
conv = create_cnn_layers(input)
lstm = create_lstm_layers(input, input_shape[0])
model = concatenate([conv, lstm],axis=1)
model = Dense(nb_class, activation='softmax')(model)
```

4.3.3 迭代训练

CLNN 模型迭代训练是通过训练数据集不断调整权值寻找具有最小误差的权值的过程。

本文使用随机梯度下降的算法进行权值的更新，同时在随机梯度下降时引入冲量，冲量用于表示上一梯度对当前梯度的贡献率。梯度下降算法采用小批量更新的策略，每一批样本数据计算一次损失函数并更新参数。随机梯度下降优化器可通过 SGD 类进行定义。

sgd = SGD(lr=0.0, decay=0.0, momentum=0.9, nesterov=True)

为 CLNN 模型配置损失函数和优化器类型通过 compile 函数完成，

model.compile(loss='categorical_crossentropy', optimizer=sgd)

以上配置所使用的损失函数为针对多类的对数损失函数。

上文中定义随机梯度下降函数时没有为 SGD 指定学习率，因为本文采取每轮训练完成后都要对学习率进行更新的策略。以下代码段为对上一章中介绍的公式3.4的具体实现。

```
def step_decay(epoch):
    initial_lrate = 0.01
    drop = 0.5
    epochs_drop = 5.0
    lrate = initial_lrate *
        math.pow(drop,math.floor((1+epoch)/epochs_drop))
    print(lrate)
    return lrate
lrate = LearningRateScheduler(step_decay)
```

训练中使用 EarlyStopping 优化训练行为，当损失不再减小时则提前停止训练过程。

earlystopping = EarlyStopping(monitor='val_loss', patience=1)

我们在训练中对全部数据指定进行 20 轮的计算，当然，在满足 EarlyStopping 的条件后训练也会提前结束。每 200 个样本计算后反向传播进行梯度的更新。以下函数最后的 callbacks 参数展示对学习率更新以及 EarlyStopping 的调用。

表 4.2 8 个 CLNN 模型生成的权值个数及 HDF5 文件大小详情

CLNN-k	**输入**	**权值个数**	HDF5(MB)
k=2	$2\times256\times1$	17,818,697	87
k=4	$4\times256\times1$	17,820,823	87
k=6	$6\times256\times1$	34,600,197	154
k=8	$8\times256\times1$	34,602,387	154
k=10	$10\times256\times1$	51,381,825	221.6
k=20	$20\times256\times1$	84,947,847	355.8
k=40	$40\times256\times1$	168,859,507	691.3
k=100	$100\times256\times1$	420,613,687	1619.3

model.fit(X_train, Y_train, batch_size=200,validation_data=(X_val, Y_val),epochs=20,callbacks=[early_stopping,lrate])

4.3.4 模型保存

经过训练得到的最终模型需要进行存储，模型分为两个部分进行存储。一是需要对 CLNN 模型的基础架构进行存储，我们使用 json 格式的文件进行存储；二是要对训练后的模型的权重参数进行保存，我们使用可提供层次数据格式存储的 HDF5 进行存储。在实时识别阶段我们将使用该模型进行 VoIP 流量的识别。表格4.2展示了本文训练所得的 8 个 CLNN 模型生成的权值个数及存储为 HDF5 文件得大小详情。

第四节 VoIP 流量实时识别系统

VoIP 流量实时识别系统是部署在大规模真实网络出入口节点上的用于在线识别 VoIP 流量的分布式系统。为了应对大规模真实网络上复杂多变的环境，我们采用 Apache Kafka 流处理平台接入真实网络环境。Kafka 流处理平台使用 ZooKeeper 作为其后端分布式服务器管理、协调 Kafka 代理，可以为处理实时数据提供高吞吐、低延迟的性能，解决了真实网络中流速不均衡造成的难题。其次，Kafka 除了为我们提供负载均衡的环境之外，也作为我们的数据备份模块备份网络流量。为了达到实时计算的目的，我们采用 Apache Storm 作为流式计算框架执行计算任务。用户通过自定义 Spout 和 Bolt 指定信息源和计算措施从而达到批量、分布式处理流式数据的目的。本节我们将分为 Kafka 和 Storm 两个模块进行介绍。

4.4.1 Kafka 模块

Kafka 模块是用于直接对接真实网络环境的基础模块，其可以按照不同交换机和路由器创建标题，供 Storm 模块进行消费。

图片4.5展示了真实网络如何接入 Kafka 集群的结构图。本文实验中将局域网中的路由器接入 Kafka 集群模仿真实网络环境中的交换机。

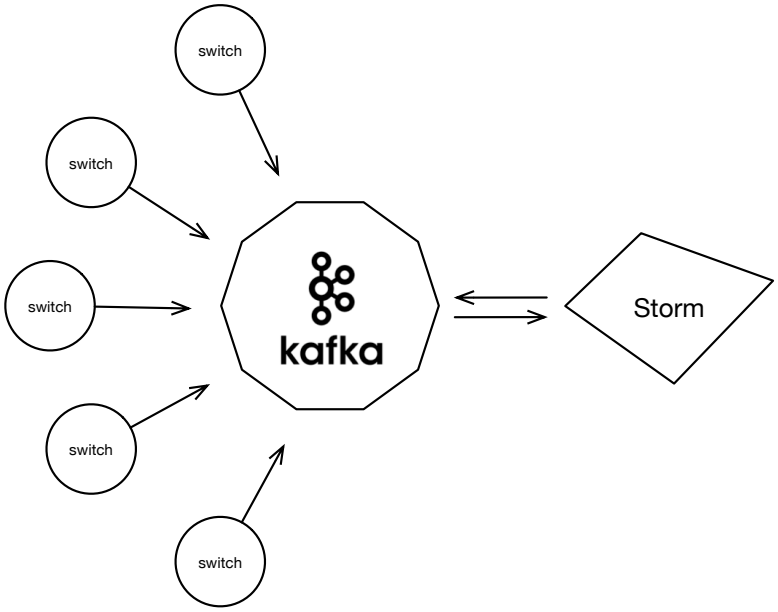

图 4.5 真实网络环境接入 Kafka 集群结构图

Kafka 提供良好的 Java 编程接口，我们主要使用其提供的生产者接口将真实网络中的流量发送到 Kafka 集群中。以下分别为发送单条数据和发送多条数据所对应的接口。

public void send(KeyedMessage<K,V> message)

public void send(List< KeyedMessage<K,V> > messages)

4.4.2 Storm 模块

Storm 模块是支持实时识别的核心模块，其对来自 Kafka 集群的动态的、连续的以及大数据量的流量作出及时的处理。

本文使用 Kafka Spout 作为消费者客户端消费 Kafka 集群中未处理的流量，同时 Kafka Spout 也作为 Storm 拓扑结构的数据源将流量吐到 Storm 拓扑中进

行处理。其次，我们使用两个 Bolt 分别作为流量捕获器的 UDP 识别和分流部分，后继 Bolt 可能是一个 VoIP 流过滤器也可能直接是一个分类器，这根据不同的 k 值作出不同调整。具有较小 k 值的 VoIP 流不易进行初步 VoIP 流过滤操作，其次较小的 k 本身要求要高的实时性，因此不经过 VoIP 流过滤器直接送入分类器进行识别。最后一个 Bolt 提供写入 Redis 数据库的操作。图片4.6展示了 Storm 模块的拓扑结构图。其中分类器 Bolt 对应多种 CLNN 模型所构成的分类器，它们产生的结果和该结果所对应的可能性都将输出到 Redis 数据库中，网络管理员可以根据实时性和准确度要求拿取符合要求的结果。

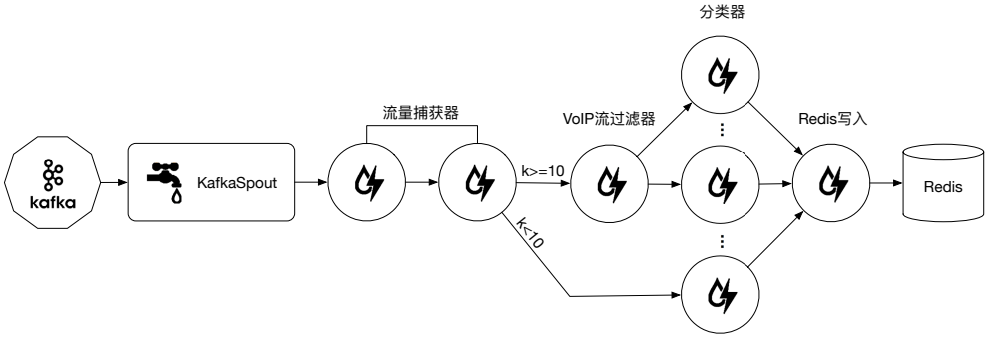

图 4.6 Storm 模块拓扑结构图

以下代码段展示了 Storm 拓扑的搭建过程。

```
TopologyBuilder tb = new TopologyBuilder();
tb.setSpout("spout", new TrafficSpout(), 3);  // executor num
tb.setBolt("udpbolt", new UDPBolt()).shuffleGrouping("spout");
tb.setBolt("pendingflowsbolt", new PendingFlowsBolt())
        .fieldsGrouping("udpbolt", new Fields("key"));
tb.setBolt("classifierbolt10", new ClassifierBolt10())
        .shuffleGrouping("pendingflowsbolt", "10");
tb.setBolt("filterbolt", new FilterBolt())
        .shuffleGrouping("pendingflowsbolt", "100");
tb.setBolt("classifierbolt100", new ClassifierBolt())
        .shuffleGrouping("filterbolt");
tb.setBolt("rediswriterbolt", new RedisWriterBolt())
        .shuffleGrouping("classifierbolt10")
```

```
.shuffleGrouping("classifierbolt100");
```

第五节 本章小结

本章依托第三章设计的移动应用流量识别算法，实现了一个移动应用流量识别系统。本章第一节首先系统的介绍整体框架，分为离线训练阶段和在线识别阶段做了简单介绍；第二节对本文采用的构建数据集的方法和数据集详细信息做了介绍；第三节对本文所使用的 CLNN 模型的结构做了详细介绍，并从编程的角度介绍了模型的搭建、训练和输出的过程；最后第四节介绍了为支撑大规模网络环境中的识别，我们采用的 Kafka 和 Storm 框架，并使用这两种框架开发了适用于本文所提出的方法的应用环境。

第五章 实验结果和性能评估

本章将从 CLNN 模型准确率和 VoIP 识别系统的实时性两个方面进行评估。针对本文所采取的 VoIP 流量，使用本文所搭建的 CLNN 模型进行训练和验证，并分析了 8 个不同 k 值训练出的 CLNN 模型的识别准确率。同时，为证明本文所提取的特征集的有效性，我们使用 4 种机器学习的方法通过该特征集训练数据集并进行了对比。最后，从实时性的角度分析实时系统的可用性。

第一节 实验基础

5.1.1 实验环境

实验环境包括开发环境和运行环境两个部分，开发环境用于训练 CLNN 模型和实时系统开发，运行环境用于搭建分布式服务器、运行 Kakfa 和 Storm 集群、存储识别结果和回放 VoIP 流量。

表格5.1展示了本文开发环境和软件。本文深度学习方面的内容基于 Python 开发，使用的深度学习框架为 Keras。本文使用 GPU 环境加速深度学习过程，使用 CUDA 作为其运算平台。分布式集群相关使用 Java 进行开发。

表 5.1 开发环境

	项目	具体环境及版本
硬件环境	CPU	四核 3.6GHz Intel Core i7
	GPU	NVIDIA GeForce GTX 1070
	RAM	8GB
软件环境	操作系统	CentOS7
	GPU 运算平台	CUDA Toolkit 8.0
	深度框架	Keras
	编程语言	Python 2.7.10, Java 1.8.0

表格5.2展示了本文 VoIP 实时识别系统的运行环境。本文使用 ZooKeeper 提供分布式协调服务，Kafka 和 Storm 作为运行在 ZooKeeper 之上的分布式应用提供存储和计算功能。键值数据库使用 Redis 提供分布式查询功能。本文为模拟真实网络环境的 VoIP 流量，使用了 tcpreplay 软件回放我们采集到的 VoIP

流量。

表 5.2 运行环境

	项目	具体环境及版本
硬件环境	服务器	3 台, RAM 8GB, 磁盘空间 200GB
软件环境	操作系统	CentOS7
	分布式系统	ZooKeeper 3.4.12
	集群服务	Kafka 2.0.0, Storm 1.2.2
	数据库	Redis 5.0.3
	回放软件	tcpreplay

5.1.2 数据集

本文针对 10 种流行的 VoIP 应用和多种非 VoIP 的视频和即使游戏应用进行了流量的捕获工作。为了提升识别的泛化能力，我们部署了 8 台计算机，其中 6 台计算机安装了 Windows 操作系统，2 台计算机安装了 Linux 操作系统，2 台 Linux 计算机安装了提供支持在 Linux 环境下使用的 4 种 VoIP 应用，6 台装载 Windows 操作系统的计算机均安装了 10 种 VoIP 应用。其次我们捕获了视频应用、视频通话应用和即时游戏应用的 UDP 流量作为非 VoIP 应用流量。为了保证 VoIP 通话线路的泛化性，我们从国内包括河北省、吉林省、广东省和四川省多个地方拨打电话用于流量获取工作。

本文将采集到的数据分成两个数据集，数据集 1 用于训练 CLNN 模型，数据集 2 用于进一步验证 CLNN 模型并且用作回放流量验证实时系统的性能。5.3展示了数据集 1 的统计信息和经过处理后用于训练的子流 ^{10} 个数。数据集 2 不做子流处理，其流量大小与数据集 1 相同。训练过程中数据集 1 中的 80% 子流用作训练 CLNN 模型，20% 的子流用于训练过程中验证准确率。

5.1.3 训练参数

训练过程中使用的 SGD 算法是一种学习率非自适应的算法，学习率的设置对实验结果有直接的影响。我们结合深度学习一般经验并按照实验结果进行调整，初始学习率设置为 0.01，衰减规则按照本文以上所述每两轮训练进行一次衰减。nesterov 动量因子设置为 0.9，它会吸收上一梯度 90% 的贡献。我们针对 8 个 k 值分别对数据集训练了 20 轮，每 200 个样本进行一次梯度更新。

表 5.3 数据集 1 详细信息

VoIP **应用类型**	**平台**	**数据集大小** (MB)	**子流数** (k=10)
Skype	Windows, Linux	3908.8	669984
UUCall	Windows	2709.4	691566
KCCall	Windows	3128.8	808224
ALTCall	Windows	2795.2	692002
Jumblo	Windows, Linux	3704.6	871468
Zoiper	Windows, Linux	4418.1	677114
Xlite	Windows, Linux	5165.3	638288
Eyebeam	Windows	4524.7	616773
ExpressTalk	Windows	4633.1	602637
Bria	Windows	4476.0	598724
non-VoIP	Windows	11324.5	973522

第二节 CLNN 模型评估

5.2.1 评估指标

在多分类问题中，不同的场景需要不同的评估指标。一般来说，准确率（Accuracy）就可以评估一个分类模型的性能，用于衡量分类正确的比例。可以使用如下公式计算准确率：

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} 1_{\hat{y}_i = y_i} \tag{5.1}$$

其中，$\hat{y}_i$ 表示被分类模型预测的类别，y_i 表示样本的真实类别。准确率是一个适用范围最广的评估指标，可以反映系统的全局识别性能，但是准确率在反映分类器对正负样本的识别能力上有着不足。本文引入真正类率（True Positive Rate, TPR）来表达分类器所识别出的正实例占所有正实例的比例，负正类率（False Positive Rate, FPR）在表达能力上与 TPR 是互补的关系。对于二分类问题，即将样本分为正类和负类两种情况的问题来说，会出现 4 种分类结果：如果一个实例是正类并且也被预测成正类，即为真正类（True Positive, TP），如果实例是负类被预测成正类，称之为假正类（False Positive, FP）。相应地，如果实例是负类被预测成负类，称之为真负类（True Negative, TN），正类被预测成负类则为假负类（False Negative, FN）。通过这种分类方式，我们重新表达准确率如下：

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.2}$$

表 5.4 8 个模型对训练数据集和验证数据集的识别率

模型 k	损失	训练集	验证集
k=2	0.015865	0.995560	0.996130
k=4	0.007670	0.997700	0.997860
k=6	0.004933	0.998520	0.999340
k=8	0.004393	0.998690	0.999130
k=10	0.001159	0.999650	0.999650
k=20	0.000546	0.999880	0.999880
k=40	0.002920	0.999120	0.999460
k=100	0.006689	0.997620	0.998730

评估指标 TPR 和 FPR 计算如下：

$$TPR = \frac{TP}{TP+FN} \tag{5.3}$$

$$FPR = \frac{FP}{FP+TN} \tag{5.4}$$

5.2.2 识别性能评估

根据以上 CLNN 模型架构及实验基础中使用的参数，我们对 11 类流量分为 8 个 k 值进行了训练得到了 8 个 CLNN 模型。本节将对这 8 个模型按照第一节中介绍的评估指标进行评估。

表格5.4展示了 8 个模型对训练数据集和验证数据集识别的准确率。训练数据集来自数据集 1 中的 80%，验证数据集来自数据集 1 中的 20%。我们可以看到其识别率均接近于 1。

因为训练过程中普遍存在过拟合的现象，甚至在验证阶段可能不会表现出来。因此我们使用数据集 2 进行全面的验证。图片5.1中的前十个图表展示了使用数据集 2 进行验证的 10 种 VoIP 应用流量所对应的 TPR，后两个图表展示了数据集 2 的 FPR 和 TPR 分布。很明显的，在前 4 张图表中我们可以看到一些异常值，分别是使用模型-20 进行识别 Skype 流量、模型-4 识别 Xlite 流量、模型-6 识别 Eyebeam 流量和模型-2 识别 Bria 流量时，这表明这四种模型在训练时针对这几种流量出现了过拟合现象。我们在实时识别过程的过程中暂时放弃使用这几个模型识别出的结果对应四种流量的结果。在未来的研究过程中，第一步我们需要将数据集 2 合并到数据集 1 中再次训练，第二步我们考虑加入 L1 和 L2 正则层避免过拟合现象。抛除以上几个异常值，综合显示 CLNN

表 5.5 VoIP 实时识别系统识别 100 个流所耗费的时间

Model	k=2	k=4	k=6	k=8	k=10	k=20	k=40	k=100
Time(s)	27.55	20.12	29.09	30.53	48.00	73.62	141.33	309.51

模型有优秀的识别性能。

图片5.2展示了针对两个数据集识别时识别正确和识别错误的结果所对应的概率分布。如图所示识别正确的结果所对应的识别概率普遍接近于 1，而识别错误的结果所对应的概率分布在 0.2-0.98 之间，普遍要低于识别正确的结果所对应的概率。该概率将作为实时识别系统初步判断识别结果的阈值。

第三节 实验对比

5.3.1 特征集对比

由于 VoIP 数据集的特殊性，本文所使用的识别方案无法与已存的识别方案直接进行对比。我们使用本文 CLNN 模型提取的特征集与文章[13] 中的所使用特征集进行对比，分别使用 SVM，随机森林，C5.0 决策树和朴素贝叶斯四种机器学习方法对两种特征集训练了分类器。图片5.3展示了 CLNN 模型提取的特征集与文章[13] 中使用特征集的对比结果。

5.3.2 其他方法对比

第四节 实时性评估

为了验证本文搭建的 VoIP 流量实时识别系统的实时性，本文使用数据集 2 模拟真实网络环境中的流量进行实时性验证。我们使用 tcpreplay 软件回放所收集到的 VoIP 流量，tcpreplay 可以将我们收集到的 pcap 文件进行回放，并且其支持在回放的过程中修改网络层报头。我们通过 tcpreplay 工具在流量纯净的网卡上进行回放工作，我们记录从收到 VoIP 流的第一包到识别出结果所耗费的时间。表格5.5展示了实时识别系统识别 100 个 VoIP 流所耗费的时间。可以看到，当我们使用 k 值为 100 的分类器进行识别平均识别每个流需耗费 3 秒钟，使用 k 值为 2 的分类器耗费 0.3 秒钟。对于不同情形下的识别需求，网络管理员可以按需对识别结果作出应对。

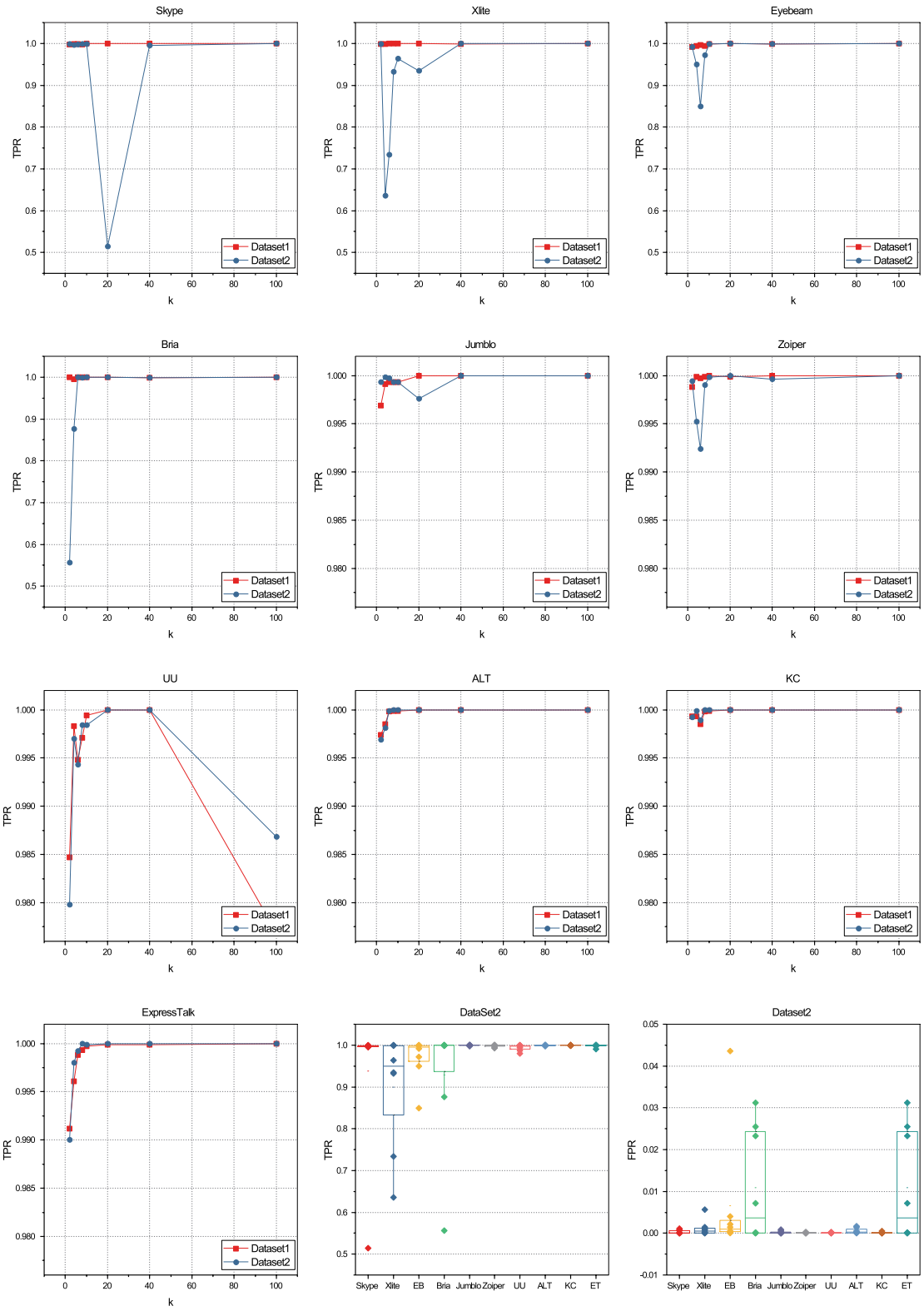

图 5.1 8 种 CLNN 模型的 TPR 和 FPR 结果

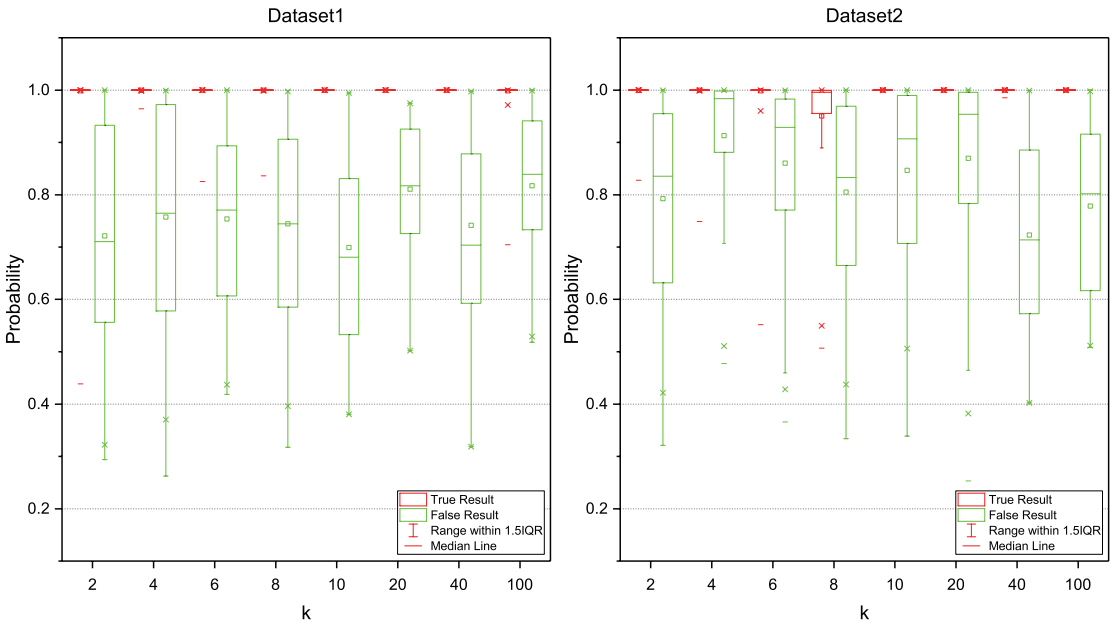

图 5.2 识别结果的概率分布状况

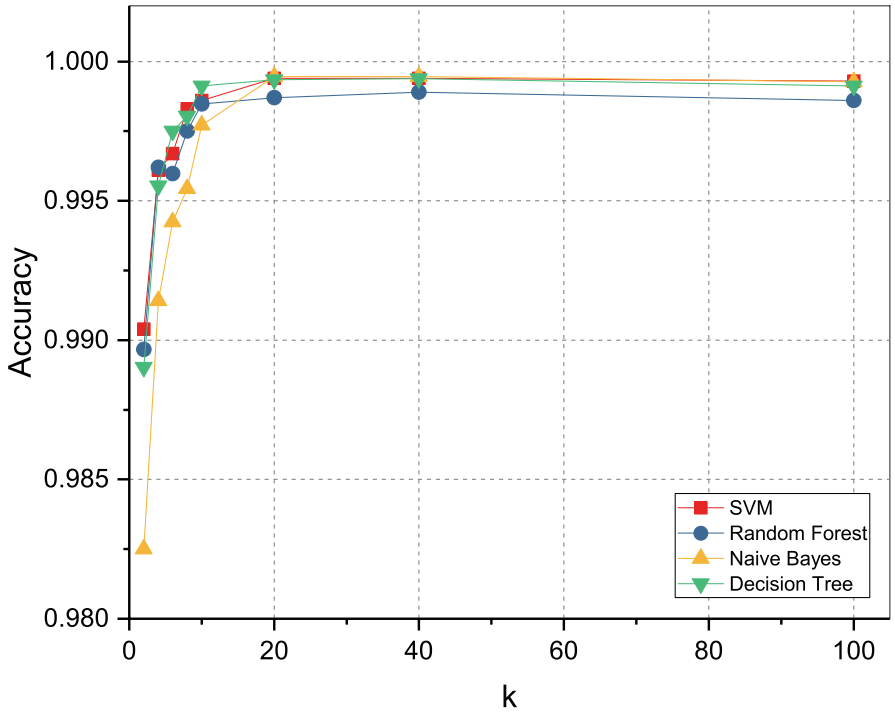

图 5.3 CLNN 模型提取特征集与文章[13] 中特征集对比

第五节 本章小结

本章首先介绍了 CLNN 模型的训练环境和 VoIP 流量实时识别系统的运行环境，之后对训练数据集和深度学习所使用的参数做了详细介绍。然后针对 CLNN 模型的识别性能从不同的层面做了评估，证明了 CLNN 模型的有效性。其次，我们还将本文所提出的方法与其他文章中的识别方法做了比较。最后，本文对 VoIP 流量实时识别系统的实时性作了评估。

第六章 总结与展望

VoIP 应用的广泛使用势必会造成诸多的社会问题，为了使 VoIP 应用是可控的，对于 VoIP 应用流量的有效监管是必要的。由于 VoIP 应用的加密性和 P2P 的特性，VoIP 流量的识别难度要远远高于其他流量的识别。本文总结整理了现存的 VoIP 流量识别方法，并分析了这些方法的优缺点。在此基础上，我们针对 VoIP 应用媒体传输流量进行了分析，设计了可以自动提取媒体传输流量的 CLNN 模型，该模型可以从时域和频域两个维度上最大限度的提取其特征用于 VoIP 识别。其次，本文提出了针对 VoIP 子流的实时识别方法，设计了可以应用于大规模真实网络的 VoIP 流量实时识别系统。同时，我们对于文中使用的各种方法进行了实验验证，证明了方法的可靠性。

第一节 论文总结

本文的主要工作如下：1）本文研究了 VoIP 流量识别的关键技术。对于 VoIP 流量识别方法分为多个层次进行了介绍，分析了每种方法的针对点和适用性范围。

2）本文研究了与 VoIP 流量识别相关的各方面的知识。包括 VoIP 相关技术、流分类、深度学习等多个方面的内容，在此基础上，本文分析了 VoIP 流量所携带的各种可能用于进行识别的特征。之后结合卷积神经网络和长短期记忆网络模型在时域频域方面的建模能力，设计了可以自动提取 VoIP 流量特征的网络模型。本文同时在 VoIP 识别实时性方面作出要求，设计了基于子流进行识别的方法。结合深度学习方面的能力，该模型在保证识别精度的同时也保证了识别要求的实时性。

3）针对本文所提出的方法，我们设计了并实现了一个可以应用于大规模真实网络的 VoIP 流量实时识别系统。该系统采用 Storm 作为其流计算引擎，Kafka 作为其负载均衡系统，保证了在大规模真实网络中的识别性能。在 VoIP 识别系统中引入了 CLNN 模型作为分类器，该分类器保证了对 VoIP 流量的识别精度。其次，我们对实时识别的性能通过加入 VoIP 流过滤器进行提升，从而提高了识别系统的实时性能。

第二节 问题和展望

本文提出的 VoIP 流量实时识别的方法在准确率和实时性方面都表现出了良好的性能，同时也可以部署到大规模的真实网络环境中。但是其中也存在一些问题，这些问题将作为我们进一步的研究内容：

1）本文实验中采集的数据集不够广泛，只能针对 10 种类别的 VoIP 应用流量进行识别，其次非 VoIP 应用的流量覆盖也不够广泛，对于未知的流量类型并没有一个良好的解决办法。在现在的实时系统中仅仅依靠识别结果的概率分布进行判断。下一步我们需要增加 VoIP 应用和非 VoIP 流量的种类，并且进一步扩大训练数据集，使得 CLNN 模型识别性能更加稳健。

2）使用深度学习进行训练所要求的庞大的数据集导致训练时间过长。当我们尝试向需要识别的 VoIP 流量中新增一种新的类型，当前我们不得不重新训练数据集来获得 CLNN 分类器，这就造成了不必要的训练过程。构建 VoIP 应用种类足够充足的数据集当然是解决此问题的最优方法，但是为了防止扩展 VoIP 类型带来的种种消耗，我们仍需研究新方法解决该问题。

参考文献

[1] 2017 Global Fraud Loss SurveyCFCA, https://zh.scribd.com/document/368471387/2017-Global-Fraud-Loss-Survey-CFCA-pdf.

[2] 信息通信行业防范打击通讯信息诈骗白皮书（2018）, http://www.caict.ac.cn/kxyj/qwfb/bps/201807/P020180711616875493434.pdf.

[3] 2017 年中国反通讯网络诈骗报告, https://gw.alicdn.com/bao/uploaded/LB16jHhkhPI8KJjSspfXXcCFXXa.pdf.

[4] Taliban Using Skype Phones to Dodge MI6, The Daily Mail, https://www.dailymail.co.uk/news/article-1055611/Taliban-using-Skype-phones-dodge-MI6.html.

[5] Lashkars own Skype frazzles Indian intelligence, Times of India, https://timesofindia.indiatimes.com/india/Lashkars-own-Skype-frazzles-Indian-intelligence/articleshow/12934037.cms.

[6] Curtis J, Cleary J, McGregor A, et al. Measurement of voice over IP traffic. In: Passive and Active Measurement conference, 2000.

[7] Chen M, Zhang G-X and Bi J-P. Research of SIP-based VoIP traffic identification methodology. Appl. Res. Comp, 2007, 24.

[8] Sen S, Spatscheck O and Wang D. Accurate, scalable in-network identification of p2p traffic using application signatures. In: Proceedings of the 13th international conference on World Wide Web, 2004: 512 ∼ 521.

[9] Idrees F, Aslam U. A generic technique for Voice over Internet Protocol (VoIP) traffic detection. 2008.

[10] Li B, Jin Z and Ma M. VoIP traffic identification based on host and flow behavior analysis. In: Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, 2010: 1 ∼ 4.

[11] Rathore M, Paul A, Ahmad A, et al. High-speed network traffic analysis: detecting VoIP calls in secure big data streaming. In: 2016 IEEE 41st Conference on Local Computer Networks (LCN), 2016: 595 ∼ 598.

[12] Saqib N A, SHAKEEL Y, Khan M A, et al. An effective empirical approach to VoIP traffic classification. Turkish Journal of Electrical Engineering & Computer Sciences, 2017, 25(2): 888 ∼ 900.

[13] Alshammari R, Zincir-Heywood A N. Identification of VoIP encrypted traffic using a machine learning approach. Journal of King Saud University - Computer and Information Sciences, 2015, 27(1): 77 ∼ 92.

[14] Yildirim T, Radcliffe P J. VoIP traffic classification in IPSec tunnels. In: International Conference on Electronics and Information Engineering, 2010: V1 ∼ 151 ∼ V1 ∼ 157.
[15] Gomes J V, Inacio P R M, Pereira M, et al. Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties. IEEE Transactions on Parallel & Distributed Systems, 2013, 24(10): 2004 ∼ 2014.
[16] Qin T, Wang L, Liu Z, et al. Robust application identification methods for P2P and VoIP traffic classification in backbone networks. Knowledge-Based Systems, 2015, 82: 152 ∼ 162.
[17] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation, 1997, 9(8): 1735 ∼ 1780.
[18] Gers F A, Schmidhuber J and Cummins F. Learning to forget: Continual prediction with LSTM. 1999.
[19] Gers F. Long short-term memory in recurrent neural networks, 2001.
[20] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115 ∼ 133.
[21] Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition[M]. In: Competition and cooperation in neural nets. Springer, 1982: 267 ∼ 285.
[22] Rumelhart D E, Hinton G E and Williams R J. Learning representations by back-propagating errors. Readings in Cognitive Science, 1988, 323(6088): 399 ∼ 421.
[23] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278 ∼ 2324.
[24] Krizhevsky A, Sutskever I and Hinton G E. ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, 2012: 1097 ∼ 1105.
[25] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science, 2014.
[26] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1 ∼ 9.
[27] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition. In: Computer Vision and Pattern Recognition, 2016: 770 ∼ 778.
[28] Real-Time Transport Protocol (RTP) Parameters, https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml.
[29] Sainath T N, Vinyals O, Senior A, et al. Convolutional, long short-term memory, fully connected deep neural networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, 2015: 4580 ∼ 4584.

图清单

表清单

致谢

感谢您使用本模板。

个人简历