中国科学院大学

University of Chinese Academy of Sciences

硕士学位论文

隧道内混合网络行为分割与精细化识别技术研究

作者姓名：赵盼盼

指导教师：苟高鹏 高级工程师

中国科学院信息工程研究所

学位类别：工学硕士

学科专业：网络空间安全

培养单位：中国科学院信息工程研究所

2022 年 6 月

Research of Tunnel Mixed Traffic Behavior Segmentation and Refined Identification Technology

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Science in Engineering

in Cyberspace Security

By

Zhao Panpan

Supervisor: Associate Professor Gou Gaopeng

Institute of Information Engineering, Chinese Academy of Sciences

June, 2022

中国科学院大学
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的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对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或致谢。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

随着隧道使用的增多，网络管理面临着巨大挑战。隧道内应用识别是加强网络管理和提升网络服务质量过程中重要的一步。目前隧道内应用识别的研究主要是基于统计特征的机器学习方法和利用包到达时间和包长度序列的深度学习方法。然而，这些研究主要依赖于单应用假设，即隧道内仅运行单个应用。在真实场景中，隧道内会运行多个应用。由于隧道内不同应用产生的流量具有相同的五元组信息，无法通过明文信息得到每个应用的开始和结束时间。与加密应用流量识别相比，隧道内应用的识别变得更加困难。针对以上问题，本文开展了以下三个方面的工作：

1. 针对缺少隧道内混合网络行为数据集的问题，本文提出了隧道流量回放与隧道混合流量生成算法，生成隧道内包级别标注的隧道混合流量。此外，还利用首尾段分割方法在混合流量上进行了分割。实验使用三种隧道的九个混合数据集验证方法的有效性。结果表明，该方法在三种隧道的九个数据集上的精度、召回和 F1 值均达到 85% 以上，优于已有的先进研究方法。

2. 针对隧道内混合网络行为难以进行精细化识别的问题，本文提出了网络行为转换检测、Burst 分割聚合和端到端的识别方法。它充分考虑了隧道内混合网络行为复杂多样的问题，通过对多种类型的混合流量进行分割，实现隧道混合流量的识别。实验使用四种隧道的 40 个数据集和 20 个公开数据集对方法进行了评估，结果表明分割准确率达到 93%，端到端识别的准确率达到 80% 以上，证明了所提方法对于隧道混合流量识别的有效性。

3. 面向真实场景中隧道混合网络应用识别的需求，构建了基于多级分割的隧道内混合网络应用识别的原型系统。该系统使用 Burst 进行首次分割，通过分割决策模型对不同的分割结果选择性处理。首尾段方法处理两应用混合数据，网络行为转换检测方法进行二次分割，最后端到端模型处理更加复杂的混合流量。通过多级分割处理，该系统可以实现对隧道内混合网络行为的自动化分析。

关键词：隧道技术，混合网络行为流量，网络行为分割，精细化识别

Abstract

With the increasing use of tunnels, network management faces enormous challenges. In-tunnel application identification is an important step in the process of strengthening network management and improving network service quality. The current research on application identification in tunnels is mainly based on machine learning methods based on statistical features and deep learning methods using packet arrival time and packet length sequences. However, these studies mainly rely on the single-application assumption, that is, only a single application runs inside the tunnel. In a real-world scenario, multiple applications are running inside the tunnel. Since the traffic generated by different applications in the tunnel has the same quintuple information, we cannot get the start and end time of each application through the plaintext information. The identification of applications within the tunnel becomes more difficult than the identification of encrypted application traffic. In response to the above problems, this paper has carried out the following three aspects of work:

1. Addresses the lack of a dataset of hybrid network behavior within a tunnel. This paper proposes a tunnel traffic playback and tunnel mixed traffic generation algorithm to generate tunnel mixed traffic marked at the packet level in the tunnel. In addition, the mixed traffic is segmented using the head and tail segment segmentation method. Experiments use nine mixed datasets of three tunnels to verify the effectiveness of the method. The results show that the proposed method achieves over 85% precision, recall and F1 value on nine datasets of three tunnels, outperforming existing state-of-the-art research methods.

2. Aiming at the problem that it is difficult to fine-tune the identification of mixed network behaviors in the tunnel. In this paper, we propose network behavior transition detection, burst segmentation aggregation, and end-to-end recognition methods. It fully considers the complex and diverse behavior of the mixed network in the tunnel, and realizes the identification of the mixed traffic of the tunnel by dividing various types of mixed traffic. Experiments evaluate the method using 40 datasets and 20 public datasets

of four tunnels. The results show that the segmentation accuracy reaches 93%, and the end-to-end identification accuracy reaches more than 80%, which proves the effectiveness of the proposed method for tunnel mixed traffic identification.

3. To meet the requirement of identifying mixed network applications in tunnels in real scenarios, a prototype system for identifying mixed network applications in tunnels based on multi-level segmentation is constructed. The system uses Burst for the first segmentation, and selectively processes different segmentation results through a segmentation decision model. The end-to-end method handles the mixed data of the two applications, the network behavior transition detection method performs secondary segmentation, and finally the end-to-end model handles more complex mixed traffic. Through multi-level segmentation processing, the system enables automated analysis of mixed network behavior within tunnels.

Keywords: Tunnel Technology, Mixed Network Behavior Traffic, Network Nehavior Segmentation, Refined Identification

目 录

图形列表

表格列表

第 1 章　绪论

1.1　研究背景

隧道技术是一种封装技术，即将其他协议产生的网络数据包封装在一个特定协议中，在网络中进行传递 [1]。这种特定的协议被称为隧道协议 [2]。隧道协议将这些其他协议的数据帧或包重新封装在新的包头中发送。新的包头提供了路由信息，从而使封装的负载数据能够通过互联网络传递。隧道的工作过程如图 1.1 所示，首先通过隧道启动节点对数据包进行封装，然后在网络中传输，最后通过隧道终结节点进行解封装，到达目的地 [3]。

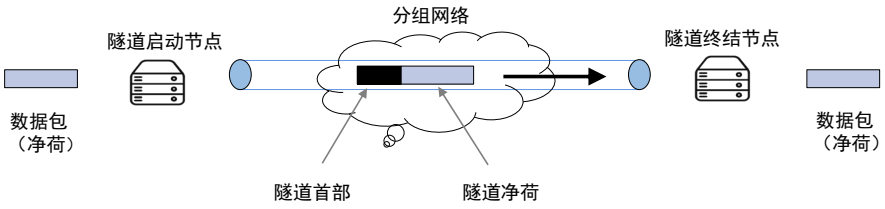


图 1.1 隧道的基本组成

Figure 1.1 Tunnel protocol.

隧道协议按照所处的协议层级不同，可以分为应用层隧道协议、传输层和应用层之间隧道协议、网络层隧道协议和网络接口层隧道协议。在图 1.2 中列出了几种常见的隧道协议。二层隧道协议（Layer 2 Tunneling Protocol，L2TP）、点对点隧道协议（Point to Point Tunneling Protocol，PPTP）它们对应于 TCP/IP 四层模型的网络接口层，它们都是将用户数据封装在点对点协议（Point to Point Protocol，PPP）的帧中进行传输 [4, 5]。互联网安全协议（Internet Protocol Security，IPSec）协议对应于 TCP/IP 四层模型的网络层 [6]。安全套接字协议（Secure Sockets Layer，SSL）工作在传输层和应用层之间，保护应用层的通信安全 [7, 8]。安全外壳协议（Secure Shell，SSH）和 SOCKS v5（SOCKetS）协议工作在应用层。

与其他技术相比，隧道技术具有多项优势。一方面隧道技术具有最小成本的优势，它利用公共网络传输数据，无需购买网络设备和搭建专用线路，节省了成本。另一方面，隧道技术具有安全性的优势。隧道技术通过使用加密机制，保障

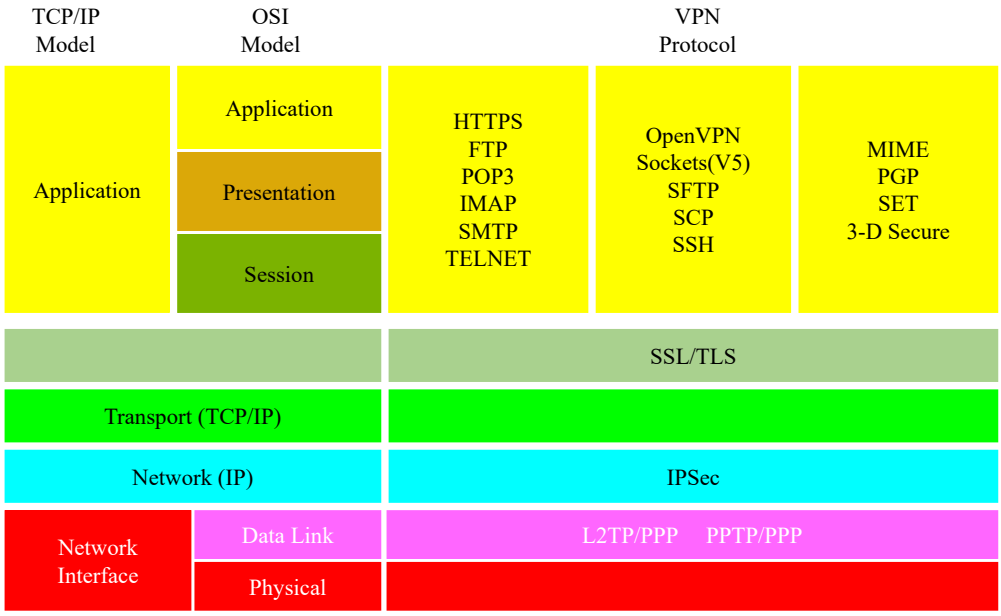


图 1.2 隧道协议

Figure 1.2 Tunnel protocol.

隧道内数据的安全。通过使用隧道技术建立的隧道，能够将网络数据流传送到特定的地址，隐藏私有的网络地址，提供数据安全支持。隧道技术凭借其多项优势，吸引越来越多的人使用隧道技术进行通信。

隧道技术是把双刃剑，在有效保护用户隐私的同时，也给网络安全管理带来严峻的挑战。从用户角度来看，隧道技术凭借其封装和加密特性，使得隧道用户越来越多，隧道内承载的网络应用数量也在不断增加，隧道流量在网络流量中的占比越来越大。2020 年我国互联网网络安全态势综述里面提到 [9]，在新冠疫情防控期间的远程办公需求明显增多，隧道技术成为远程办公人员接入单位网络的主要技术手段之一。从网络安全管理角度来看，隧道技术使得网络安全管理变得更加困难。Gartnet 2020 年报告 [10] 中提到，隧道流量削弱了深度防御的效率，将端点和 DMZ 服务器暴露在出站和入站流量的威胁之下，70% 恶意服务通过隧道技术和加密技术绕过防火墙和入侵检测系统。综上来看，随着隧道流量的增长，互联网的安全管理、资源分配和流量控制面临巨大的挑战。

与加密流量相比，隧道流量经过了封装，隧道内所有的行为具有相同的发送者和接收者，不同网络行为产生的隧道流量具有相同的五元组明文信息（源 IP、源端口、目的 IP、目的端口和协议版本号），因此无法利用五元组明文信息获得

每个行为的开始和结束时间 [11]。与此同时，隧道内网络行为存在各种形式混合和重叠，因此很难提取出纯净的单一网络行为流量。由于无法获知单一网络行为的开始和结束时间，现有的加密流量识别方法，不能直接用于隧道流量分类 [12, 13]。针对隧道内混合的流量，如何设计合理、有效的识别方法给网络安全研究人员带来了新的挑战 [14]。

1.2　研究意义

隧道是利用互联网的基础设施建立网络之间的虚拟链接以传输数据的一种方法。具体来说，隧道是一种封装技术，它将其他协议的数据包重新封装到一个新的报头中并发送它们。新的报头提供了路由信息，因此，封装后的数据可以通过互联网进行传输。一旦数据包到达网络的末端，它们将被解封装并转发到最终目的地。随着互联网的发展，加密网络流量的数量正在快速增长。由于隧道流量的封装和加密特性，它占据了加密流量的很大一部分。越来越多的用户使用隧道技术来保护通信的安全。隧道技术在保证了通信安全，同时也给网络管理带来了挑战。一些恶意用户通过隧道技术绕过防火墙系统，使得网络存在很大的安全问题。因此，隧道流量识别在网络管理和安全方面具有重要意义 [15, 16]。

隧道流量识别对于服务质量保证、网络规划建设和网络异常检测等均具有重要意义 [1, 17]，是进行流量工程、实施 Qos 保障的基础。当前网络安全和隐私保护意识的不断提高，隧道协议应用越来越广泛，隧道流量呈现增长的趋势，隧道流量识别已成为当前网络管理的巨大挑战。基于当前隧道流量识别给网络管理和安全带来的挑战体现在：

1. **隧道流量存在混合和重叠的问题**。隧道具有封装和加密特性，隧道内承载的网络行为都会被封装为相同的协议，具有相同的五元组信息 [18, 19]。隧道内的流量难以利用明文信息直接提取出每个网络行为的流量，隧道流量的识别变得更加困难。而且与加密流量相比，隧道流量存在混合和重叠的问题，隧道内网络行为识别变得更加困难。

2. **隧道流量更加复杂多样**。隧道流量经过封装和加密后，原始流量特征发生了大的变化。由于隧道协议的加密处理方式和封装格式也存在较大差异，使得不同隧道协议产生的流量差距较大，识别特定隧道协议下的应用流量需要采取针对性的识别方法。

3. **亟需进行隧道流量精细化识别**。隧道技术广泛应用使得隧道流量爆发式增长，给流量识别带来新的挑战。细粒度的网络行为管理需要准确的隧道流量识别。识别流量是否经过隧道封装和加密是远远不够的，因为实际网络管理中需要识别加密隧道下的不同应用以及采用隧道传输的应用层协议。因此，为了有效的提升网络管理水平和改善服务质量，需要对隧道流量进行精细化识别。

综上所述，如果没有有效的隧道内网络行为识别方法，则无法分析出封装的网络应用类别。因此，面向隧道内行为的有效识别可以为解决现阶段广泛存在的网络管理、安全管理和网络取证提供技术支撑，具体而言：

1. **网络管理**。网络用户由于自己爱好、需求和观念的不同，在网络上会进行不同的网络行为。但所有的网络用户都有一个相同的需求，那就是享受优质的网络服务 [20]。对于互联网而言，需要根据用户的需求提供个性化的优质网络服务，首先要识别出用户的网络行为，调配好网络资源，提供优质服务。隧道流量将不同流量封装在一起，使得流量之间的差异性减少，隧道内网络行为识别更加困难。因此准确识别出隧道内网络行为，对于网络管理者来说至关重要。

2. **网络取证**。由于隧道技术加密的特性，在保护用户隐私的同时，给犯罪分子提供了技术支持。攻击者可以使一个被防火墙阻挡的协议包在另一个没被防火墙阻挡的协议里，用来巧妙地逃避防火墙规则。而且被封装的攻击流量由于加密的因素被混淆，很容易隐藏其攻击行为。恶意软件也可以通过隧道技术将自己流量伪装在常规协议下，绕过入侵检测系统将机密信息发送外网，对用户的人身及财产安全造成巨大的损失。通过对隧道内网络行为分析，可以对这些攻击者的行为轨迹进行监控，从而对不法份子进行打击 [21, 22]。

3. **安全管理**。隧道具有加密和封装特性，可以很轻松的绕过防火墙和入侵检测系统。随着隧道技术的发展，流量的伪装性和混淆性得到了提高，其自适应能力逐渐增强 [23]。这种复杂的网络环境给网络防护工作造成了很大的困扰。为了保护正常网络用户的上网安全，监控复杂的网络流量，将隧道内的行为准确识别出来，找到可疑的攻击流量，对进行后续流量过滤和网络防御有着重大的作用，有利于维护互联网的网络安全。

1.3　研究内容和创新点

本文的主要研究目标是隧道内混合网络行为分割与精细化识别，其目的在于将不同隧道协议下所承载的混合网络行为进行精细化识别。隧道流量经过了封装和加密，网络行为存在混合的问题。如何在真实操作的条件下识别隧道内网络应用类别或具体应用，目前隧道内混合网络行为识别研究存在三个现状：

1. **缺少隧道内混合网络行为数据集**。关于隧道内混合网络行为识别的研究较少。目前隧道内网络行为识别主要依赖于“单应用假设”，即隧道内仅运行单个应用，而真实场景中隧道内会运行多个应用。目前公开的数据集主要是隧道内单行为的数据集，缺少带标注的隧道内混合网络行为数据集，相应的识别工作也难以开展。而且现有的网络流量标注方法，难以做到包级别的标注，隧道内混合网络行为数据集标注存在困难。

2. **缺少隧道内混合网络行为的有效分割**。隧道流量经过了封装和加密，不同网络行为产生的流量具有相同的五元组信息，无法利用明文信息提取单一网络行为流量。因此，隧道内网络行为存在混合。目前没有提出隧道内混合网络行为流量有效的分割方法，其中混合流的研究主要集中在 Tor 上混合网络行为的识别。Tor 上混合流量和隧道内网络行为流量差别较大，如何从隧道混合网络行为中提取出单一网络行为是一项有挑战性的工作。

3. **缺少隧道内混合网络行为的精细化识别**。在隧道网络行为识别领域，目前主要集中于隧道内单一网络行为的识别，识别的粒度主要集中在隧道检测、隧道内大类行为识别和隧道内单应用识别，对隧道内混合应用的识别研究较少。

针对于隧道混合网络行为识别面临的三个现状，本文分别对其进行深入研究，具体而言：

1. **隧道内混合网络行为数据集构建及行为分割**。针对于隧道内网络行为识别缺少数据集的问题，提出了隧道回放和混合流生成算法产生隧道内混合网络行为数据集。隧道回放方法在两台建立隧道连接的主机之间，模拟客户端和服务端的通讯过程，产生隧道混合流量。混合流生成算法利用隧道内数据包的累计到达时间间隔生成隧道内多种类型的混合网络行为数据集，实现包级别的标注。基于首尾段分割方法对隧道内混合网络行为进行分割识别。首尾段分割方法针对隧道内应用两两混合的问题，利用首段提取出首个应用的流量特征，尾端提取出第二个应用的流量特征。

2. **隧道内混合网络行为的精细化识别**。针对于隧道内网络行为存在混合难以识别的问题，提出了网络行为转换检测、Burst 分割聚合和端到端的识别方法。网络行为转换检测方法首先识别出隧道内网络行为的分割点，其次提取出单一网络行为流量，最后再进行识别。Burst 分割聚合方法首先将混合网络行为流量分割为若干个 Burst, 然后对 Burst 聚合为网络行为段，最后对每个段进行识别，并通过多数表决的方法识别出混合网络行为。端到端的方法对隧道内混合网络行为整体进行处理，充分考虑每部分在时间上的关联性，实现隧道内混合网络行为的识别。

3. **隧道内混合行为精细化识别原型系统**。利用 Burst 切分隧道内混合网络行为流量，通过分割决策模块选择性处理不同类型的 Burst 流量。对于单应用流量，不需要进一步处理，直接进行识别。对于应用两两混合的流量，通过首尾段分割方法进行识别。对于多应用混合的，首先通过网络行为转换检测进行识别，然后网络行为转换检测分割失败的流量，最后再利用端到端的模型进行识别。

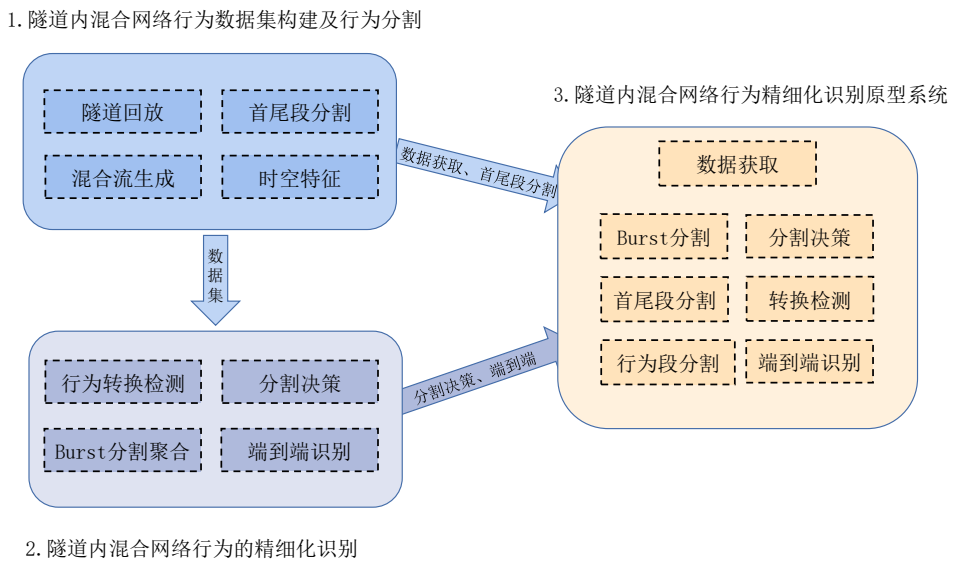


图 1.3 论文组织架构

Figure 1.3 Thesis Organization.

以上三个研究点之间的关系如图所示，研究点 1 为研究点 2 提供混合数据集，研究点 1 和研究点 2 共同组成研究点 3 原型系统的核心部分，其中研究点 1 通过隧道回放和混合流生成方法构建混合数据集，通过首尾段分割进行混合流量识别。研究点 2 通过网络行为转换检测、Burst 分割聚合、分割决策和端到端

识别方法进行混合流识别。研究点 3 基于首尾段分割、分割决策、端到端等技术实现隧道内混合网络行为的精细化识别。

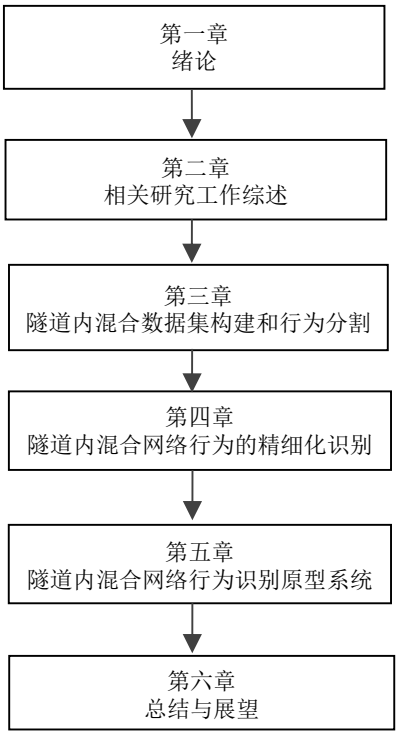


图 1.4 论文组织架构

Figure 1.4 Thesis Organization.

1.4　论文组织结构

围绕上述研究目标和内容，本论文共分为 6 章，具体组织结构如图 1.4 所示。

第一章，绪论。首先，本章介绍了本文的研究背景和意义，阐述了识别隧道内网络行为的重要性。随后，分析了该研究目前存在的三个难点及其对应的研究方案，并对本文的研究内容进行了介绍。最后，介绍了本文整体的组织架构。

第二章，相关研究工作综述。本章从隧道内网络行为识别和 Tor 上混合网络行为识别两方面阐述了目前国内外研究进展情况。首先，针对于隧道内网络行为识别，阐述了针对于隧道内网络行为识别方面相关技术概念以及介绍了隧道检测、隧道内大类行为识别和隧道内应用识别的典型研究。其次，针对于隧道内混合网络行为识别技术，由于目前没有专门针对于隧道内混合网络行为识别技术的研究，本章主要介绍了 Tor 上混合网络行为识别方法，包括基于寻找分割点、

基于网络行为段和端到端的识别方法。最后对本章内容进行了小结。

第三章，隧道内混合数据集构建和行为分割。本章从产生带标注的隧道内混合网络行为数据集出发，提出了隧道回放方法和混合流生成的两种方法。首先，隧道回放方法将加密网络行为流量回放进隧道产生隧道内混合网络行为流量。混合流生成方法将隧道内单网络行为流量按照数据包累计到达时间产生隧道内混合网络行为流量，然后利用首尾段方法对隧道内混合网络行为进行分割识别。最后对本章内容进行了小结。

第四章，隧道内混合网络行为的精细化识别。针对隧道内混合网络行为，提出了网络行为转换检测和网络行为端相结合的分割决策、基于端到端模型的两种方法，这些方法对隧道内混合网络行为进行分割识别。在前一章生成的混合数据集上对方法进行了验证，最后对本章各种方法进行了小结。

第五章，隧道内混合网络行为识别原型系统。该章考虑到真实操作中存在单隧道多行为的情况，从而导致隧道内长流无法分割的问题。首先提出了 Burst 分割的方法，并结合以上所有研究点提出了隧道内混合网络行为原型系统的组织架构并测试。最后对本章进行了小结。

第六章，总结和展望。此章对本文的研究内容和工作做出了总结，对下一步工作进行了展望。

第 2 章　相关研究工作综述

随着隐私泄露事件的频发，网络用户开始重视对隐私信息的保护。隧道技术凭借其封装和加密特性，受到越来越多人的青睐。隧道技术被广泛应用于保护通信的安全，隧道流量占比越来越大。隧道流量的增长使得网络安全管理和服务质量提供面临严峻挑战。隧道内网络行为识别技术的目的在于识别出隧道内不同的网络行为，识别粒度更以具体的任务来确定，例如按照隧道协议、应用大类行为和产生流量的具体应用等属性来划分，最终实现隧道协议检测、大类应用识别和应用识别。

目前混合流的相关研究主要集中在 Tor 上。因此，本文主要从隧道内网络行为识别和 Tor 上混合网络行为识别两个方面展开了综述。首先按照识别粒度，从隧道检测、隧道内大类行为识别和隧道内应用识别三个方面阐述了隧道内网络行为识别相关研究的现状。然后基于分割识别技术，从基于寻找分割点、基于网络行为端分割和基于端到端识别三个技术层次阐述了 Tor 上混合网络行为识别相关研究 [24]。最后，对本章内容进行了总结。

2.1　隧道内网络行为识别技术

2.1.1　隧道内网络行为识别技术概述

隧道内网络行为是指识别隧道内承载的网络行为类别，并根据任务要求，识别不同的粒度。隧道内网络行为识别按照识别粒度的不同，可以划分为隧道检测、隧道内大类行为识别和隧道内应用识别 [25, 26]。隧道检测是指识别隧道流量使用的协议，例如 SSL、SSH、IPSec 等。隧道内大类行为识别是指识别隧道内进行的大类行为，例如 Video、Browsing、Chat 等 [27, 28]。隧道内应用识别是指识别隧道内运行的应用，例如 Skype、Twitter、Youtube 等。

隧道内网络行为识别的思想是从隧道流量中提取有效特征，然后使用分类器算法构建模型，进行隧道流量的识别。隧道检测是指利用不同隧道协议在通信过程中的差异性，找出其中有明显区分性的规律和特征，实现隧道检测。由于不同隧道协议差别较大，通常检测过程相对比较简单，识别准确率较高。隧道内大类行为和隧道内应用识别是指对同一隧道协议下的不同网络行为进行识别。隧

道协议将不同网络行为进行了封装和加密，流量原始特征发生了大的变化，而且同一隧道内的不同网络行为经过封装和加密后，彼此之间差异性减小。因此隧道内大类行为和隧道内应用识别极具挑战性。

2.1.2 隧道检测技术

隧道检测是从加密网络流量中识别出哪种隧道协议的流量，是进行隧道内网络行为分析的第一步。目前隧道的检测主要集中于 SSL 隧道、HTTP 隧道、HTTPS 隧道等的检测。

Maccarthy 等 [29] 在不使用 IP 地址、端口号或数据包有效负载的情况下，基于数据流持续时间和数据包长度等统计特征，在 AdaBoost 集成学习算法、C4.5 决策树算法和朴素贝叶斯模型上进行了测试。实验结果表明 AdaBoost 模型识别效果最好，SSL 隧道的识别率达到 96%。Pang 等 [30] 选择每个包中的五位操作码作为流量指纹，对五种类型的数据包进行分类，并以一个通信的前 10 个数据包作为流量特征识别早期的 OpenVPN 隧道流量，识别准确率为 99.98%。He 等 [31] 利用域名字符串特征和长度统计特征，使用多个机器学习算法在自采集数据集上进行了测试，其中随机森林算法分类效果最好，DNS 隧道识别率达到 99%。

Liu 等 [32] 利用时间间隔特征、请求数据包大小特征、记录类型特征、子域熵特征等多维特征，利用 SVM 分类器。实验使用自采集的数据集多模型进行了评估。结果表明该模型对于 DNS 隧道的识别率达到 99.96%。Lin 等人 [33] 的研究提出了数据包长度、数据包时间间隔统计特征和随机森林分类器的方法。实验中在自采集的数据集上对模型进行了评估。实验结果表明，该方法实现了 DNS 隧道、HTTP 隧道和 HTTPS 隧道的有效检测。

Cheng 等 [34] 提出了一种 SSL 隧道利用每条流的前 N 个数据包，使用编码器和注意力机制方法，在公开数据集 ISCX2016 上识别率达到 95% 以上。Davis 等 [35] 提出了一种基于机器学习算法的 HTTP 隧道和 DNS 隧道的检测方法。该方法利用请求的数量、URL 熵等 24 维特征，使用 SVM 算法作为分类器。实验结果表明，该方法在自采集的数据集上识别结果为 99%。Montazeri 等 [36] 在自收集的 DOH 隧道数据集上提取出了时间序列特征、负载特征（加密套件列表等）、统计特征等，并在随机森林、支持向量机、C4.5 决策树算法、二维卷积神经网络模型上进行了测试。结果表明，随机森林算法取得最好的识别效果。

余等 [37] 提出一种基于自动特征工程与压缩感知相结合的网络隧道检测方

法。通过自动特征工程挖掘出更深层次的网络隧道特征。同时通过压缩感知算法在不损失高维特征精度的基础上实现降维。最后在大规模真实数据集上进行了测试，结果表明 DNS 隧道的 F-measure 值达到 95%, HTTP 隧道的 F-measure 值达到 82% 以上。Ishikura 等 [38] 出了一种基于缓存属性感知特性的 DNS 隧道检测方法，所提出的 DNS 缓存服务器上的缓存命中率 CHR^n、访问命中率 AHR^n 和访问错过计数 AMC^t 可以有效地表征 DNS 隧道流量。在此特征上引入基于规则的过滤器和基于递归神经网络过滤器实现了更高的 DNS 隧道攻击检测率。

综上所述，目前隧道内检测已经取得很好的识别效果，隧道检测已经不是问题。总体看来，隧道检测使用的特征还主要是手动提取的，使用的分类器还主要是一些机器学习算法，其中随机森林算法、支持向量机算法用的最多。此外，每个研究中使用数据集也主要是各自收集的。

2.1.3　隧道内大类行为识别技术

隧道内大类行为识别是从隧道流量中识别出不同大类行为的流量，这些大类型为包括：浏览、电子邮件、聊天（使用即时消息应用程序时产生的流量）、Stream(使用多媒体程序例如 Youtube 和 Vimeo 时产生的流量)、文件传输（发送或接收文件和文档的流量）、VoIP（语音应用程序生成的流量，例如 Skype 等）等。隧道内大类行为识别是在隧道检测的基础上更进一步分析了隧道内的行为。与隧道检测相比，隧道内大类行为识别是更精细化的识别。

Leroux 等 [39] 提出了一种隧道内单大类行为识别识别的框架。该框架利用包长、包时间间隔等统计特征，使用随机森林作为分类器。实验中利用自采集的数据集对该框架进行了评估，结果表明可以达到 97% 的准确率，识别的大类行为包括 VoIP、Browsing、Bittorrent 等。Shapira 等 [40] 将隧道单网络行为流量转化为二维图片，然后利用卷积神经网络进行识别。在 ISCX VPN-nonVPN 公开数据集上对该模型进行了测试，识别的大类行为包括 Chat、Browsing、Video 等, 识别结果可以达到 99.7%。

Li 等 [41] 在 ISCX VPN-nonVPN 数据集上提出了基于深度学习的隧道内大类行为识别框架。该框架使用端口、协议、流持续时间、包长度、每秒包数等 41 维特征，在卷积神经网络和循环神经网络上面进行了识别，识别的大类行为包括：Chat、File、Email、Audio 和 Video 等。卷积神经网络识别准确率高于循环神经网络，取得最好的识别效果。Lin 等 [42] 使用每条流的前 N 个包，每个包的

前 M 个字节作为特征，利用卷积神经网络和双向 LSTM 作为分类器。在 ISCX VPN-nonVPN 数据集上对模型进行了评估。实现结果表明识别结果达到 92.7%。

Guo 等 [43] 利用卷积自动编码 (CAE) 和卷积神经网络 (CNN)，将流量样本预处理为会话图片，然后进行识别。基于 CAE 的方法，利用 CAE 的无监督性质来提取隐藏层特征，可以自动学习原始输入和期望输出之间的非线性关系。基于 CNN 的方法在提取图像的二维局部特征方面表现良好。结果表明，基于 CNN 的模型在六大类中识别准确率达到 92.92%。

Cui 等 [44] 提出了一种新的基于会话数据包的加密网络流量分类模型。该方法使用胶囊神经网络 (CapsNet)，引入了两次分割机制来稀释干扰流量，增加有效流量的权重。然后利用 CapsNet 学习加密流量的空间特征，并通过 softmax 分类器输出加密流量分类的结果。在公开的 ISCX VPN-nonVPN 数据集上的服务和应用方面评估了所提出的加密流量分类模型。实验结果表明，CapsNet 的性能优于最先进的加密流量分类方法。

Chen 等 [45] 采集了 5 种隧道下的 7 大类行为的混合流量。提出了利用时间窗口提取单行为流量，然后利用卷积神经网络进行识别。该方法仅简单的利用滑动窗口对混合流量进行处理，当网络行为流量差别较大时，效果很差，而且该方法无法对提取到的混合行为流量进行处理。

目前隧道内大类型为的识别，主要是针对公开数据集上的单行为流量，研究的重点主要放在隧道内单网络行为的识别上，只有一篇论文提到了对混合大类行为进行流量的处理，方法简单，使用受限。

2.1.4 隧道内应用识别技术

隧道内应用的识别是在隧道内大类行为识别的基础上进一步的识别，主要是从流量层面识别隧道内承载了哪些应用。与隧道内大类行为识别相比，识别的粒度更细。

Zhang 等 [46] 提出了一种隧道内单应用识别的框架。该框架使用数据流序列作为输入，使用一维卷积神经网络进行识别。在公开数据集 ISCX VPN-nonVPN 对该框架进行了评估，识别的应用主要包括 Skype、Twitter、Youtube、Vimeo 等，识别结果达到 97%。Yao 等 [47] 利用循环神经网络 (RNN) 对时间序列网络流量进行建模，引入注意力机制，使用注意力辅助长短期记忆（LSTM）和分层注意

力网络（HAN）对隧道流量进行分类。最后，在 ISCX VPN-NonVPN 数据集对方法进行了评估。实验结果表明所提出的方法达到了 91.2% 的准确率。

Zheng 等 [48] 提出了一种应用元学习方法来进行隧道内单应用识别：基于流的关系网络 (RBNN), 该方法从原始流中学习代表性特征，然后在统一的框架中对它们进行分类。最后，在真实世界的网络流量数据集上验证了 RBRN 的有效性，实验结果表明 RBRN 可以实现出色的分类性能，并且在加密流量分类方面优于最先进的方法。Sun 等 [49] 提出了一种从流量结构和流量流数据中学习特征表示的新方法, 利用两层图卷积网络（GCN）架构进行流特征提取和加密流量分类, 使用自动编码器来学习流数据本身的表示，并将其集成到 GCN 学习的表示中。在 ISCX VPN-nonVPN 和 UTSC-TFC2016 两个公开数据集上均取得很好的识别效果。

He 等 [50] 提出基于卷积神经网络实现对隧道内应用的识别。该分类模型通过结合卷积层和池化层深度挖掘和学习 SSH 隧道内不同应用的流量特征。在其自收集数据集上对模型进行了评估。结果表明，该模型对 SSH 隧道内的 11 种应用程序识别准确率达到 98% 以上。korczynski 等 [51] 提出了在安全套接字层/传输层安全 (SSL/TLS) 会话中传输的应用流量的随机指纹。指纹是基于一阶齐次马尔可夫链，该方法对于隧道内应用具有很好的识别结果。

Tian 等 [52] 提出了一个卷积注意力网络（CAT）。在 CAT 中，首先通过网络流量的注意机制实现了不同字节的重要性。然后，卷积神经网络（CNN）用于自动学习特征并将输出送到 softmax 函数以获得分类结果。它使 CAT 能够从网络流量数据中学习到足够的信息，并确保分类的准确性。该模型在在公共加密网络流量数据集 ISCX2016 上取得了很好的识别结果。

Chen 等 [45] 采集了 SSH、SSL、IPSec、L2TP 和 PPTP 隧道下的单网络行为数据集。实验中使用 1000 维包长序列作为输入，利用卷积神经网络提取有效特征，通过使用 arcface 损失函数增加加性角度间隔, 在角度空间上最大化分类界限，提高类内可分性同时增加类内紧度和类间差异。通过该方法实现了 5 种隧道下 20 多种应用的识别，识别结果均在 85% 以上。

隧道内网络行为识别如表 2.1 所示，目前隧道内应用的识别主要集中单应用的识别，关于隧道内混合网络行为识别的研究较少，且方法简单，分割效果较差。

表 2.1 隧道内网络行为识别国内外研究现状总结

Table 2.1 Summary of research status for network behavior recognition in tunnels

识别粒度	代表论文	数据集	特征	模型
隧道检测	Cheng,et al.(2020)	ISCX2016	每条流的前 N 个数据包	编码器 注意力机制
	Lin H,et al.(2019)	自采集（单）	包长度、 时间间隔统计特征	随机森林
	Montazeri et al.(2020)	自采集 (单)	时间序列特征 负载特征等	随机森林
大类行为识别	Leroux S,et al.(2018)	ISCX2016（单）	包长度序列 包时间间隔序列	随机森林
	Li Y,et al.(2021)	ISCX2016 (单)	包长度序列	二维卷积神经网络
	Chen.(2021)	自采集 (混合)	1000 维包长度序列	一维卷积神经网络 Arcface 损失
	Lin K,et al.(2021)	ISCX2016 (单)	前 N 个包 前 M 个字节	卷积神经网络 双向 LSTM
应用识别	Zhang J,et al.(2019)	ISCX2016（单）	数据包长度 时间序列	一维卷积神经网络
	Yao H,et al.(2019)	ISCX2016 (单)	包长序列	分层注意力网络
	Zheng W,et al.(2020)	ISCX2016 (单)	原始流序列	自编码器 元学习

2.2 Tor 上混合网络行为识别技术

2.2.1 Tor 上混合网络行为识别技术概述

Tor 是一种匿名通信网络，旨在通过提供隐藏通信内容和元数据的手段，来保护用户免受恶意网站和网络窃听者的攻击。Tor 通过三跳电路路由连接，并使用洋葱路由对通信内容进行分层加密，这样所有的中继器都不能同时知道通信的起点和目的地。

Tor 上网络行为识别是指识别出 Tor 上承载的网站类别，掌握 Tor 上进行何种网络行为。Tor 上网络行为识别可以按照不同网站产生的流量和同一网站的不同页面产生的流量来划分 [53]。按照不同网站产生的流量可以分为 Google.com、YouTube.com、Facebook.com 等; 从同一网站内不同网页产生的流量来分，以

Youtube.com 为例，可以分为首页、探索、shorts 等。Tor 上用户倾向于有多个打开的选项卡或窗口，这允许他们同时加载几个页面。因此，Tor 上的网络行为存在混合。Tor 上混合网络行为识别按照方法不同可以分为基于寻找分割点的方法、基于网络行为段分割的方法和基于端到端识别的方法。

2.2.2　基于寻找分割点的方法

基于寻找分割点的方法是指寻找两个行为之间的分割点，根据分割点进行分割。通过分割将不同的网络行为分割开来，使得每段都是单个网络行为，对单一网络行为进行识别。该方法是将隧道内混合网络行为识别分为两个阶段，首先是分割阶段，其次是识别阶段。

Juarez 等 [54] 的研究第一次提出了单网页假设不成立，并用实验验证了这个问题。结果表明当网页存在混合的时候，网页识别效果急剧下降。Wang 等 [55] 提出了两级分割架构，首先基于时间的分割算法，当两个相邻数据包空闲时间间隔超过阈值则进行分割。对于低于时间阈值没有成功分割的混合流量，再通过分割点发现算法进行二次分割。分割点发现算法利用数据包时间间隔和数据包方向等 23 维特征，通过寻找 15 个最近的相邻数据包来对每个候选单元格进行评分：来自“正确分割”类的数据包增加分数，而来自“错误分割”类的相邻数据包降低分数，得分最高的候选单元格是真正的分割点。

Xu 等 [56] 首次提出了分割点检测中存在数据不均衡问题，并提出 Balace-Cascade 算法解决了不均衡问题，然后选择了往返时间（RTT)、文档长度和传入数据包的大小、传出数据包的数目、Burst 大小和数量等特征，最后利用 Xgboost 算法识别分割点。实验收集了两标签的混合流量，并对模型进行了评估。实验结果显示，该方法在 Tor 上平均识别 TPR 为 64.94%。

Cui 等 [57] 提出使用隐马尔科夫模型识别分割点的方法。该方法充分利用混合数据包在时间维度上的关联性，将隐马尔可夫模型应用于网络流量跟踪，得到每个网络数据包属于每个类（网站）的概率（概率矩阵）。然后从左到右遍历每个数据包作为一个分割点，将流量分为两个部分，并在两个部分中测量每个网站的概率。依次计算每部分最大的预测概率，如果两部分预测概率都是最大，则是真实分割点。实验在自采集混合数据集上对该方法进行了评估。结果表明该方法分割点准确率达到 80%, 首个网站的识别准确率达到 70%，第二个网站的识别准确率达到 69%。

基于分割点检测方法的优点是有效的分割了连续网站，缺点是分割点检测耗费了大量的时间和精力，重叠网站的识别效果较差。因此，如何在保证准确率的情况下，减少寻找分割点的时间，是使用该方法需要考虑的问题。

2.2.3 基于网络行为段分割的方法

基于段分割的 Tor 混合行为识别，它的主要思想将混合流划分为若干个网络行为段（segemnt），然后对所有网络行为段进行预测，通过网络行为段的多数投票表决，得到整条流的预测结果。

Cui 等 [57] 提出按照数据包和时间均匀划分为若干个 Section。然后对每个 Section 进行识别，最终通过多个 Section 的识别结果决定整个混合流量的识别结果。实验在自采集数据集上进行了测试，结果表明基于时间的 Section 划分识别效果最好，第一个网站的识别率达到 78%, 第二个网站的准确率为 65%。

Gu 等 [58] 主要的想法在于，用户在打开第一页后，由于思考时间而会延迟访问第二页。实验分析了在延迟中传输的匿名流量，并选择了细粒度的特征来识别第一页。此外，排除了第一页的流量，并利用粗特征来识别第二页。在真实环境中采集了混合数据集，并对贝叶斯模型进行了评估。实验结果表明当延迟设置为 2 秒时，该方法识别第一页的准确率为 75.9%，第二页的准确率为 40.5%。

基于段分割方法的优点是不用花费太多时间寻找分割点，避免了重叠部分对流量识别的影响。缺点是段分割是不精确的分割，存在分割不彻底的段，部分段内可能存在混合流量。而且段的大小对结果影响较大，段太大可能包含混合流量，存在混合段的干扰。段太小使得可利用信息太少，识别准确率下降。因此如何选择合适的段，也是使用该方法面临的一个挑战。

2.2.4 基于端到端的识别方法

基于端到端的识别方法，将混合流量的分割与识别交给同一个模型来完成，实现隧道内混合网络行为识别的研究。

Guan 等 [59] 提出了一种名为 BAPM 的块注意分析模型作为一种新的多标签攻击模型。BAPM 充分利用了包括重叠区域在内的整个多标签数据包跟踪，以避免信息丢失。它从方向序列生成一个选项卡感知的表示，并执行块分割，以尽可能清晰地分离混合页面选项卡，从而缓解信息混淆。然后使用基于注意力的分析方法对属于同一页面选项卡的数据块进行分组，最后在一个全局视图下同时

识别多个网站。实验在手动合成的混合数据集和真实环境的数据集上对该模型进行了评估。结果表明即使在更大的重叠区域下，BAPM 也优于其他识别方法。

表 2.2 Tor 上混合网络行为识别技术总结

Table 2.2 Summary of Mixed Network Behavior Recognition Technology on Tor

技术	论文	数据集	方法
分割点寻找	Wang T,et al.(2016)	Tor 混合数据集 (120 个网页，每个网页 40 个实例）	提出两级分割架构：基于时间进行首次分割，然后利用分割发现算法进行二次分割
	Xu Y,et al.(2018)	Tor 混合数据集（50 个网页，每个网页 50 个实例）	提出 BalaceCascade 算法解决不均衡问题，然后利用 Burst 大小等多维特征，使用 Xgboost 进行识别。
	Cui W, et al.(2019)	Tor 混合数据集（100 个网站，每个网站 40 个实例）	将隐马尔可夫模型应用于混合流量数据包预测
网络行为段分割	Cui W,et al.(2019)	Tor 混合数据集 (100 个网站，每个网站 40 个实例）	利用时间信息和数据包数目将混合流均匀分割，逐一识别
	Panchenko A,et al.(2016)	Tor 混合数据集	粗粒度特征识别第一个页面，细粒度特征识别第二个页面
端到端识别	Guan Z,et al.(2021)	Tor 混合数据集（50 个网站，每个网站 90 个实例）	它从方向序列生成一个选项卡感知的表示，并执行块分割，以尽可能清晰地分离混合页面选项卡，从而缓解信息混淆

2.3　本章小结

目前隧道内混合网络行为识别的研究较少，因此本章从隧道内网络行为识别和 Tor 上混合网络行为识别两个方面对相关研究工作进行了综述。首先对隧道内网络行为进行了介绍，按照隧道内识别粒度的不同，划分为隧道检测、隧道内大类行为识别和隧道内应用识别三个方面。绝大多数研究都是针对隧道内单网络行为的识别，没有充分考虑到真实环境中隧道流量存在混合的问题。一些考虑到隧道内存在混合网络行为的研究，也只是简单的进行分割，对于混合的网

络行为流量没有进一步的处理。Tor 上混合网络行为识别研究按照分割技术划分为：基于分割点寻找的方法、基于网络行为段分割的方法和基于端到端的识别方法。通过对 Tor 上混合网络行为识别研究的分析，为后续隧道内混合网络行为识别的研究提供了很大的帮助。

第 3 章　隧道内混合数据集构建和行为分割

本章从构建隧道内混合网络行为数据集出发，提出了隧道回放、隧道内混合流量生成算法的两种方法。首先详细介绍了隧道回放、隧道内混合流生成算法的原理，其次介绍了构建的数据集类型及大小。然后从隧道内应用识别角度出发，提出了 TunnelScanner 方法用于隧道内混合网络行为识别，详细介绍了该框架原理及流程并且使用三种隧道内混合数据对该模型进行测试。最后对本章做了小结。

3.1　引言

在真实环境中，隧道内网络行为存在混合，混合的形式复杂多样。由于隧道内网络行为的不同，隧道流量的混合形式也差别较大。为此，需要对隧道内不同类型的混合流量进行逐一分析。在真实环境中收集的流量，不能对隧道内不同类型的混合流量区分开、不能有效标注隧道内网络行为转换点、不能获得隧道内混合网络行为的重叠率。真实环境中收集的隧道流量无法对模型进行有效评估，因

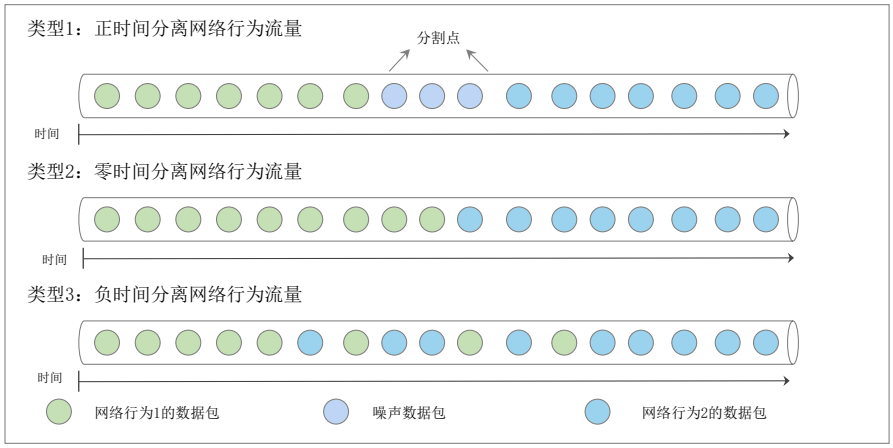


图 3.1 隧道内三种类型的混合数据

Figure 3.1 Three types of mixed data in the tunnel.

此需要生成隧道内混合网络行为数据集。生成的隧道内混合网络行为数据集要能够区分不同类型的混合流量，以便于能够针对不同混合类型的数据进行有效分析（例如，有明显分割边界的隧道混合流量和没有明显分割边界的混合流量，哪种识别效果更好）。生成的隧道内混合网络行为数据集要能够标注隧道内网络

行为转换点，以方便寻找分割点的方法能够有效的进行评估。生成的隧道内混合网络行为数据集要能够控制重叠率，以便于对不同重叠率的混合流量进行有效分析（例如，随着重叠率的增加，模型是否稳定）。因此，如何生成满足上述三个要求的隧道内混合数据集，是本章重要考虑的问题。

3.2 隧道内混合网络行为数据集构建

3.2.1 隧道内混合网络行为数据集概述

本次研究的场景是单用户使用隧道的场景，单个用户在隧道内使用多个应用的网络行为。本章节构建的数据集包括多种行为的混合，隧道内混合网络行为数据集的类型如图 3.1 所示：正时间分离网络行为流量、零时间分离网络行为流量和负时间分离网络行为流量。正时间分离网络行为流量是指两个网络行为之间有时间间隔。零时间分离网络行为流量是指两个网络行为之间时间间隔为 0。负时间网络行为流量是指两个网络行为之间存在重叠，前一个网络行为还未结束，后一个网络行为就已经出现。

3.2.2 基于隧道回放的混合数据集构建

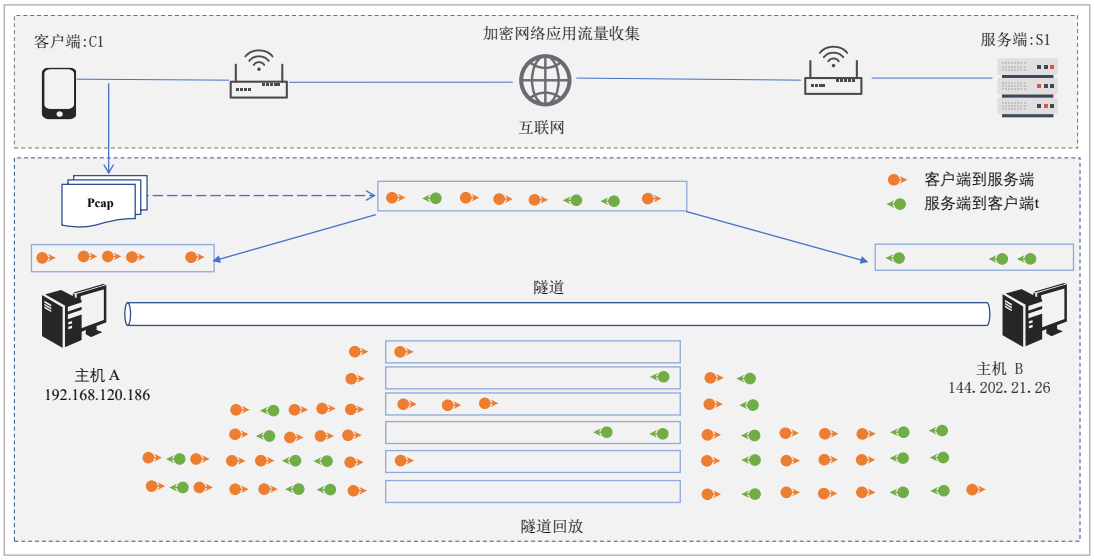


图 3.2 在隧道中回放应用程序数据

Figure 3.2 Play back application data in a tunnel.

3.2.2.1　加密网络行为数据集收集

由于移动流量涉及到用户的隐私数据，因此公开移动流量数据集目前不可用。因此，现有的工作使用自收集的数据集来验证所提出的方法。首先，使用黑域软件在后台关闭不相关的应用程序 (黑域是一个免费的超级实用的优化工具。它不需要根权限就可以完美地阻止应用程序启动、在后台运行和在后台醒来)。然后使用 adb-monkey 随机点击应用程序（尽可能模拟真实用户的使用过程），最后使用 tcpdump 来捕获应用程序流量。

用户在上网过程中会产生各种各样的网络行为流量，在实验中选取了几种常用的网络应用程序。这些应用程序按照大类可以分为七类。在实验室中，收集了七种网络大类的 30 个应用程序的流量。这七种类型主要是即时消息应用程序、文件传输应用程序、电子邮件客户端应用程序、电子邮件 web 应用程序、媒体应用程序、社交应用程序和 VoIP 应用程序。这些数据将用于隧道回放模块。

3.2.2.2　隧道回放混合数据集构建

现有的隧道流量收集方法存在一些不足，可伸缩性较差。首先，现有的隧道流量采集方法难以解决隧道混合流量的包级别标注问题。其次，隧道内不同网络行为的流量具有相同的五元组明文信息，因此很难过滤掉无关背景流量。在应用流量中掺杂大量背景流量，会对应用的识别带来干扰，影响识别的精度。此外，每次在收集移动应用程序流量时，需要获得移动设备的根权限，导致可扩展性较差。

为了克服上述缺点，提出了一种利用隧道回放来生成隧道流量的方法。隧道回放是指在主机的两端模拟应用程序的客户端和服务器。图 3.2 详细介绍了隧道回放的过程。在回放过程中，采用等待机制和超时重传机制，确保包发送有序，减少包丢失。

隧道回放过程如下。首先，在主机 A 和主机 B 之间建立一个隧道。主机 A 模拟应用程序的客户端，主机 B 模拟应用程序的服务器。然后，主机 A 按顺序读取 Pcap 文件的 C2S（客户端到服务器）方向上的数据包，并通过隧道将它们发送到主机 B。接下来，主机 B 按顺序读取 Pcap 文件，当接收到 C2S 方向的数据包时，将 S2C（服务器到客户端）方向的数据包发送给主机 A。两个主机按顺序读取发送数据包。最后，Pcap 中的数据包被完全发送到隧道中。在隧道回放

期间，使用 Tcpdump 来捕获主机 A 上的隧道流量。

隧道混合流量的生成取决于应用流量收集模块收集的应用流量。在实验中，收集了 IPSec、SSL 等隧道中的单应用标签流量和多应用标签流量。应用主要包括 twitter、skype、dropbox、facebook 等。对于单标签应用程序，通过隧道回放为 30 个应用程序收集了 7 种类型的隧道流量，每个类包含 100 个实例。对于多应用程序标签数据集，收集了三种类型的隧道流量。通过 pcap 分割、随机（不同应用程序流量）和有序（相同应用程序流量）混合、pcap 合并、隧道回放等操作，生成了隧道中正时间分离应用流量混合流量。在这节使用应用程序流量收集模块收集的 30 个应用程序流量生成了三种类型的混合流量。

3.2.3 基于生成算法的混合数据集构建

3.2.3.1 基于生成算法的隧道内混合数据集的构建

算法 1 隧道内混合应用流量生成算法

输入: 数据包长序列 L, l、数据到达时间序列 T, t, T_{noise}

输出: 混合数据包长度序列 L_{mix}, 混合数据包时间序列 T_{mix}

procedure MTG(L, l, T, t, T_{noise}, R) ▷ 包长序列、包时间序列
  if R less than 0 **then** ▷ 正时间分离应用流量
    $L_{mix} \leftarrow \{L, l\}$
    $T_{mix} \leftarrow \{T, T_{noise}, t\}$
  else if R equals 0 **then**
    $L_{mix} \leftarrow \{L, l\}$
    $T_{mix} \leftarrow \{T, t\}$ ▷ 零时间分离应用流量
  else
    $L_{mix} \leftarrow \{L_0, ..., l_0, L_{n-m}, ..., L_n, ..., l_k\}$
    $T_{mix} \leftarrow \{T_0, ..., t_0, T_{n-m}, ..., T_n, ...t_k\}$ ▷ 负时间分离应用流量
  end if
  return L_{mix}, T_{mix} ▷ 混合包长度和时间序列
end procedure

混合流量生成模块是在真实的隧道中模拟并生成混合应用程序流量。现在，学术界和工业界还没有隧道混合网络行为的公开数据集。隧道内混合网络行为数据集存在难以进行包级别标注的问题。针对隧道混合流量标记的问题。为了解决隧道回放的不足，提出了一种生成隧道混合流量的算法，如算法 1 中所述。

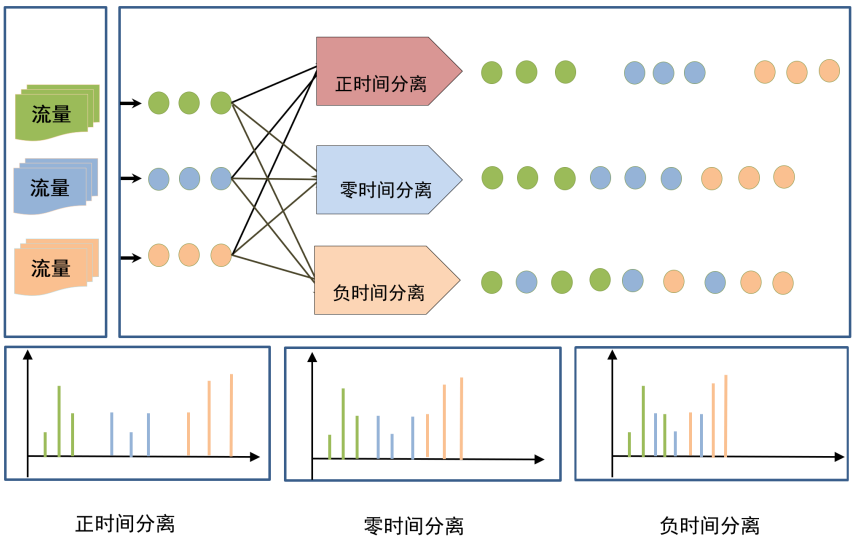


图 3.3 混合流生成方法

Figure 3.3 Mixed traffic generation method.

如算法 1 所述，L_{mix} 表示一个由来自两个混合应用程序的数据包跟踪组成的数据集 (例如，Twitter)。输入参数为包长度序列 L，l、包时间间隔序列 T, t、两个混合应用标签 a_1, a_2 和参数 R。输出为混合包长度序列 L_{mix}、包时间间隔序列 T_{mix} 和包标签序列标签 mix。算法 1 根据重叠率参数 R 生成不同类型的混合流量。如果 R 小于 0，|R| 表示正时间分离应用的时间间隔。第二个应用程序的第一个时间间隔是 |R|。该算法忽略了两种应用在切换期间隧道噪声的影响，将混合应用的包长度序列直接拼接成混合数据包长度序列。将零时间分离应用程序的包长度序列和包时间序列直接拼接成一个混合流量序列。负时间分离应用流量按照重叠率确定重叠部分，然后按照累计到达时间间隔生成隧道混合流量。

表 3.1 自采集数据集

Table 3.1 Self-collected datasets.

数据集名称	隧道层次	隧道名称	应用类别	流数	混合类型	数据集数目	大小
l2tp-data	链路层	L2TP	30	4500	3	10	23.6GB
ipsec-data	网络层	IPSec	30	4360	3	10	22.3GB
ssl-data	应用层	SSL	30	5100	3	10	36.3GB
ssh-data		SSH	30	5700	3	10	35.6GB

利用隧道回放方法生成正时间分离应用流量，使用混合流量生成算法生成

零时间和负时间分离应用流量，构建了隧道内应用混合的数据集和应用内行为混合的数据集。首先是自采集数据集上应用混合：共采集了 IPSec、SSL、L2TP 和 SSH 四种隧道内 30 个应用的流量，利用隧道回放和混合流生成算法，最终每种隧道下得到正时间分离、零时间分离和负时间分离三种类型的混合流量。负时间分离类型的数据集包括重叠率为 5%，10%，15%，20%，25%，30%，35%，40% 的 8 个数据集。

表 3.2 公开数据集

Table 3.2 public datasets.

数据集名称	数据集来源	数据集名称	类型	网站类别	混合流数	数据集数目	混合类型
tor14-data	USENIX2014	knndata	Tor	60	18000	10	3
tor17-data	USENIX2017	walkiebatch	Tor	100	30000	10	3

其次，利用混合流生成算法（MTG 算法）还在公开数据集上产生了混合流量：USENIX2014、USENIX2017 两个公开数据集。每个公开数据集包括三种类型的混合数据集：正时间分离应用流量、零时间分离应用流量和负时间分离应用流量。其中负时间分离类型的数据集包括重叠率为 5%，10%，15%，20%，25%，30%，35%，40% 的 8 个数据集。

3.3 基于首尾段分割的隧道内混合网络行为识别

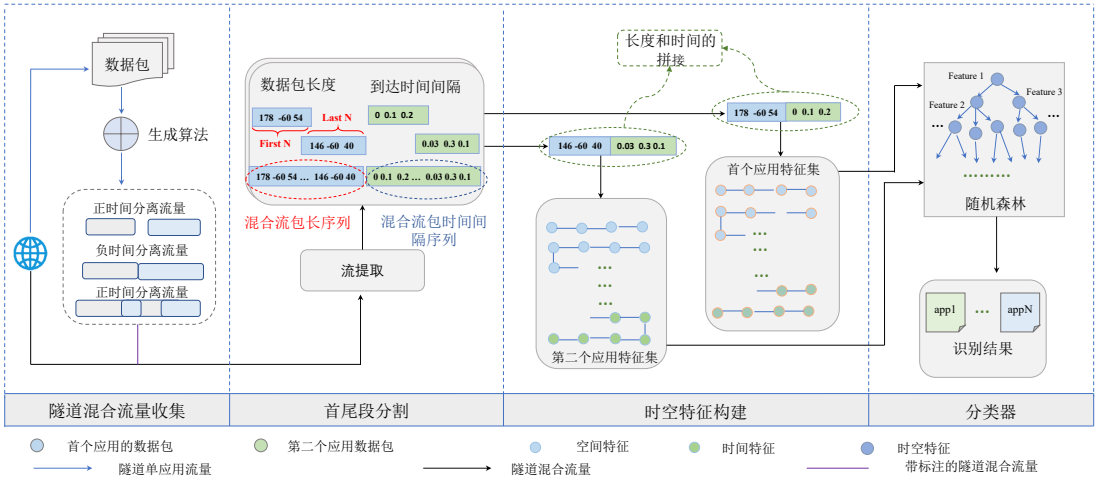


图 3.4 TunnelScanner 框架图

Figure 3.4 The framework of TunnelScanner.

在生成的隧道混合数据集上，本节提出基于首尾段分割的隧道内混合网络行为识别框架，其具体处理过程如图 3.4 所示。其处理思想是，对于隧道内混合网络行为，首先提取出单网络行为，然后对单网络行为流量进行识别。TunnelScanner 框架主要包括首尾段分割模块、时空特征构建模块和分类器模块。首尾段分割模块从隧道混合网络行为流量里面提取出单网络行为流量。时空特征构建模块从提取的单网络行为流量提取时间和空间特征。分类器模块对提取到的特征进行识别。

3.3.1　首尾段分割模块

在混合流量生成模块中生成了三种类型的混合流量，并将在此模块中进行分割。理论上，如果混合流量没有被分割，最终的分类器对混合流量没有完全的预测能力。因此，混合流量需要先进行分割，才能进入特征提取模块。

在目前的相关研究中，首先需要找到混合流的分割点，然后对分割出的单网络行为流量进行识别。然而以前的方法有两个缺点。首先，寻找分割点会增加识别的延迟时间。其次，分割点的分割精度直接影响应用程序的识别结果。

在 TunnelScanner 方法中，不再花费太多的时间来寻找分割点。相反，通过在混合流量前后取若干个网络数据包 N 来进行流量识别。这些 N 个数据包被称为网络行为段。首先，提取了混合流量的包长度序列和包时间间隔序列。然后选择混合流量前后的 N 个数据包，构建首个和第二个应用的指纹。使用 N 个数据包的包长度和包时间间隔作为特征提取模块的输入。对于段大小为 N，希望尽可能满足以下条件：

1. N 的值应该尽可能大，使得段内单个应用程序包含足够的信息。

2. N 的值应该尽可能小，以便只包括少数应用程序。理想情况下，只包含一个应用程序。

很容易看出上述两个条件是矛盾的：N 的值越小，它包含的应用信息越少。N 的值越大，可能包含的应用数目就越多。在实验模块中，将展示如何选择 N 的值。

3.3.2　时空特征构建模块

隧道内网络行为在时间维度上存在混合，因此时间信息对于隧道内混合网络行为识别至关重要，时间维度的差异体现出隧道内不同网络行为的混合程度。

隧道内网络行为的差异性在空间表现出数据包长度的变化，空间信息有助于隧道内网络行为的识别。在提取混合流量前后的 N 个数据包后，从数据段中提取特征。其中提取了一个名为 MaxCum 的特征，包括空间和时间的统计特征。

空间特征主要包括三个维度数据包的统计特征：上行数据包、下行数据包、双向数据包。每个方向都包括：最大值、最小值、均值、方差、标准差、偏度、峰度、百分位数等 31 维的统计特征。时间特征主要是数据包时间间隔的统计特征，也包括 31 维的统计特征。从原始流量中提取了 124 维的统计特征作为候选特征。在实验中，将介绍如何平衡好精度和训练时间，选择合适的特征。

3.3.3 分类器模块

时空特征构建模块提取出了 MaxCum 特征。在分类器模块，使用的是随机森林算法。随机森林是一种集成学习算法，属于 Bagging 算法。通过结合多个弱分类器，对最终结果进行投票或平均，从而使整个模型的结果具有较高的精度和泛化性能。

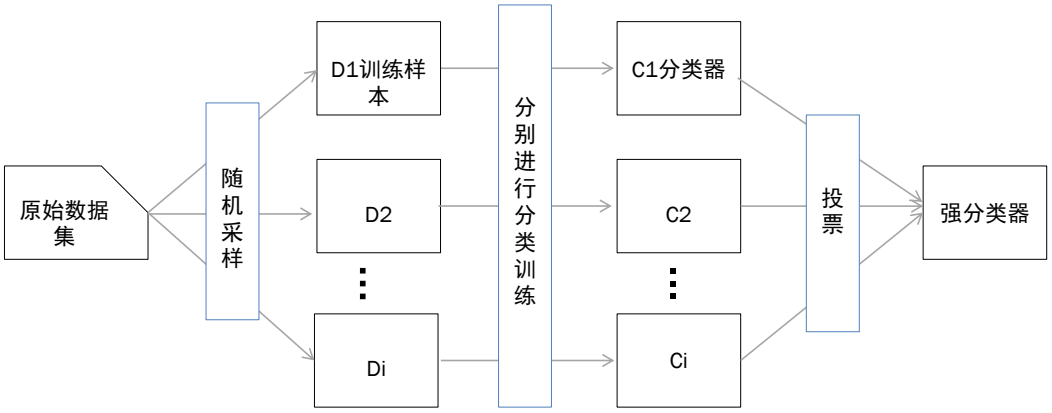


图 3.5 随机森林框架图

Figure 3.5 The framework of RF.

随机森林算法工作如图 3.5 所示，在原始数据集上通过有放回抽样重新选出 M 个新训练集来训练分类器。它使用训练出来的多个分类器集合对新样本进行分类，然后多数投票方法统计所有分类器的分类结果，出现结果最高的即为最终标签。随机森林采用基尼系数进行特征选择，如 3.1 公式所示：

$$Gini(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{3.1}$$

K 是类别总数，p_k 是样本属于第 k 类的概率。

3.4　实验评估

3.4.1　基于首尾段分割的识别结果

3.4.1.1　实验设置

实验环境。实验环境的操作系统为 Ubuntu 16.06TLS，CPU 型号为 Intel Xeon E5-2640，GPU 型号为 GTX 1080Ti，内存大小为 94GB，采用的编程语言为 Python3.8 语言，使用 Keras 框架搭建模型。

数据集。数据集是第三章节产生的 SSL、SSH 和 L2TP 隧道下混合应用流量，主要包括 ssl-data、ssh-data 和 l2tp-data。该数据集包括正时间分离应用流量、零时间分离应用流量和负时间分离应用流量。

比较方法。为了证明提出的特征 MaxCum 可以更详细地表征数据以及提出的混合流量分割方法有效，与 AppScanner、Cumul、Section-Time 和 Section-Length 方法进行了比较：

- Appscanner: 由 Taylor[60] 在 EuroS & P 提出的移动加密应用流量识别框架。它利用随机森林模型，使用 54 维数据包长度的统计特征，对于 Google Play 商店中 110 个最受欢迎的应用程序识别达到 99% 准确率。
- Cumul: 由 Panchenko[61] 在 NDSS 上提出的加密流量识别方法，该方法使用累计数据包长度特征。在二分类和多分类中可以达到 96% 准确率。
- Section-Time 和 Section-Length：由 Cui[57] 在 AsiaCCS 提出的 Tor 混合网络行为识别框架。它利用时间维度信息和数据包数目信息来分割混合流量，将混合流量分为许多个数据段。默认认为每个段内仅包括单网络行为，然后对网络数据段进行识别。最后通过多个数据段的投票表决得出混合流量类别。

评估方法。实验中对模型和特征集的评价方法是 10 倍交叉验证方法。通过将识别标签与真实标签进行比较，得到了 n 类的 i 类的真阳性 (TP)、假阳性 (FP)、真阴性 (TN) 和假阴性 (FN)。为了评价该方法的性能，其计算结果为：

$$Accuracy = \frac{\sum_{i=1}^{n}(TP_i + TN_i)}{\sum_{i=1}^{n}(TP_i + FP_i + TN_i + FN_i)} \tag{3.2}$$

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i} \tag{3.3}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i} \tag{3.4}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.5}$$

3.4.1.2 特征选择

在时空特征构建模块中，选择了常用的 124 维特征作为候选特征。在本节中，为了从 124 维的统计特征中选择有效的特征，进行了一个实验，随机选择段大小为 50。最后，利用随机森林算法对这些特征进行了有效性验证。

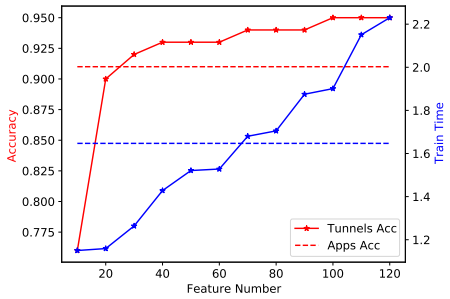


图 3.6 特征数量对分类结果的影响

Figure 3.6 Impact of the number of features on the classification results.

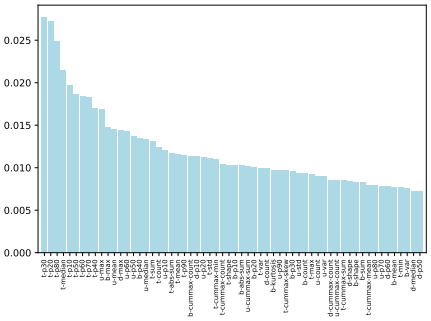


图 3.7 前 60 个特征重要性得分

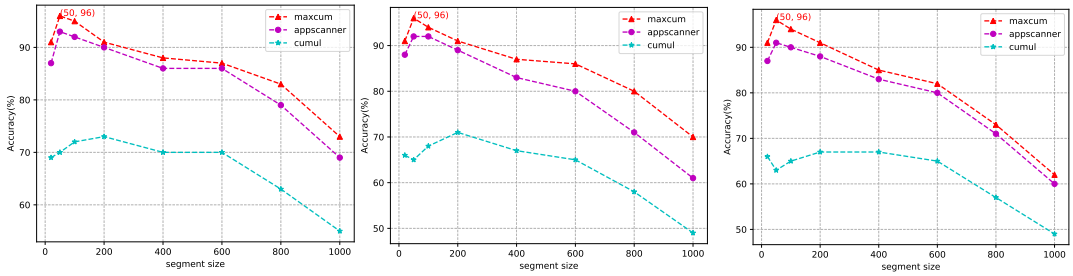
Figure 3.7 Top 60 feature importance scores.

在实验中，首先使用随机森林算法来衡量 124 维统计特征的重要性，并对衡量结果按降序排序。在特征重要性指数的降序中，每次添加 10 个特征来训练模型。具体结果分别如图 3.6、图 3.7 所示。

如图 3.6 所示，当选择前 30-60 个特征时，训练精度高于 AppScanner，训练开销低于 AppScanner。基于精度和训练时间两个方面考虑，选择前 60 个特征作

为 MaxCum。前 60 个特征的详细显示如图 3.7 所示。到达时间间隔的统计特征具有较高的重要性得分，这说明了时间信息的重要性。

3.4.1.3　段长度的影响



(a) 正时间分离　　**(b) 零时间分离**　　**(c) 负时间分离 (10%)**

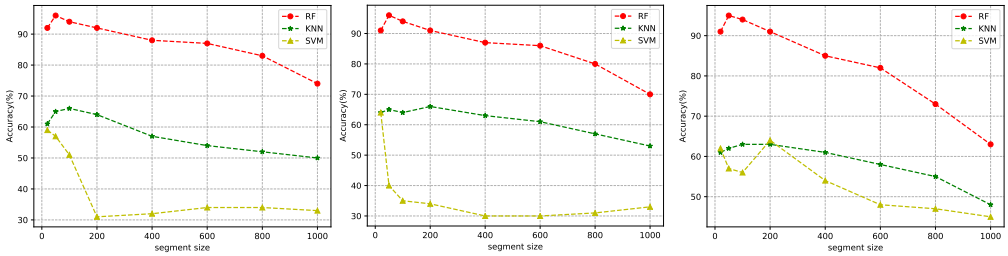
图 3.8 分段大小对 SSL 隧道中分类精度的影响

Figure 3.8 Impact of segment size on classification accuracy in ssl tunnel.

在进行隧道混合流量识别之前，将分析不同长度的网络行为端对识别结果的影响。在本节中，进行了一个实验来查看不同的段大小对识别精度的影响。在实验中，分别使用了 MaxCum、AppScanner 和 Cummul 特征，并选择随机森林作为分类器。此时，MaxCum 特征包括 60 维的统计特征。网络行为段的大小在 20 到 1000 之间选择。

图 3.8 显示了所提出的特征与 AppScanner 和 Cummul 两种特征的识别准确率的比较。根据结果发现，很明显，提出的特征在三种情况下都优于其他特征。隧道中不同应用的混合主要是由于隧道中不同应用的使用时间的重叠。因此，数据包到达时间间隔是一个非常重要的特征信息，这也是 MaxCum 表现良好识别性能的原因之一。

如图 3.8 所示，可以看出，不同网络行为段长度下的识别结果有很大的差异。当网络行为段较小时，准确率低的主要原因是信息太少，无法有效的刻画应用程序。当网络行为段太大时，低精度原因是，该网络行为段可能包括混合应用程序的流量，存在混合流量的干扰。当网络行为段大小为 50 个数据包时，识别精度达到峰值，识别精度在三种情况下都取得了很好的效果。因此，在该模块选择 50 个数据包作为网络行为段的大小。



(a) 正时间分离　(b) 零时间分离　(c) 负时间分离 (10%)

图 3.9 不同分类器中 maxcum 特征的分类结果

Figure 3.9 Classification results of maxcum feature in different classifiers.

3.4.1.4 分类器选择

为了确定选择哪个分类器算法进行隧道流量识别。实验中，使用提取的 MaxCum 特征，在加密流量识别领域常用的三种机器学习分类器上面进行了测试，即 K 近邻（KNN，k=10）、支持向量机（SVM）和随机森林（RF）。

图 3.9 显示了三种分类器算法的识别结果。随机森林在三种数据集上均表现最好。这是因为随机森林由多个决策树组成，最终的预测结果由所有决策树投票决定。SVM 在这三个分类器中性能最差。其原因是 SVM 不适合进行大规模的数据处理。当数据样本相对较小时，SVM 更容易掌握数据与特征之间的非线性关系。根据分类器选择实验结果，选择随机森林作为分类器。

3.4.1.5 比较实验

为了测量 TunnelScanner 的识别效果，在三种隧道中的九种数据集进行了测试，并与其他方法进行了比较。比较结果如图 3.10 所示，可以得出以下结论：

- TunnelScanner 方法不仅在 SSL 隧道上取得了很好的性能，而且在 SSH 和 L2TP 上也表现出了很好的性能。在三个隧道的九种数据集上，TunnelScanner 的精度、召回和 F1 均在 85% 以上。实验结果证明了 TunnelScanner 方法的稳定性和通用性。
- TunnelScanner 达到了最好的性能，并且优于所有的识别方法。TunnelScanner 略优于 AppScanner，主要是因为该方法使用了累积最大值和数据包时间的统计特征。隧道内混合流量主要在时间维度上存在混合和重叠，因此时间维度特征有助于表征混合流量的混合情况，对于混合流量的识别非常重要。累计最大值方法对时间和空间维度信息进行累计叠加，可以有效表征单应用网络行为，有助

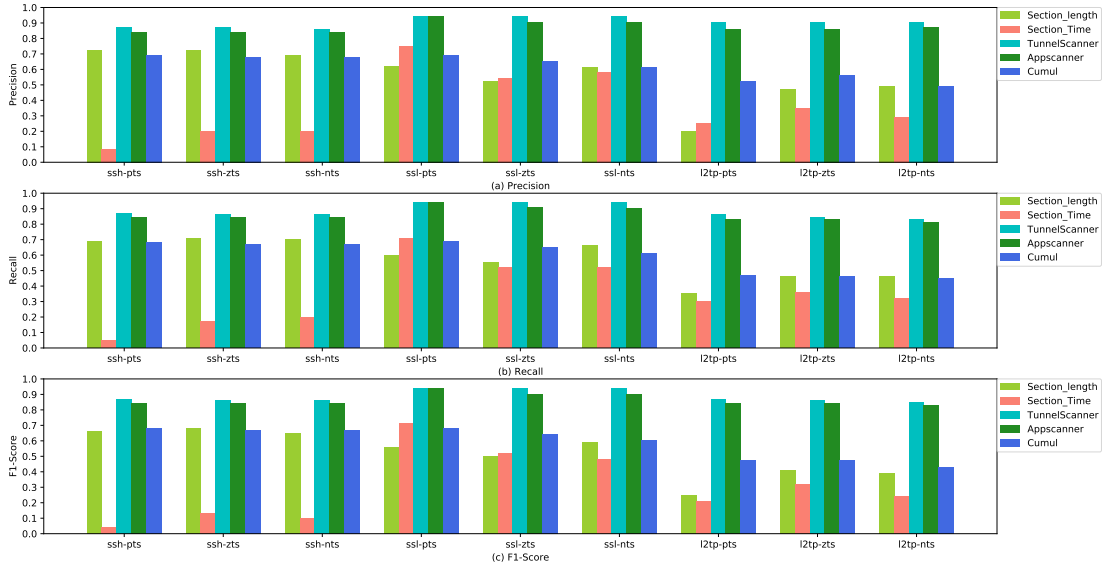


图 3.10 三个隧道不同方法的比较结果

Figure 3.10 Comparison results among different methods in three tunnels.

于应用的识别。结果进一步说明了 TunnelScanner 所使用的 MaxCum 优于其他特征，表明了时间维度和空间维度信息在混合流量识别领域的重要性。

- TunnelScanner 方法比 Section-Time 方法和 Section-Length 方法可以更好地分割混合流量。Section-Time、Section-Length 和 TunnelScanner 都以包长度和数据包时间间隔序列为输入，TunnelScanner 的性能优于其他两种方法。Section 只能用于序列长度差异不大的混合流量识别。当长流和短流混合时，识别效果相对较差。TunnelScanner 在混合流量之前和之后使用若干个数据包来分割流量，它的优点是避免了长流对短流的影响。

3.4.1.6　不同重叠率下识别结果

表 3.3 TunnelScanner 在三种隧道的分类结果

Table 3.3 Classification results of TunnelScanner on three tunnel datasets.

隧道协议	重叠率						
	0.05	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**
SSL	0.93	0.90	0.90	0.86	0.83	0.82	0.78
SSH	0.88	0.84	0.83	0.81	0.77	0.75	0.70
L2TP	0.90	0.89	0.85	0.85	0.82	0.81	0.76

在负时间分离应用流量数据集上对模型进行了评估，选择网络行为端为 50，

识别结果如表所示。如表 3.3 所示，同一隧道不同重叠率下的识别结果略有差异，实验结果表明，随着重叠率的增加，识别结果略有下降。重叠率的增加导致结果下降，主要原因在于，首尾段里面掺杂了混合区域的流量，对识别产生影响，导致准确率低。同一重叠率不同隧道的识别结果也略有差异，SSL 隧道混合应用识别结果高于 SSH 和 L2TP。产生这种的原因主要在于隧道自身建立差异产生的影响。整体看来，在三种隧道混合数据集上识别结果均在 70% 以上，实现了有效的识别。

3.5 本章小结

本章介绍了隧道内混合网络行为数据集构建构建方法，主要包括基于隧道回放的方法和基于混合流生成的算法。首先介绍了基于隧道回放的方法，该方法包括加密网络行为数据集收集、隧道回放混合数据集构建两个部分。其次介绍了基于生成算法的方法，该方法利用数据包累计时间间隔信息生成三种类型的隧道混合流量数据。它充分考虑了多种网络行为在时间维度上的混合和重叠，更加接近真实环境中产生的隧道混合流量数据集。最后提出了基于首尾段分割的隧道混合网络行为识别模型 TunnelScanner，识别隧道混合网络行为。随后使用 3 种不同类型隧道数据集，对模型进行了测试。证明该框架在不同的隧道混合流量数据集上的识别结果均保持一个良好的水平，说明了基于首尾段分割的思想在隧道内混合流量数据集领域的一个可用性，因此可以有效降低隧道内混合网络行为流量造成的高误分类问题。

第 4 章　隧道内混合网络行为的精细化识别

本章从识别隧道内混合应用类别组成角度出发，提出了隧道内混合应用流量精细化识别模型。首先介绍了分割决策和基于端到端识别的两种方法。其次，使用第 3 章产生的隧道混合流量数据集，对各种方法进行了测试，并与现有的识别方法做了对比，证明了两种方法对隧道内混合应用流量识别的可行性。最后对本章做了小结。

4.1　引言

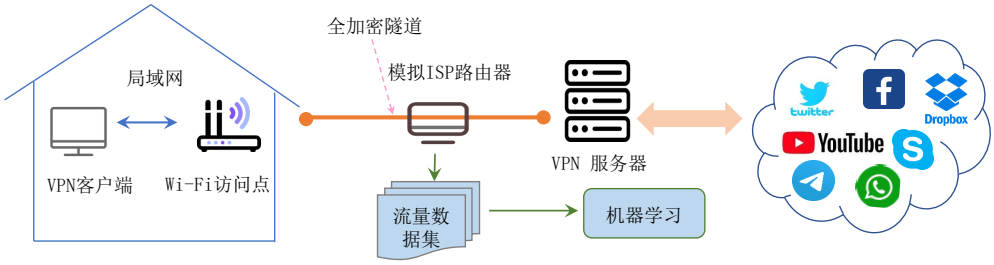


图 4.1 一个客户端使用加密隧道生成的单条流

Figure 4.1 Single Stream from one client using an encrypted tunnel.

隧道内网络行为识别的研究主要从流量层面实现网络防护和管理。现有关于隧道内网络行为识别的方法主要分为基于机器学习的方法和基于深度学习的方法。但是，这些方法主要集中隧道检测、隧道内大类识别，识别粒度较粗，也主要集中于隧道内单网络行为识别。然而在实际场景中，网络行为在时间维度上存在交叉和重叠，使得隧道内产生的网络流量存在混合和重叠。此外，隧道的封装和加密特性使得隧道流量具有相同的五元组信息，无法利用明文信息（五元组信息）提取出隧道内单网络行为，流量识别变得更加困难。而且隧道内网络行为复杂，产生的流量差别较大。如何对隧道内混合网络行为进行有效识别，是本章考虑的重点。

4.2　基于分割决策的隧道内混合网络行为识别

对于隧道内混合网络行为，提出了 TMT-RF 的两级分割框架如图 4.2 所示。该框架包括三部分：网络行为转换检测模块、Burst 分割聚合模块与特征提取和

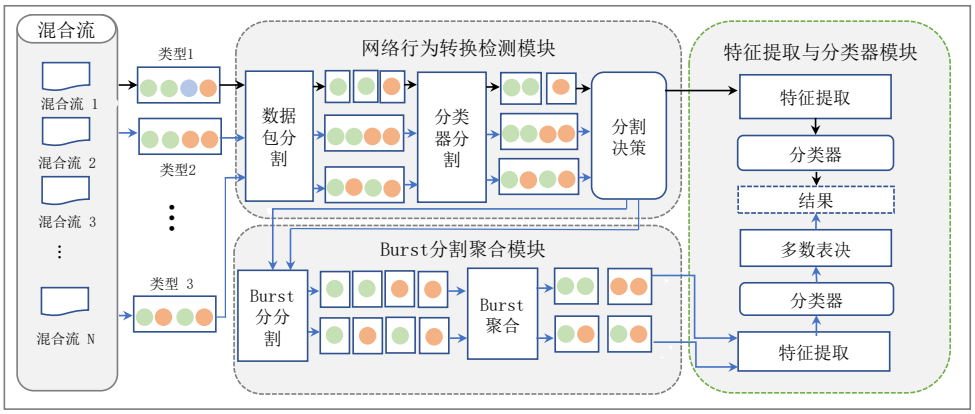


图 4.2 TMT-RF 框架图

Figure 4.2 The framework of TMT-RF.

分类器模块。网络行为转换检测模块利用数据包分割、分类器分割和分割决策进行首次分割，Burst 分割聚合模块进行二次分割，提取出单网络行为流量。特征提取和分类器模块从单网络行为流量提取特征，进行识别。

4.2.1 网络行为转换检测模块

网络行为转换检测模块有三个步骤，如图 4.2 所示。两个应用程序的正时间间隔都伴随着隧道噪声，即 IPSec 隧道中的心跳数据包。这些数据包是网络行为发生转换的标志，可以通过识别噪声数据包，实现网络行为转换的检测。因此，混合流量的分割是基于这样一个规则，即如果多个连续的数据包被识别为隧道噪声，则混合隧道流量将被划分为两个部分。由于隧道噪声流量的包长度是非常可区分的，为了使其高效和快速，该模块使用包长度阈值方法进行首次分割。

基于网络数据包长度阈值的分割，一些低于该阈值的应用流量也可以被识别为隧道噪声，从而导致多次切割。本节使用机器学习技术来识别在第一部分中过滤掉的噪声，并决定是否分割。基于数据包的分割和基于分类器的分割产生多个不同的结果后，分割决策对不同的结果选择不同的处理方式。基于数据包的分割将产生三种类型的结果，包括混合流量被正确分割、单个应用程序流量被分割以及多应用程序混合流量未被完全分割。

基于数据包的分割。第一种混合流量可以通过隧道噪声进行分割。隧道噪声的包长度分布与所应用的包长度分布有很大的不同。隧道噪声的数据包长度分布在一个较小的值范围内。因此，可以通过数据包长度特征来识别隧道噪声，实

现对隧道混合流量的分割。L_{split} 是用于识别隧道噪声的数据包长度阈值。其中选择的 L_{split} 是为了寻求最大限度地提高隧道噪声识别的准确性。如果 L_{split} 的值不合适，它可能会导致混合流的多次分割和没有被完全分割。本节希望任何错分的结果都尽可能少。因此，必须选择一个合适的值。本节在实验中将展示如何选择 L_{split}，以及它如何影响分割的准确性。

基于分类器的分割。基于数据包长度阈值的分割方法会产生三种类型的分割结果。第一种类型的分割结果是混合流量被正确地分割。这个结果不需要进一步处理，可以直接输入到分类器中进行识别。因此，本节在分割实验中，希望尽可能多的产生第一种类型的分割结果。理想情况下，希望全是第一种类型的分割结果。第二种类型的分割结果是，单个应用流量被过度分割了。当数据包长度阈值不合适时，就会发生这种情况。隧道内混合流量很有可能被分割成多个部分。基于分类器的方法是为了减少第二种类型分割结果的出现。第三种类型的分割结果是，多应用的混合流量没有被完全分割开。当两个应用之间有重叠时，就会发生这种情况。基于分类器的方法并没有试图解决这个问题，分割失败的混合流量将在 Burst 分割聚合模块中进一步的处理。

基于分类器的分割思想不仅旨在进一步提高隧道噪声识别的精度，而且大大提高了通过分类器进行分割的准确率。随机森林是一种由决策树组成的集成学习算法。它是 Bagging 的一种扩展变体。随机森林的随机性主要体现在每棵树的随机训练样本中，属性的选择是随机的。随机森林模型简单而高效，且开销较低。

对于基于分类器的分割，利用数据包长度序列的 54 维统计特征，选择了随机森林算法构建分类器。分割准确率可以用以下公式来衡量。

$$diff(P,Q)=\begin{cases}1 & Q-R\le P\le Q+R\\ 0 & else\end{cases} \tag{4.1}$$

P 是预测分割点，Q 是实际分割点，R 是误差范围。

$$SA=\frac{1}{N}\sum_{i=1}^{N}diff(P_{predict}(i),P_{true}(i)) \tag{4.2}$$

其中 $P_{true(i)}$ 为第 i 个样本的真实分割点的网络数据包序列号，$P_{predict(i)}$ 为第 i 个样本的预测分割点的网络数据包序列号，N 为样本总数。

分割决策。在前两次分割之后，可能会产生三种类型的分割结果。有必要识别不同的结果，以便进一步处理。在分割决策中，使用 K 近邻算法（KNN）作为分类器，这是一个监督学习算法。KNN 根据公式 4.3 的距离测量方法得到 k 个最近邻。在 KNN 分类中，输出的是一个类的成员关系。一个对象由其邻居的多数投票进行分类，该对象被分配给其 k 个最近邻居中最常见的类。

$$L(x_i, x_j) = (\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^2)^{\frac{1}{2}} \tag{4.3}$$

分割决策以单个应用流量和多应用混合流量作为输入，并使用 KNN 进行二分类。如果输入是多应用的混合流量，则该预测为真（1）。否则，输出为假（0）。如果输出结果为真，则需要由 Burst 分割聚合模块进一步处理，然后发送到分类器。如果输出结果为假，表明分割成功，则可以直接发送到分类器进行识别。

4.2.2 Burst 分割聚合模块

网络行为转换检测模块可以很好地分割第一种类型的混合流量，但它不能分割第二种和第三种类型的混合流量。Burst 分割聚合模块用于解决这些类型的流量。第三种类型的混合流量存在重叠。重叠率是重叠部分中数据包总数与整个数据流中包总数的比值，如下所示。

$$OR(T) = \frac{T_{overlap}}{T} \tag{4.4}$$

其中，$T_{overlap}$ 是流量的重叠部分，T 是整个流量。重叠部分对于流量识别没有帮助，相反，还会给流量识别带来干扰。之前的思想是花时间寻找分割点，然后提取出单一网络行为流量，输入分类器进行识别。然而在 Burst 分割聚合模块不再花费大部分时间寻找分割点，而是将重叠的流量分成许多段，并对每个段进行预测。其中 Burst 是所有一起出现的网络包的一组，它满足最新来的网络数据包与其相邻的前一个网络数据包具有相同方向的条件。从客户端到服务端的数据包被标记为正号。从服务端到客户端的数据包被标记为负号。

Burst 分割。数据包的粒度太细，而数据流的粒度则太粗糙。数据流可以根据特定的任务被划分为 Burst，Burst 按照划分方式不同，分为方向 Burst 和时间 Burst。这里使用的是方向 Burst（在本章简称 Burst），一个 Burst 是由在同一方向上的几个连续的数据包组成的。重叠隧道流量根据 Burst 被分成几个部分。不同

应用程序的重叠流的长度不同，产生的 Burst 的数量也可能不同。一个 Burst 可以包含一个、两个或多个数据包。

Burst 聚合。按照 Burst 分割模块分割处理后，将产生多个 Burst。如果简单的根据 Burst 进行分分割，然后进行识别，就会存在以下两个问题：

1. **表示能力有限**。Burst 中包含的数据包太少，描述应用程序的能力也很有限。这样使得可以利用的信息太少，会导致识别结果不高。

2. **延迟时间增加**。如果按方向 Burst 进行分割，就会产生大量的 Burst。每个 Burst 都需要进行识别，这将增加分类器的延迟时间和工作负担。

为了解决由 Burst 分割引起的上述两个问题，将若干个连续的 Burst 聚合成一个段（segment）来进行分割。如果一个段包含太多的 Burst，它将减少延迟时间。同时，该段可能包含两个以上的应用数据包，也会导致分割精度下降。如果一个段包含太少的 Burst，它将不会减少延迟时间，但是包含太少的应用信息会导致识别结果下降。因此，希望选择一个尽可能满足以下条件的合适值 $N_{combine}$：

- 每个数据段都应该包含尽可能少的应用程序，以减少段内存在混合流量情况的发生。理想情况下，每个段只包含一个应用程序。
- 每个段都应该尽可能为相应的应用程序包含尽可能多的信息，为应用识别提供更多维度信息，提高识别结果。同时减少分类器的识别压力。

很容易看出，上述两个条件是相互矛盾的：数据段越短，它包含的信息越少，使得应用识别结果下降。数据段越长，就越容易包含多个应用程序，使得段内存在混合流量。在实验中将展示如何选择 $N_{combine}$ 以及它如何影响分类的准确性。

4.2.3　特征提取和分类器模块

特征提取。特征提取包括从每个部分提取的 54 个统计特征 (使用 Appscanner 中的特征)。对于每个段，将考虑三个方向的数据包，包括仅传入数据包、仅传出数据包和双向数据包。双向数据包由输入和输出的数据包组成。对于每个方向（共 3 个），计算以下值，包括最小值、最大值、平均值、中值绝对偏差、标准差、方差、偏态度、峰度、百分位数（从 10% 到 90%）和该系列中的元素数（共 18 个）。

分类器。该框架使用随机森林模型作为分类器。本章节随机森林模型的首次使用是在网络行为转换检测模块的分类器分割部分。该模块使用的随机森林模

型与前面的区别在于：基于分类器模块的随机森林模型主要用于识别网络行为转换点（隧道噪声流量），这个模块的随机森林模型是多元分类器用于识别应用。

多数表决。KNN 通过 K 个最近邻的多数投票得到分类结果。在该模型中也使用了多数投票。分类器将输出网络数据流多个部分的识别结果，并通过多数投票，以确定哪些应用程序构成了数据流。例如，一个混合流量序列的各段识别结果次数最高的是：Twitter、WhatsAPP，则这个混合流量序列的识别结果为 Twitter、WhatsApp。

4.3 基于端到端的隧道内混合网络行为识别

前面基于分割决策模型构建隧道内混合网络行为分割识别系统时，事实上需要多个模型：用于识别隧道噪声的网络行为转换检测模型、用于识别混合流类型的分割决策模型和用于识别隧道流量的分类模型。而在包括混合流识别等在内的许多领域中，近年来的一个趋势是，将本来由多个模型完成的任务，通过一个单一的模型来完成，这类方法被称为端到端方法。端到端方法拥有诸多好处。从工程上讲，这种方法能够简化开发框架，缩短开发周期，从而节省开发成本。从学术上讲，使用单一模型能够避免多个模型之间的错误积累，很多情况下反而能够获得更好的性能。

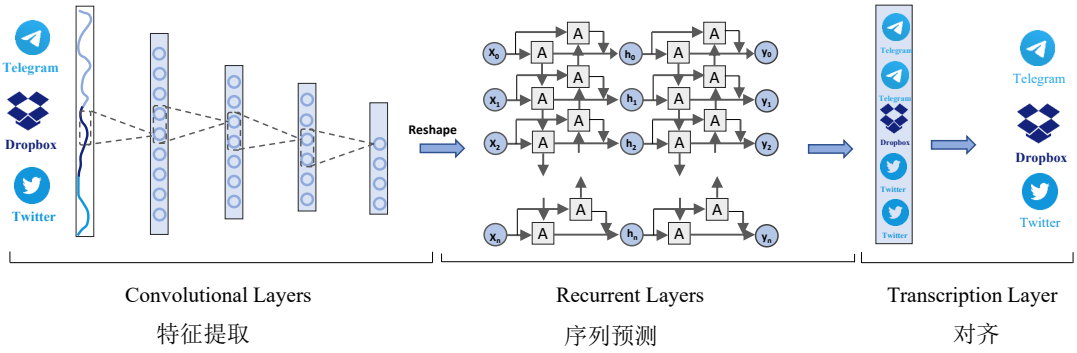


图 4.3 CRNN 框架图

Figure 4.3 The framework of CRNN

卷积神经网络是一种端到端的识别方法，可以实现序列预测。鉴于端到端方法在混合流量识别的多项优势，本节采用卷积循环神经网络的模型（CRNN) 对隧道混合流量进行分割识别。CRNN 模型主要包括三个部分，特征提取、序列预测和对齐。特征提取模块主要是使用深度卷积神经网络（CNN），对输入的隧道

混合流量提取特征，得到特征图。序列预测模块使用循环神经网络（RNN）对特征序列进行预测，对序列中的每个特征向量进行学习，并输出预测标签分布。对齐模块使用 CTC 损失，把从序列预测模块中获取的一系列标签分布转换成最终的标签序列。该方法的优点在于不需要花费时间寻找分割点，减少了时间延迟，而且可以进行整体训练。

4.3.1　特征提取模块

特征提取模块从隧道混合网络行为流量里面提取特征，得到特征图。近些年，卷积神经网络被广泛应用于流量识别领域提取特征，取得了很好的识别效果。卷积神经网络通过不断卷积池化的结构来构建网络，通过卷积能够利用局部信息的特性来提高在流量上的处理性能。卷积神经网络被广泛应用于流量识别领域提取有效特征。为此，在序列预测模型中使用卷积神经网络从隧道混合网络行为流量提取特征，卷积层的组成部分是通过使用标准 CNN 模型中的卷积层和最大池化层（去掉全连接层）来构建的。该组件从输入隧道流量中提取连续的特征表示。在输入网络之前，所有的流量都需要缩放到相同的长度，然后从卷积层组件生成的特征图中提取出一系列的特征向量，卷积层组件是循环层的输入。特征序列的每个特征向量在特征图上按列从左到右生成。

一维卷积神经网络主要由一组滤波器组成，对输入进行卷积运算，并将运算的结果传至下一层。网络流量具有一维特性，因此卷积神经网络维度选择为一维。模型包括 4 个一维卷积，他们的卷积核尺寸都设置为 3，输入通道分别设置为 1，32，32，32，卷积核的步长分别设置为 1，2，1，2，padding 设置为 0（默认），bias 初始化为 0，并且使用 Leaky Relu 激活函数作非线性处理，避免训练时出现梯度爆炸和梯度消失。

4.3.2　序列预测模块

Burst 分割聚合方法在将混合流分割为段的时候，没有考虑段与段之间的上下文信息。然而在实际中，隧道内网络行为在时间上具有关联性，因此在 CRNN 模型中，考虑了段与段之间的上下文信息，加入了 RNN 模型。一方面，RNN 模型具有很强的序列上下文信息捕获能力，基于上下文识别隧道网络行为段会更加准确。另一方面，RNN 可以将误差微分反向传播到它的输入，即卷积层。使可以在一个统一的网络中联合训练循环层和卷积层。RNN 能够对任意长度的序

列进行操作，具有处理不同输入的能力。

LSTM 是一种特殊的 RNN, 两者的区别在于普通 RNN 单个循环内部只有一个状态。而 LSTM 的单个循环结构内部有四个状态。相比于 RNN，LSTM 循环结构之间保持一个持久的单元状态不断传递下去，用于决定哪些信息要遗忘或者继续传递下去。单个 LSTM 由一个单元模块和三个门组成，即输入门、输出门和遗忘门，门函数如表达式 4.5 所示。

$$\begin{aligned} f_t &= \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ i_t &= \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \\ o_t &= \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \\ h_t &= o_t \circ \sigma_h\left(c_t\right) \end{aligned} \tag{4.5}$$

其中，f_t 为遗忘门，i_t 为输入门，o_t 为输出门。W_f、U_f 为权重，h_{t-1} 为中间态，x_t 为输入，c_t 为 cell state，b_f 为偏置。

GRU 是 LSTM 网络的一种效果很好的变体，它较 LSTM 网络的结构更加简单，而且效果也很好。GRU 可以解决 RNN 网络中的长依赖问题。GRU 模型中只有两个门，分别是更新门和重置门。门函数如表达式 4.6 所示。

$$\begin{aligned} z_t &= \sigma\left(W_z \mathrm{x}_t + U_z\, \mathrm{h}_{t-1}\right) \\ r_t &= \sigma\left(W_t \mathrm{x}_t + U_t\, \mathrm{h}_{t-1}\right) \\ \tilde{h}_t &= \tanh\left(W \mathrm{x}_t + U\left(r_t \circ \mathrm{h}_{t-1}\right)\right) \\ \mathrm{h}_t &= \left(1 - z_t\right) \circ \mathrm{h}_{t-1} + z_t \circ \tilde{h}_t \end{aligned} \tag{4.6}$$

其中，z_t 是更新门，r_t 是重置门，$\tilde{h}_t$ 是候选隐藏状态，h_{t-1} 是隐藏状态。x_t 是输入，W_z、U_z 为权重。

双向 RNN 结构有利于捕获上下文信息，使得多层 RNN 结构有利于捕获高级语义，在 NLP 领域有很好的性能提升。在这里也选择了双向 GRU 和双向 LSTM，捕获流量层面时间维度的上下文信息。

4.3.3 对齐模块

隧道流量进行序列预测的时候，预测输出和实际标签可能存在着长度不一致，没有进行对齐。对齐模块用于解决神经网络数据的真实标签和预测数据输出

不能对齐的问题。在端到端的混合流识别场景中，解析出的数据是 tensor 变量，并没有标识来分割混合流量，在用模型预测输出的时候也没有这种分隔符。但是数据的标签是分割明显的流量。在序列预测模块输出的元素是每个应用的概率，这就需要 CTCLoss 来进行辨别。该方法优点是不用强制对齐标签且标签可变长，仅需输入序列和监督标签序列即可进行训练。该方法也应用于其他场景文本识别、语音识别以及手写字识别等工程领域。用于 CTC 的损失函数是 Label Error Rate(LER)，损失函数如下：

$$LER\left(h,S'\right)=\frac{1}{|S'|}\sum_{(x,z)\in S'}\frac{ED(h(x),z)}{|z|} \tag{4.7}$$

其中，S' 是测试集合，x 是输入序列，z 是目标序列。每个样本构成一个序列对 (x,z) 。h 是一个时序分类器。ED(h(x),z) 函数计算 h(x) 和 z 的最小编辑距离。

4.4　实验评估

4.4.1　基于分割决策的识别结果

在本节中，首先介绍了实验的相关设置。然后，分析了三个因素（数据包长度阈值、衡量范围和 Burst 聚合数目）对识别结果的影响。最后，分析了三种情况下的流量识别结果和比较实验的结果。

4.4.1.1　实验设置

数据集。在实验中利用第三章节产生的 IPSec 隧道下的数据集 ipsec-data，主要包括三种类型的数据集：

- 正时间分离应用程序流量。从 30 个应用程序中，任意选择两个应用程序产生的流量进行混合，并收集包含 60 个类的混合流量，每个类包含 100 个实例。
- 零时间分离应用程序流量。这个数据集包含 60 个类，每个类包含 100 个实例。
- 负时间分离应用程序流量。具有负时间分离的应用程序流量包括两种类型的数据集，即 5% 和 10%。每种类型的数据集包含 60 个类，每个类包含 100 个实例。

比较实验。为了证明提出的方法在隧道混合流量上的识别效果，与 Section-Time 和 Section-Length 方法进行了比较，总结如下：

- Section-Length（基于数据包的）：利用均匀分割思想，分块进行识别。该方法使用数据包数目将混合流均匀地分成几个部分，预测每个部分，并得到分类结果。
- Section-Time（基于时间）：使用数据包到达时间间隔信息将混合流均匀地分成几个分段，并识别每个段，得到分类结果。

在 Section 方法的论文中使用 K 近邻算法（KNN）构建分类器模型。在比较实验中，发现随机森林 (RF) 比 K 近邻算法的识别效果更好，因此选择了随机森林构建分类器模型。

网络行为转换检测模块的设置。在实验中介绍 L_{split} 的选择过程。网络行为转换检测分类器以数据包长度序列的 54 维统计特征作为输入，该分类器使用随机森林模型。分割决策使用 KNN 模型。

Burst 分割聚合模块的设置。实验中将有 $N_{combine}$ 参数的选择过程介绍。分类器以数据包长度序列的 54 维统计特征作为输入，分类器使用随机森林模型。

评估指标。实验中使用准确率、精度、召回和 F1 值作为评价指标，相关公式如第三章的公式 3.2、3.3、3.4 和 3.5。分割点识别准确率的计算公式如 4.2 所示。

4.4.1.2 数据包长度阈值的影响

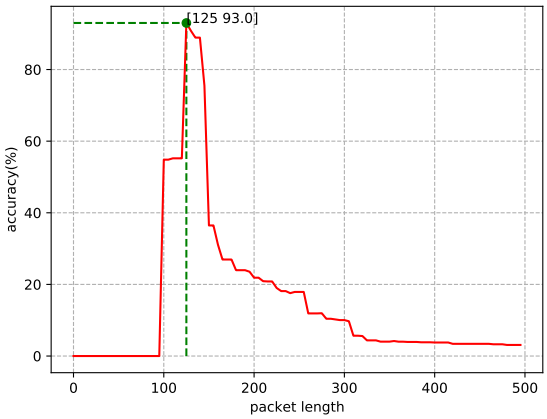


图 4.4 分割的准确性随数据包长度阈值的变化

Figure 4.4 The accuracy of the split varies with the packet length threshold .

为了分析数据包长度阈值对分割精度的影响，本节进行了数据包长度阈值分割混合流量的实验。在实验中，L_{split} 的值设置为从 5 到 500。

图 4.4 显示了在使用不同的数据包长度阈值时，隧道混合流量的分割准确率。当包长度阈值为 100 或 145 时，分割结果发生显著变化。当数据包长度在 100 - 145 的范围内，则分割准确率大于 50%。当数据包长度为 125 时，分割准确率达到最高值为 93%。结果表明，当数据包长度阈值在 100-145 之间时，有助于提高分割准确率。

可以看出，数据包长度阈值对分割准确率有很大的影响。随着数据包长度阈值的增加，分割的精度首先增加，然后呈下降趋势。如果选择的 L_{split} 太小，低于噪声包长度，难以识别隧道噪声，从而降低了分割准确率。当 L_{split} 被选择得太大时，高于隧道噪声的数据包长度会过滤掉一些应用程序的流量，从而降低分割准确率。

4.4.1.3　衡量范围的影响

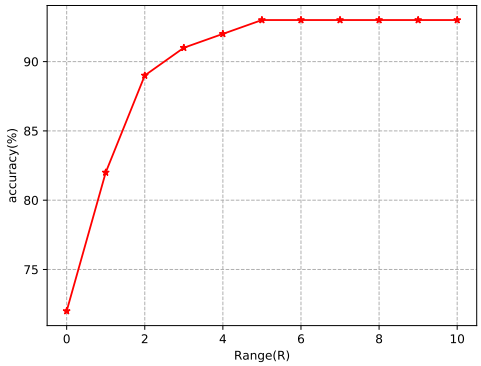


图 4.5 不同测量范围指标下分割的预测精度

Figure 4.5 Prediction accuracy of split with varying measure range .

为了分析不同衡量范围指标下分割准确率的变化，本节进行了一次实验，实验结果如图 4.5 所示。

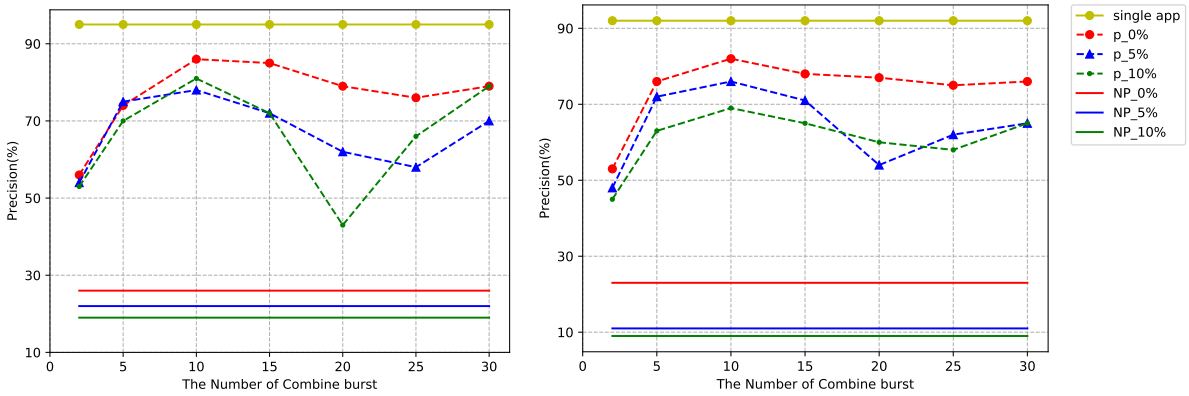
公式 4.1 中的 R 表示在分割点误差范围内的前后 R 个数据包都是正确的分割点。如图 4.5 所示，在不同衡量范围下，分割准确率略有差别。随着 R 从 0 增加到 10，混合流的分割准确率从 71% 提高到了 93%。当 R 的值达到 5 时，分割准确率不再提高。结果表明，分割点的分割偏差保持在前后 5 个数据包内。

从图 4.5 可以看出，随着 R 的增加，分割准确率开始提高，然后稳定下来。当 R 设置较小时，准确率就会很低。因此，在实验的误差容忍度范围内，R 设置应尽可能的大。

4.4.1.4 Burst 聚合数目的影响

本节进行了一个实验来查看 Burst 聚合个数对分类结果的影响。实验结果如图 4.6 所示。在结果中，P 表示 TMT-RF 框架处理的识别结果，NP 表示没有经过 TMT-RF 框架未处理的识别结果。

图 4.6 分别显示了首部和尾部应用识别的精度。在不同的重叠率下，应用程序的识别结果差异较大。经过 Burst 分割聚合模块处理后，混合流量的识别精度提高了 50%。由于 Burst 聚合数目 ($N_{combine}$) 的不同，识别精度也会发生变化。实验表明，如果 Burst 聚合个数为 10 或 15，有助于提高识别结果。



(a) 首部应用 **(b) 尾部应用**

图 4.6 应用的识别结果

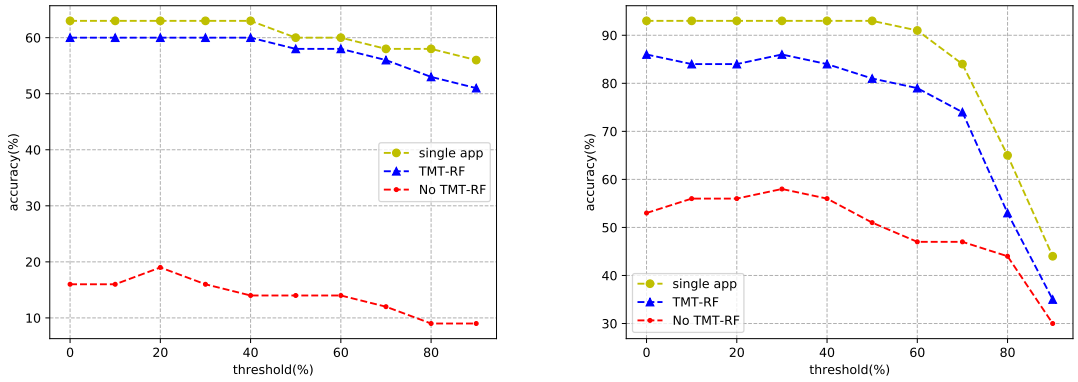
Figure 4.6 Classification results for multiple applications.

实验结果表明，总体上升趋势为先升后降。太小的 $N_{combine}$ 选择可能导致包含的应用信息太少，使得难以有效识别，降低了识别精度。$N_{combine}$ 选择太大可能会导致段内包含多个应用程序信息，降低识别精度。

4.4.1.5 正时间分离应用识别结果

通过以上分析了数据包长度阈值和测量范围对分割精度的影响，在实验中，将 L_{split} 设为 125，R 设为 6。如图 4.7 所示，经过 TMT-RF 框架处理后，混合流量中首部应用的识别结果与单应用假设的识别结果非常接近。

与未经 TMT-RF 框架处理的结果相比，隧道内混合流量的识别结果提高了 30%。与尾部应用相比，首部应用更接近于单个应用的分类结果。总的来说，可以看出，混合流对尾部的影响比首部应用更大。主要在于对隧道混合流量进行分割的时候，首部应用的后面数据包和尾部应用的前面数据包会被误切割。这表



(a) 首部应用　　(b) 尾部应用

图 4.7 正时间分离应用的分类结果

Figure 4.7 Classification results for multiple applications.

明，前面数据包对隧道流量识别的影响远大于后面数据包，前面数据包对于流量识别更有帮助。

4.4.1.6　零时间分离应用识别结果

通过分析 Burst 聚合数目对应用识别的影响，将 $N_{combine}$ 设为 10。在这种情况下，首尾应用程序之间没有重叠，所有三种方法都取得了很好的识别效果。在表 4.1 中，R 表示应用程序流量的重叠率，P 表示概率阈值。从表 1 可以看出，在不同的输出概率阈值下，该方法的识别结果变化不大。首部应用的识别结果优于尾部应用的识别结果。

表 4.1 不同重叠率下应用的识别精度

Table 4.1 The precision of the applications under different overlap rates.

重叠率（%）	0					5					10				
置信度（%）	0	20	40	60	80	0	20	40	60	80	0	20	40	60	80
首部应用	0.87	0.87	0.85	0.87	0.84	0.83	0.75	0.61	0.83	0.71	0.78	0.78	0.70	0.76	0.71
尾部应用	0.84	0.84	0.84	0.74	0.71	0.76	0.76	0.65	0.58	0.56	0.69	0.69	0.67	0.62	0.57

从表 4.2 中可以看出，Burst 分割聚合方法在该数据集上比其他方法具有更好的识别结果。Burst 分割聚合方法得到 87% 的精度和 76% 的 f1 值。总的来说，Burst 聚合已经取得了很好的性能，因为它使用了同一方向数据包之间的信息。结果表明，Burst 聚合方法对提高隧道流量的识别具有很大的帮助。

表 4.2 在精度、召回和 F1 值的实验结果

Table 4.2 Experimental result on precision, recall and F1.

重叠率（%）	0						5						10					
	首部			尾部			首部			尾部			首部			尾部		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
TMT-RF	0.87	0.71	0.76	0.74	0.45	0.56	0.83	0.69	0.73	0.58	0.40	0.47	0.76	0.69	0.72	0.62	0.17	0.27
Section-length	0.83	0.18	0.22	0.45	0.33	0.34	0.61	0.11	0.13	0.58	0.29	0.38	0.59	0.09	0.11	0.52	0.17	0.22
Section-time	0.78	0.45	0.51	0.64	0.25	0.26	0.76	0.26	0.28	0.47	0.29	0.27	0.72	0.26	0.30	0.32	0.23	0.21

4.4.1.7 负时间分离应用识别结果

本节将 $N_{combine}$ 设置为 10，如上所示。如表 4.1 所示，不同重叠率下隧道混合流量的识别结果。从表 4.1 中可以看出，重叠率对应用程序识别的影响较大，重叠率越高，识别准确率越低。从表 4.1 中还可以看出，输出阈值对识别的精度也有一定的影响。

如表 4.2 所示，随着重叠率的增加，Burst 分割聚合方法与其他方法之间的差距逐渐增大。与最先进的方法相比，Burst 分割聚合方法在不同的重叠率下表现出最好的性能。Burst 分割聚合方法的精度略高于其他两种方法，召回率和 f1 值远大于这两种比较方法，这主要是由于 Burst 聚合方法使用了数据包之间的方向信息。随着重叠率的增加，Burst 分割聚合方法在精度、召回和 f1 值方面下降最小，在比较方法中表现出较强的鲁棒性。

整体看来，随着重叠率的增加，隧道混合流量的识别结果下降明显。通过三种类型混合流量的识别结果，表明该方法可以很好处理正时间分离应用流量、零时间分离应用流量和低重叠率的负时间分离应用流量，但对于高重叠率的负时间分离应用流量处理效果较差。

4.4.2 基于端到端的识别结果

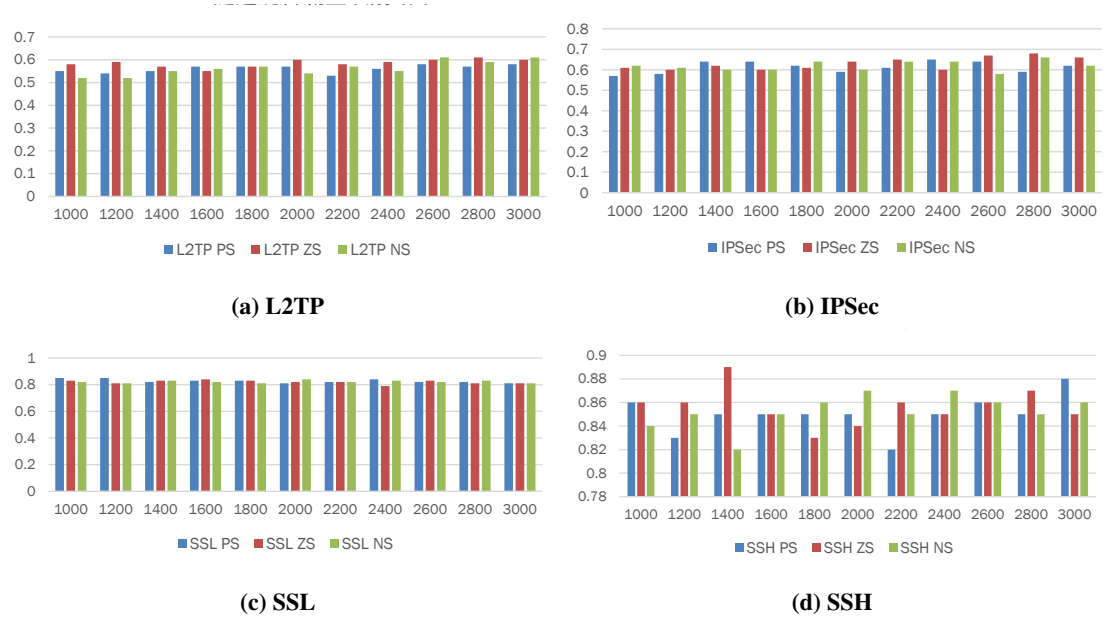
4.4.2.1 实验设置

数据集。本节利用第三章节产生的自采集数据集：IPSec、SSL、L2TP 和 SSH 隧道内的混合应用的数据集: ipsec-data、ssl-data、l2tp-data 和 ssh-data, 共计 40 个数据集。公开数据集是两种 Tor 上混合网站的数据集：tor14-data 和 tor17-data，共计 20 个数据集。

评价指标。我们使用前面第三章的评价指标（3.2）。

4.4.2.2　模型选择

本节实验的重点是如何设计 CRNN 模型。在此，本节选择了 CGRU、CLSTM、C-BiGRU 和 C-BiLSTM。



(a) L2TP　　**(b) IPSec**

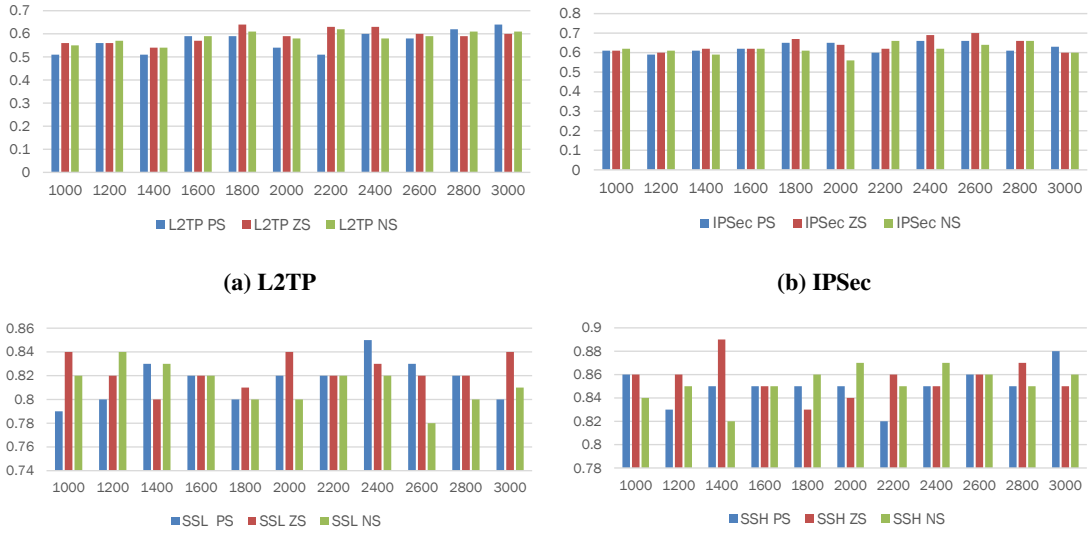
(c) SSL　　**(d) SSH**

图 4.8 CGRU 模型在四种隧道数据集上的分类结果

Figure 4.8 Classification results of the CGRU model on four tunnel datasets .

CRNN 模型主要由 CNN、RNN 和 CTC 损失组成，在四种隧道 40 个数据集上的识别结果如图 4.8-4.11 所示。当 RNN 选择 GRU 或者 LSTM 的时候，CRNN 模型在 L2TP 和 IPSec 三种类型的数据集上，识别结果达到 60% 以上。在 SSL 和 SSH 三种类型的数据集上，识别结果达到 80% 以上。当 RNN 选择双向 GRU 和双向 LSTM 的时候，CRNN 模型在在 L2TP 和 IPSec 三种类型的数据集上，识别结果达到 90% 以上。在 SSL 和 SSH 等三种类型的数据集上，识别结果达到 95% 以上。总体来看，SSL 和 SSH 的识别效果最好，可以看到不同隧道内的混合流量识别结果略微有差异，主要在于隧道协议之间的差异性。其中隧道内三种类型的混合流量：正时间分离应用数据集、零时间分离应用数据集和负时间分离应用数据集的识别结果差别不大，原因在于 CRNN 模型忽略了重叠部分对识别的影响，对于重叠部分有着容错机制。综合看来，无论是双向 GRU，还是双向 LSTM 都表现出很好的识别效果。

从同一个隧道内应用流量的识别结果来看，在不同序列长度输入下，识别结

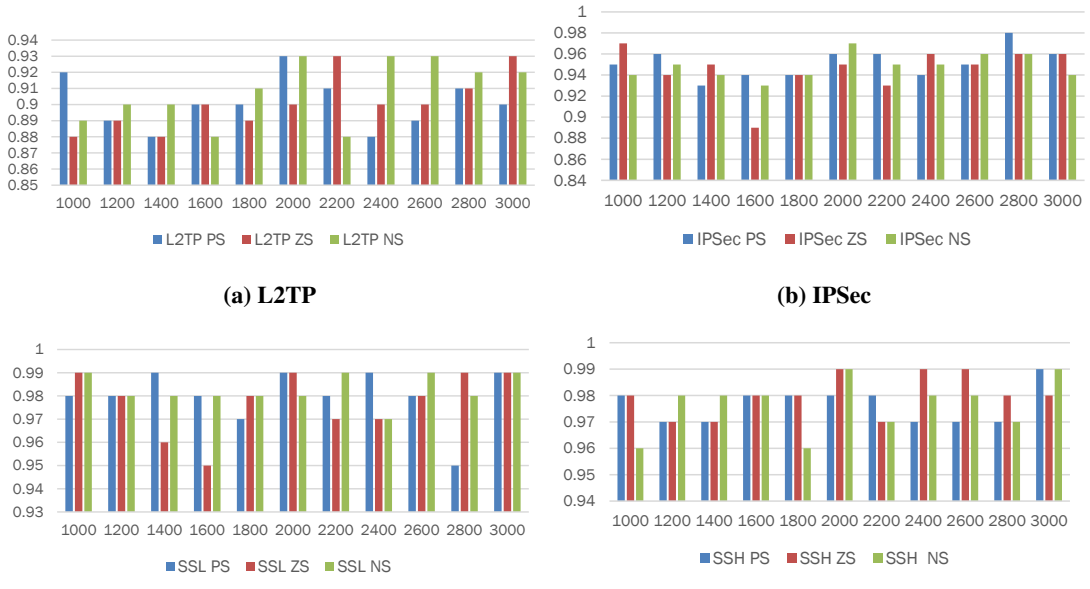


(a) L2TP **(b) IPSec**

(c) SSL **(d) SSH**

图 4.9 CLSTM 模型在四种隧道数据集上的分类结果

Figure 4.9 Classification results of the CLSTM model on four tunnel datasets .

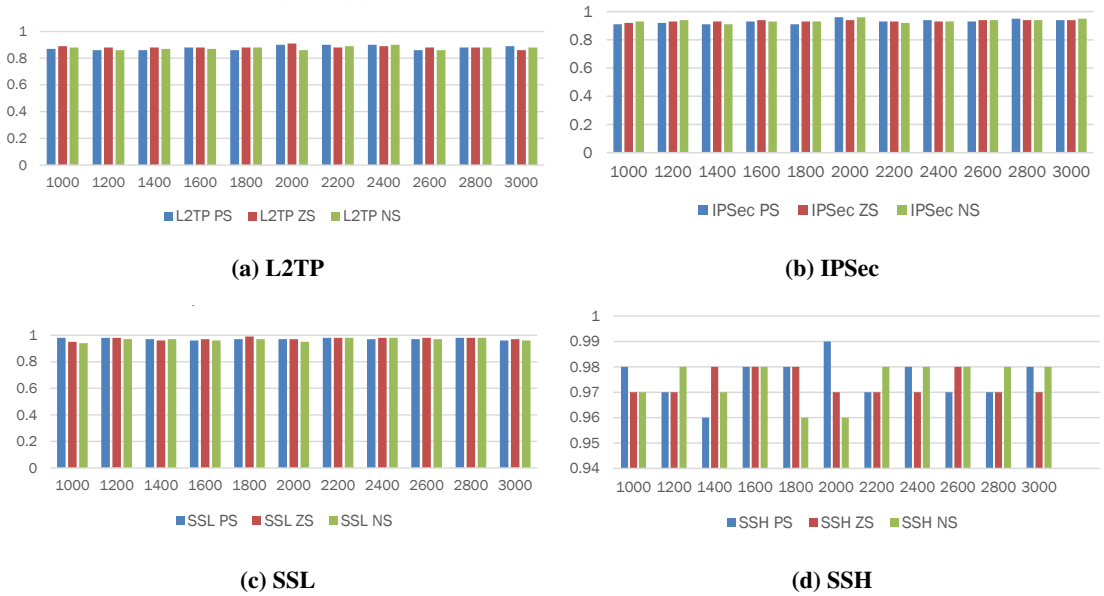


(a) L2TP **(b) IPSec**

(c) SSL **(d) SSH**

图 4.10 C-BiGRU 模型在四种隧道数据集上的分类结果

Figure 4.10 Classification results of the C-BiGRU model on four tunnel datasets .



(a) L2TP　　**(b) IPSec**

(c) SSL　　**(d) SSH**

图 4.11 C-BiLSTM 模型在四种隧道数据集上的分类结果

Figure 4.11 Classification results of the C-BiLSTM model on four tunnel datasets .

果虽有略微差异，但整体看来变化不大。原因在于 CRNN 模型可以接受任意长度的输入，对于不同长度的序列都能够有效的识别。在双向 RNN 模型下，对于同一个的长度的序列输入，精度、召回和 F1 值差别不大，基本都保持在 85% 以上。证明了该方法在隧道内混合应用流量识别领域的有效性。

通过上述模型分析，双向 RNN 模型优于单向的，原因在于双向 RNN 充分利用了上下文信息，单向 RNN 仅利用了上文信息。单向 RNN 仅根据前面的信息推出后面的信息，在隧道内混合应用流量识别方面只根据前面信息是不够的。隧道内应用流量的混合不仅与前一段的转态有关，还与下一段的状态有关，因此双向 RNN 可以更好的处理隧道内混合应用流量识别问题。GRU 只含有两个门控结构，且在超参数全部调优的情况下，与 LSTM 性能相当。双向 GRU 模型和双向 LSTM 模型的分类结果比较接近，但双向 GRU 结构更为简单、参数和所需训练样本较少，收敛速度更快。综上来看，双向 GRU 能够捕获上下文信息，使用多层 GRU 结构有利于获取高级语义，所以选择双向 GRU 模型作为循环层，其中使用多层 GRU 结构叠加。

4.4.2.3　隧道混合流量的识别结果

通过模型选择，得到了 C-BiGRU 模型，在自采集数据集和公开数据集上进行了测试。

表 4.3 公开混合数据集上识别结果

序列长度	Tor14			Tor17		
	正时间分离	零时间分离	负时间分离 (10%)	正时间分离	零时间分离	负时间分离 (10%)
1200	0.88	0.88	0.86	0.80	0.80	0.77
1600	0.92	0.91	0.90	0.83	0.83	0.82
2000	0.90	0.90	0.91	0.84	0.83	0.82
2400	0.90	0.91	0.92	0.84	0.82	0.81
2800	0.93	0.93	0.94	0.86	0.82	0.85

实验中，选择准确率作为评价标准。在公开数据集上总体分类效果来看，如表 4.3 所示，在 Tor14 公开数据集上，正时间分离流量、零时间分离流量和负时间分离流量分类的准确率都达到了 90%。Tor17 公开数据集上，正时间分离流量、零时间分离流量和负时间分离流量分类的准确率都达到 85% 附近。两种公开数据集上分类结果的差距可能与数据的规模有关，Tor14 是 60 类网站流量的混合，Tor17 是 100 类网站流量的混合。数据集规模大了快 2 倍，分类准确率只有 5% 的下降，表明该模型具有很强的泛化能力。该模型在公开数据集上取得很好的识别结果。

总体来看，正时间分离应用流量识别结果和零时间分离应用流量基本相同，均优于负时间分离应用流量。其原因在于正时间分离应用流量和零时间分离应用流量仅存在网络应用的混合，没有存在重叠。网络行为之间存在明确的界限。而负时间分离应用流量存在混合和重叠，重叠流量会影响整个的识别结果。三种类型混合流量的识别结果略有些差异，但差别不大，可见该模型能够很好的处理各种类型的混合流量，原因在于该模型忽略了重叠部分的影响，弱化了重叠部分对整体识别结果的作用。该模型在公开数据集上对混合流量的识别结果均保持在一个良好水平，说明了使用端到端的思想在混合流量识别领域的一个可用性。

此外，还考虑了不同重叠率对该模型的影响，选择准确率作为衡量标准。如表 4.4 所示，从总体识别效果来看，不同隧道下识别结果略有差异。SSH 和 SSL 隧道下识别准确率略高于其他集中隧道。随着重叠率的增加，多种隧道识别结果下降。从同一个隧道内不同重叠率下的识别准确率来看，重叠率越大，混合网络行为的识别准确率越小。结果表明不同重叠率的隧道混合流量对该模型影响效

表 4.4 在自采集数据集和公开数据集上的识别结果

Table 4.4 Classification results on self-collected datasets and public datasets.

	重叠率					
	15%	**20%**	**25%**	**30%**	**35%**	**40%**
IPSec	0.95	0.93	0.92	0.92	0.89	0.88
L2TP	0.91	0.90	0.90	0.88	0.87	0.87
SSH	0.97	0.97	0.96	0.94	0.91	0.89
SSL	0.97	0.95	0.94	0.91	0.91	0.91
Tor14	0.91	0.91	0.89	0.88	0.88	0.86
Tor17	0.83	0.82	0.82	0.80	0.77	0.71

果不同。其原因在于随着重叠率的增加，导致单应用流量的可用信息减少，进而影响整个识别结果。整体来看，虽然重叠率不断增加，混合流量的识别准确率确下降不明显，主要在于该模型对重叠流量有一定的缓冲空间。此方法在混合流量识别取得了很好的识别结果，还应注意到，随着混合网络行为数据集的扩大，识别结果会有一定程度的下降。此外，端到端方法使用了 GRU 方法，训练速度较慢，受 CTC 算法对速度的要求，输出长度受到限制，识别的混合流不能太长。

4.5　本章小结

本章在第三章生成的隧道混合流量的数据集上，开展了隧道内混合网络行为精细化识别的研究。针对隧道内网络行为存在混合和重叠，本章提出了基于分割决策的分割方法和卷积循环神经网络进行识别的方法。基于分割决策的方法将分割和识别过程分开，首先将隧道内混合网络行为进行分割，提取出单一网络行为流量。然后输入单网络行为识别模型进行处理。实验结果表明该方法对于正时间分离应用流量有显著的分类效果。端到端的方法主要是利用一个模型完成分割和识别过程，可以有效处理不同混合重叠率下的隧道混合流量。在四种隧道的混合数据集和公开数据集上对该模型进行了评估，结果表明该方法可以有效识别不同重叠率下的隧道混合流量。

第 5 章　隧道内混合网络行为识别原型系统

在真实使用隧道的过程中，情况更加复杂，产生的流量更是多种多样，使得隧道内混合网络行为分析更加困难。本章针对隧道内真实使用环境，首先提出了基于 Burst 的隧道流量分割技术，其次利用分割决策模块进行模型决策，分别选择网络行为转换检测、首尾段分割处理和端到端模型处理，然后利用数据集对该系统进行了评估，验证了其方法的有效性。接着，结合 Burst 分割技术、分割决策技术、网络行为转换检测方法和端到端识别技术构建了原型系统，实现对真实隧道内混合网络行为的分析。最后对本章的相关工作进行了小结。

5.1　引言

在用户使用隧道的过程中，会产生各种类型的混合流量，而且流量之间差别较大。隧道将用户产生的所有访问不同目标的流量都统一封装成同一种隧道协议，服务端 IP 地址和端口也会被混淆为隧道服务端的 IP 地址和端口。单用户使用隧道过程中，生成多种类型网络行为的流量。隧道内的所有行为流量都具有相同的五元组信息，各行为流量难以通过明文进行区分识别。与传统加密流量相比，隧道流量存在难以分割的问题。

按照隧道内网络行为发生混合的时刻不同，分为正时间分离应用流量、零时间分离应用流量和负时间分离应用流量。正时间分离应用流量是指，各网络行为之间存在正时间的时间间隔。零时间分离应用流量是指，各网络行为之间不存在时间间隔。负时间分离应用流量是指，各网络行为之间存在重叠。

传统上对隧道混合流量的识别方法中，仅研究了隧道内单网络行为的问题，即隧道内仅承载一种类型的应用流量。而没有关注到隧道内多网络行为。隧道内多网络行为存在混合和重叠，识别之前需要进行分割然后再进行识别。隧道内多种网络行为差别较大，每种类型流量都具有各自的特点。为此，本章节提出了基于分割决策的隧道混合网络行为流量的选择性处理方法，解决隧道内多种类型混合流量的方法。

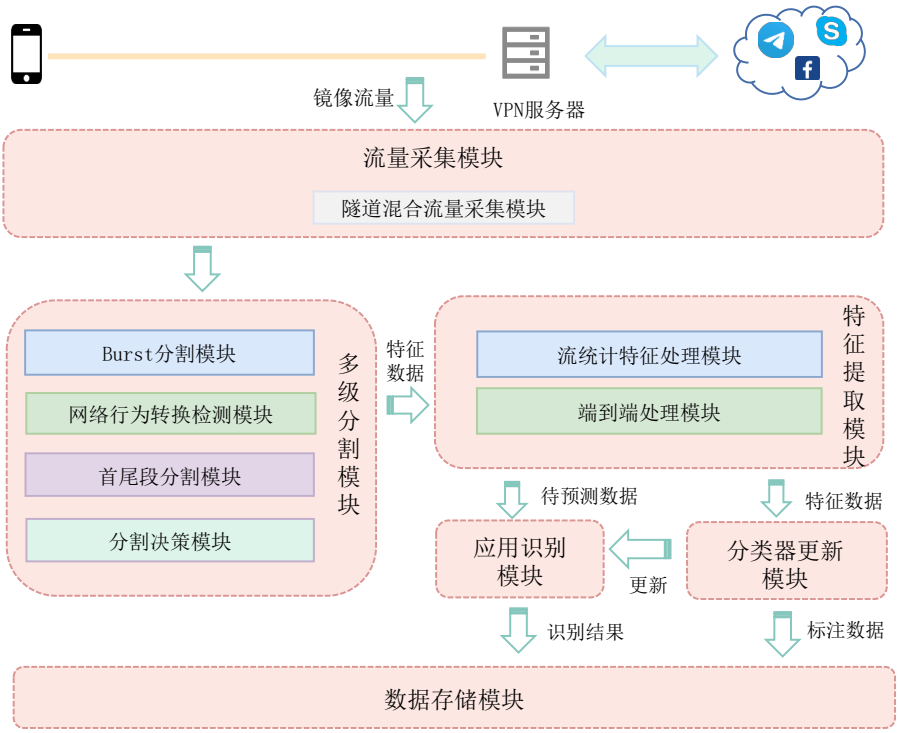


图 5.1 系统组织架构图

Figure 5.1 Organizational Structure of the System..

5.2 隧道内网络行为识别原型系统的构建

为了对真实操作的隧道内混合网络行为进行识别，基于前面章节的研究成果，设计实现了了一套原型系统。系统的组织架构图如图 5.1 所示。

整个系统处理过程包括：流量采集模块、多级分割模块、特征提取模块、应用识别模块、分类器模块和数据存储模块。在流量采集模块，系统从网络中采集隧道混合流量。为了将承载在同一个隧道内的不同网络行为流量分割出来，以便进行后续识别处理。在多级分割模块，考虑真实用户的操作，利用时间 Burst 分割处理模块进行首次分割，然后采用网络行为转换检测模块进行二次分割。其次，通过分割决策模块对前面的分割结果选择性处理。对于分割后是两应用混合的流量输入首尾段分割模块进行处理。特征提取模块通过流统计特征处理模块提取流量特征，端到端处理模块自动提取流量特征。应用识别模块对待预测得流量数据进行识别，分类器更新模块不断更新分类器。数据存储模块保存分类器的识别结果。

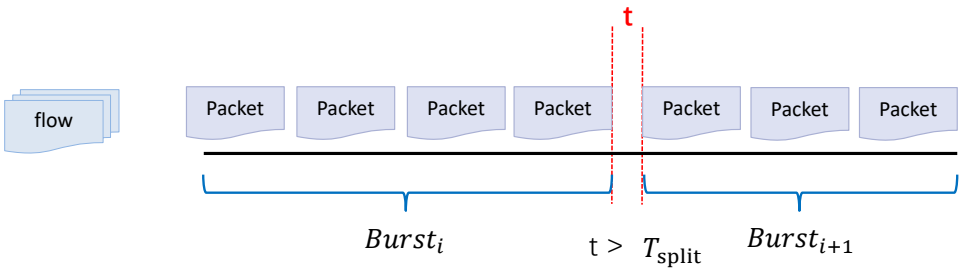


图 5.2 Burst 分割示意图

Figure 5.2 The Procession of Burst Splitting.

5.2.1　Burst 分割模块

由于隧道的封装和加密特性，所有协议的网络流量经过隧道后，都被封装为同一种协议的流量，具有相同的五元组信息（源 IP、源端口、目的 IP、目的端口和协议版本号）。因此，无法利用明文信息提取单一流量，网络流量的分割更加困难。

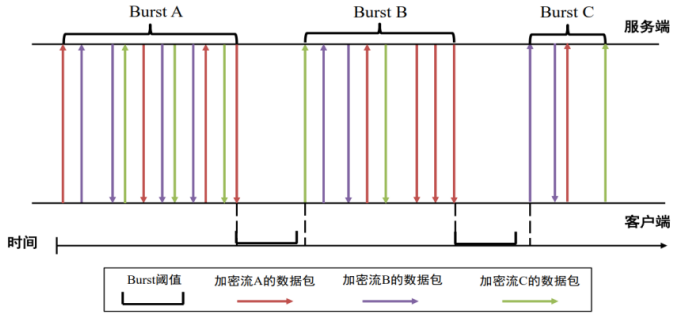


图 5.3 Burst 示意图

Figure 5.3 The Burst.

Burst，即流量突发，表示根据具体分类任务和流量特性将流切分为多个连续数据包组合。可分为方向 Burst 和时间 Burst，对应代表性切分依据为数据包方向改变和数据包间隔大于阈值。Burst 特征可以弥补流量指纹场景中的包粒度太细和流粒度太粗的缺点，可以根据具体任务构建合适粒度的基本单元，在本章节使用的是时间 Burst。

时间 Burst（以后简称 Burst）是一种流量分割方法，在一段时间产生的大量的网络流量，如果连续的两个数据包的时间间隔超过一定阈值，则两个数据包属于不同的 Burst。具体如图 5.3 所示，在流量数据包里面出现两个连续数据包的时

间间隔超过时间阈值，数据包流量被分为两个 Burst, 即 Burst A。接着，后面又出现了两个连续数据包的时间间隔超过阈值，从网络流量里面分割出了 Bust B。依次往后，分割出了 Burst C 等。Burst 可以很好的提取出短时突发的流量，具有快速、方便、简单等优势。

考虑到真实环境中，用户在隧道使用应用的过程中，会存在时间间隔（正时间分离应用流量）。正时间分离应用流量具有同一个应用的流量在时间上更加接近，不同应用流量之间具有时间间隔分离等特性。因此，Burst 分割方法可以更好的从网络流量中提取单应用流量。用户在实际使用过程中，应用之间会存在不同长度的时间间隔。Burst 的时间阈值设置不合适会对网络流量分割带来很大的影响。如果 Burst 时间阈值设置太长，会使得 Burst 包含足够的流量信息，但也会使得 Burst 内包含多个应用流量，使得提取出的网络流量不纯净。如果 Burst 时间间隔设置太短，会使得 Burst 包含足够少的应用（理想情况下仅一个应用），提取出的应用流量更加纯净，但会使得 Burst 包含的流量信息太少，单个应用流量被分割，影响识别效果。可以看到这两个是相互矛盾的，本节需要选择合适的 Burst 时间阈值进行分割。在实验部分，本节将通过实验选择合适的 Burst 时间阈值以及如何影响识别的效果。

5.2.2 网络行为转换检测模块

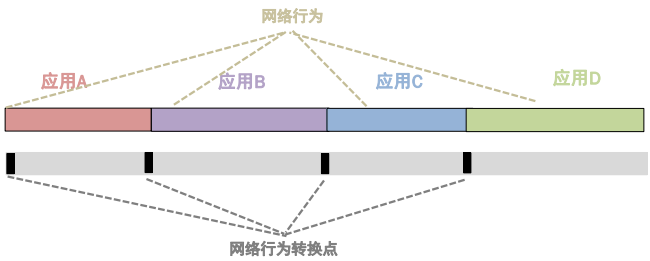


图 5.4 网络行为转换检测示意图

Figure 5.4 Schematic diagram of network behavior transition detection.

在真实环境中，应用之间的时间间隔内不是不会产生流量。而是会存在隧道自身产生的数据包（心跳数据包）。这些数据包的包时间间隔和包长度都具有显著的特性，这些特殊的数据包可以作为网络行为发生转换的标志。因此可以通过检测这些心跳数据包来分割隧道内混合网络流量。网络行为转换检测是指从网络流量中检测出网络行为发生转换时刻的方法。如图 5.4 所示，所要检测的流量

就是不同应用发生转换时刻的流量。

网络行为转换检测包括两个流程，过程如下。首先利用网络数据包长度定位到心跳数据包的位置，然后利用分类器识别心跳数据包位置的流量是否属于网络行为转换点流量，如果是的话，直接分割流量。如果不是的话，继续寻找下一个网络行为转换点。下面依次重复上述过程，直到最终寻找完网络行为转换点。基于心跳数据包和分类器的网络行为转换检测方法，可以快速定位网络行为发生转换的位置，并通过分类器做到准确的识别，具有快速、准确的优势，能够对混合流量及时的分割处理。

网络行为转换检测仅检测出网络行为在何时发生转换，不识别出是哪些网络行为。因此网络行为转换检测技术在此处仅用于分割。一个好的网络行为转换检测分割方法应具备以下两个特征：

- 每个流量片段中应只有一个网络应用，也就是说没有漏检的网络行为转换点。
- 每个相邻的流量片段属于两个不同的网络应用，也就是说没有错检，既不会出现将一个网络应用看做两个不同的应用，而使同一个网络应用的流量片段被过度分割的情况。

可以看出以上两个要求是矛盾的，漏检下降，错检就会增多。错检减少。漏检就会增多。实验中，通过选择合适的数据包包长和分类器，达到两者的平衡。

5.2.3　首尾段分割模块

传统的基于分割点寻找的首尾段分割方法，要耗费一定时间寻找分割点，，而且分割的结果直接影响识别的效果，难以做到快速、准确的识别。对于隧道内应用两两混合的流量，可以利用首尾段分割方法快速处理。首尾段分割方法是指提取首段表征前面应用，使用尾段表征后面应用。最终通过识别首尾段来识别整个混合流量。

该模块在整个框架中，主要处理应用两两混合的数据。通过分割决策模块识别出为应用两两混合的数据后，直接对该类型的数据直接进行处理。

5.2.4　端到端识别模块

传统的方法，在识别过程中将分割与识别分为两个模块，先进行分割，然后根据分割的结果再利用现有的模型进行识别。该方法不仅需要大量时间寻找分

割点，而且分割的结果对识别结果影响较大。此外，隧道内复杂混合流量（负时间分离流量）的分割效果较差，识别精度较低，传统的方法不能有效处理该类型的混合流量。

基于端到端的方法，将隧道混合流量分割和识别放在一个模型进行处理。利用卷积神经网络提取流量特征，然后利用循环神经网络对流量序列进行预测，最终通过转录层实现输出序列和真实标签的对齐，实现隧道内混合流量的端到端的识别。该方法减少了寻找分割点花费的时间，而且使用了时间维度上的重要信息。

该模块处理 Burst 分割模块、网络行为端分割模块无法处理的混合流量，利用端到端模型处理复杂的混合流量。

5.2.5 分割决策模块

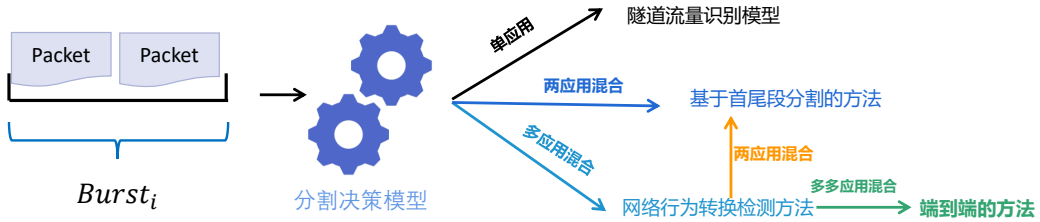


图 5.5 分割决策框架

Figure 5.5 A split decision framework.

隧道内存在多种类型的混合流量，流量形式差别较大。每一种识别方法难以对所有类型的数据都做到很好的识别效果，隧道内混合网络流量需要选择性处理。对于不同类型的方法，分别选择最优的方法分别处理。对于正时间分离的应用流量，可以使用 Burst 分割模块和网络行为转换检测模块处理，对于零时间分离和负时间分离的混合流量，通过端到端的模型进行处理。

隧道混合网络行为流量经过 Burst 分割模块处理后，存在三种情况的流量：

- Burst 分割成功。从复杂多样的隧道混合流量，提取出了单应用的流量，是最好的情况。理想情况下，希望该种类型的数据越多越好。
- Burst 分割失败。同一个 Burst 承载多个应用的流量，没有分割出所有的混合网络行为。对该种类型的流量，需要再次进行分割，提取出单应用流量。理想情况下，希望该种类型的流量越少越好。

- Burst 过度分割。单应用流量被 Burst 多次分割，多个 Burst 承载单个应用流量。对于这种类型的流量，对于流量识别影响不大。

对于这三种情况的流量，需要选择性处理。对于 Burst 分割成功和过度分割的混合流量，不需要再次进行分割，可以直接进行识别。对于分割失败的混合流量，需要再次进行分割，输入下一级分割模块进行处理。分割失败的包括两种情况：分割出的是应用两两混合的，或者多多混合的。对于两两应用混合的选择首尾段进行处理，多应用混合的选择网络行为转换检测进行下一级分割处理。

网络行为转换检测后，仍然使用分割决策再次进行选择性处理，选择过程如上所示。分割成功和过度分割的直接进行识别，分割出的两两应用混合的直接输入首尾段分割模块处理。分割出仍然是多应用混合的话，直接输入端到端模型进行分割识别。

5.3　实验评估

在该部分，首先本节对实验设置进行了简单介绍，然后对隧道内混合网络行为分割的各个模块进行了介绍，主要包括 Burst 分割模块、网络行为转换检测模块、首尾段分割模块、端到端识别模块和分割据测模块等。本节在实验中分析了各模块的参数选择及识别结果。

5.3.1　实验设置

数据集。为了验证多级分割架构分割隧道内混合网络行为流量的有效性。本节需要收集真实环境中的隧道混合流量对该框架评估。为了从真实环境中采集隧道混合流量，本节设计了真实环境中隧道流量收集框架。该框架主要包括隧道客户端和隧道服务端两个方面。本次实验评估选取 IPSec 隧道进行分析，在 Vultr VPS 上面搭建 IPSec VPN，然后在移动端连接 VPN，建立 IPSec 隧道。用户使用移动端，正常访问互联网，访问互联网的流量经过隧道客户端封装加密成隧道流量后发送给隧道服务端，具体过程如图 5.6 所示。

本框架需要采集的是隧道客户端和隧道服务端之间的隧道流量，邀请志愿者参与隧道流量的采集过程。实验中让志愿者按照平时上网习惯使用 18 种应用程序，例如 Skype、Facebook、Telegram、Dropbox 等。在前面章节手动合成的隧道混合流量里面，将混合流量分为三种：正时间分离应用流量零时间分离应用流量和负时间分离应用流量。然而在真实环境中收集的时候，很难将零时间分离应

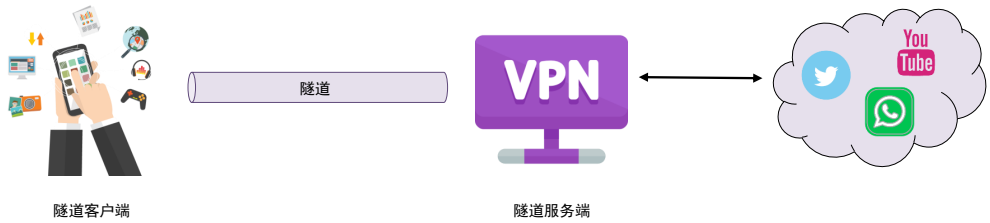


图 5.6 真实环境隧道流量生成框架

Figure 5.6 Real-world tunnel traffic generation framework.

用流量和正时间分离应用流量区分开，因此本节将真实环境中混合流量分为两类：

表 5.1 收集的应用程序名称

Table 5.1 Collected application name

Dropbox	Twitter	Facebook	Youtube	Linkedin	Instagram
OneDrive	FileZilla	Skype	Netflix	Yahoomail	Servu
Foxmail	Michat	Gmail	Outlook	Icloudmail	Hulu

1. 有时间间隔访问的流量：又称正时间分离应用流量，是指用户在访问应用的时候存在一定的空闲时间间隔，共收集 18 个应用，包括 Facebook、Twitter 等，详细如表 5.1 所示，共计 20.1GB 流量。

2. 无时间间隔访问的流量：包括零时间分离应用流量和负时间分离应用流量，是指用户访问应用的时候没有存在空闲时间间隔，共收集 18 个应用，包括 Skype、Youtube 等，详细如表 5.1 所示，共计 23.5GB。

为了针对不同类型的数据对各模块进行测试，实验中要求志愿者在标注的时候，做好不同类型的标注（有时间间隔访问流量和无间隔访问流量），收集两种类型的混合流量。

衡量指标。实验中通过统计 Burst 分割后各类型混合流量的占比，来衡量 Burst 分割效果。Burst 分割后共存在正确分割的、分割失败的以及过度分割的。正确分割的越多，分割失败的越少、过度分割的越少，表明该算法分割效果越好。实验中对模型和特征集的评价方法是 10 倍交叉验证方法。通过将识别标签与真实标签进行比较，得到了 n 类的 i 类的真阳性 (TP)、假阳性 (FP)、真阴性 (TN) 和假阴性 (FN)。为了评价该方法的性能，其计算结果为：

$$Accuracy = \frac{\sum_{i=1}^{n}(TP_i + TN_i)}{\sum_{i=1}^{n}(TP_i + FP_i + TN_i + FN_i)}$$

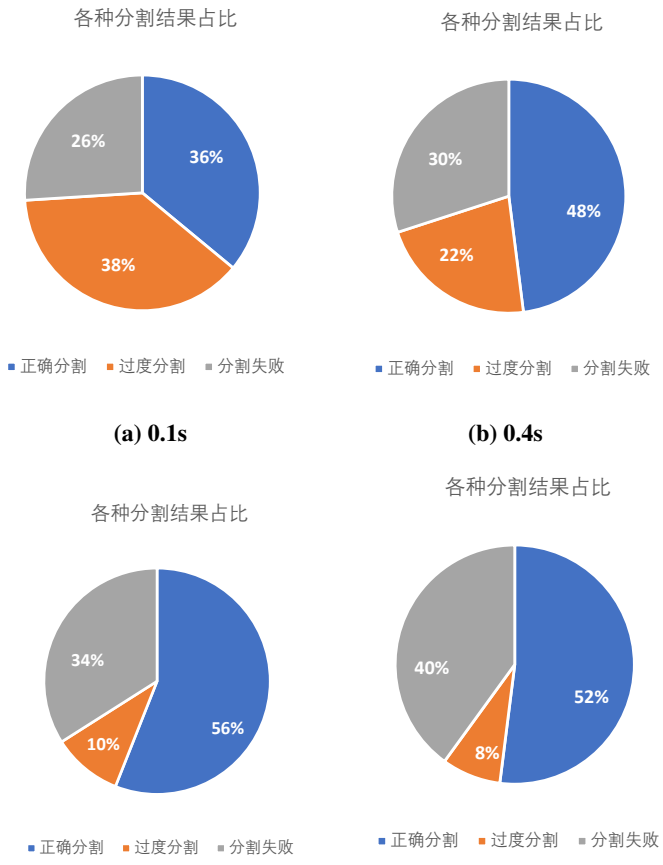
$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

5.3.2　Burst 分割模块实验评估

时间阈值的选择。为了选择合适的时间阈值，本节在数据集上分别选择了时间阈值 T_{split} 为：0.1,0.4,0.7,1，实验结果如下图所示：



(a) 0.1s　　**(b) 0.4s**

(c) 0.7s　　**(d) 1.0s**

图 5.7 不同时间阈值下的分割结果

Figure 5.7 Splitting results at different time thresholds.

如图 5.7 所示，随着时间阈值的增加，过度分割的情况越来越少，主要在于较小的时间间隔可能是应用内部产生的时间间隔，增大时间阈值 T_{split}，会减少这种情况的发生。随着时间阈值的增加，正确分割的情况呈现先增加后下降的趋势，在 T_{split} 取 0.7 的时候，达到最大占比为 56%。分割失败的一直在增加，主要原因在于，增大了时间阈值，使得小于时间阈值的混合流量无法被分割，从而使得分割失败的增多。

5.3.3 网络行为转换检测模块实验评估

网络行为转换检测，主要用来处理零时间间隔分离的应用混合流量。该模块可以处理网络行为之间时间间隔低于时间阈值，无法被 Burst 分割的混合流量。网络行为转换检测，主要是从隧道混合网络行为流量中定位识别到网络行为发生转换的时刻，实现对隧道混合网络行为的有效分割。

实验中，使用基于数据包和分类器相结合的网络行为转换检测方法，利用隧道内噪声数据包长度（心跳数据包，隧道自身产生的网络流量）定位到网络行为发生转换的时刻，然后通过分类器进行识别验证。识别结果如表 5.2 所示。总体来看，当数据包长度为 125 时，此时混合网络行为的识别准确率达到 90%，实现了很高的准确率。

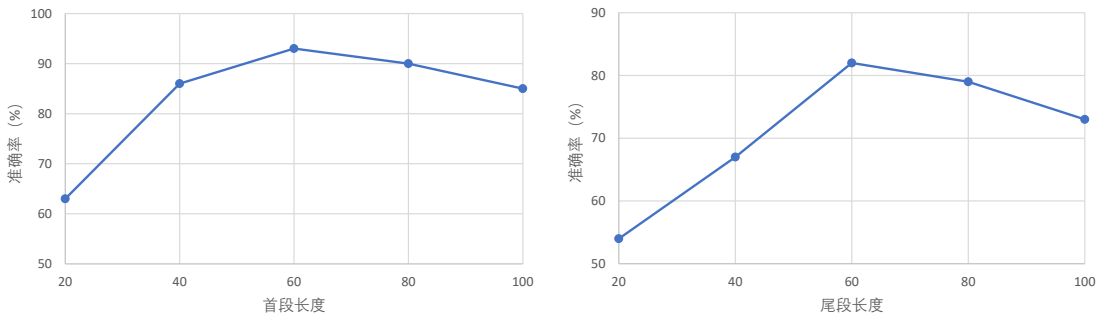
表 5.2 不同分割阈值下的识别结果

Table 5.2 Recognition results under different segmentation thresholds

	0	25	50	75	100	125	150	175	200
ACC	0.35	0.35	0.49	0.57	0.84	0.91	0.76	0.53	0.50

5.3.4 首尾段分割模块实验评估

图 5.8 所示，随着首段和尾段的增加，混合流量的识别结果先增加后下降，在段选择 60 的时候，达到最高点。段太小或者太大，识别结果都不好，主要原因在于段太小的话，包含的流量信息太少，难以准确的表征应用，提取有效的特征，导致识别准确率低。如果段太大，每个段里面就可能会受到他网络行为流量干扰，导致流量不纯净，识别结果不准确。因此选择段为 60 个数据包的时候，识别效果最好。



(a) 首段识别结果　　**(b) 尾段识别结果**

图 5.8 首尾段的分类结果

Figure 5.8 Classification results of the first and last segments.

5.3.5　端到端识别模块实验评估

端到端模块主要用于处理隧道内复杂的混合流量：正时间分离应用流量（前面分割失败的）、零时间分离应用流量和负时间分离应用流量。该模型对于隧道内高重叠率的识别效果更好。实验中，对隧道内零时间分离应用流量和负时间分离应用流量进行了测试，得出识别结果。

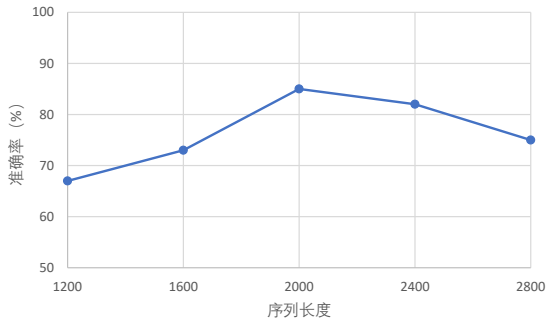


图 5.9 隧道混合流量的分类结果

Figure 5.9 Classification Results of Tunnel Mixed Traffic .

图 5.9 所示，端到端处理隧道内混合网络行为流量识别结果最高为 83%。与正时间分离应用流量相比，该类型的隧道混合流量识别结果略微下降。主要在于该模块处理的混合流量形式多样，存在的正时间分离应用流量是前面两级没有成功分割的混合流量，分割更难。而且负时间分离应用流量存在不同重叠率，差别较大。重叠率越高，隧道混合流量可以利用的纯净流量越少，使得隧道混合流量的识别更难。

5.3.6 分割决策模块实验评估

分割决策模型，主要用于识别上一级分割的结果，识别出不同类型的混合流量，并对不同类型的混合流量，选择性处理。对于分理处的单应用流量，可以直接输入分类器模型直接进行识别。对于处理后是应用两两混合的流量，输入首尾段分割模块进行处理。对于处理后仍是多应用混合的流量输入网络行为转换检测模块或者端到端识别模块。

Burst 分割后的决策模型，主要针对 Burst 分割后的混合流量，识别出三种类型的混合流量。实验中，设计了一个三元分类器，识别出单应用流量、两应用混合流量和多应用混合流量。在训练过程中，所有单应用流量标签标记为 0，两应用混合流量标签标记为 1，多应用混合流量标签标记为 2。为了减少样本不均衡对决策模型造成影响，对于三种类型的数据集（单应用流量、两应用混合流量和多应用混合流量采集相同的类别，保证这三种类型的数据集比例：单应用流量:两应用混合流量和多应用混合流量 =1:1:1。最后按照 7:3 的比例划分训练集，测试集。最后利用训练集和测试集对分割决策模型进行训练与测试。

表 5.3 分割决策识别结果

Table 5.3 The Results of Split Decision.

混合类型	精确率	召回率	F1
单应用纯净流量	0.93	0.97	0.95
两应用混合流量	0.85	0.97	0.91
多应用混合流量	0.70	0.84	0.76

分割决策结果如表 5.3 所示。总体来看，分割决策模型对于单应用纯净流量的识别效果最好，精确率达到 93%。其次是两应用混合流量的识别结果，精确率达到了 85%。最后是隧道内多应用混合流量的识别结果，精确率达到了 70%。单应用纯净流量识别效果最好，原因主要在于纯净流量没有受到其他流量的干扰，流量更加纯净，特征更加明显。两应用混合流量与单应用混合流量相比，识别结果略有下降，主要在于流量混合后，数据包长序列和数据包时间间隔发生了大的变化，而且不同重叠下流量差别也较大，因此识别结果略有下降。

5.3.7　应用识别结果

在多模块测试之后，本节对隧道内混合流量，从具体的应用组成上来分类这些混合流量究竟属于哪些应用。实验结果，如表 5.4 所示。总体看来，隧道内混合流量的识别精度大多数保持在 70% 以上。隧道内应用的识别结果差别较大，产生的原因是有两个方面的：

1. 隧道内应用的识别难度不同。隧道内一些应用的特征比较明显，识别的结果更高一些。一些应用的特征区别性不高，识别结果较低。这是隧道内应用自身原因导致的识别结果不同。

2. 隧道内应用的混合位置不同。如果隧道内应用的尾部存在混合，会对应用的识别影响不大，识别结果下降不明显。如果隧道内应用首部存在混合，则会对隧道内应用识别产生很大影响，识别结果急剧下降。

表 5.4 IPSec 下应用识别结果

Table 5.4 The Application Classification Results of IPSec.

IPSec							
应用	**精度**	**召回**	**F1**	**应用**	**精度**	**召回**	**F1**
dropbox	0.76	0.85	0.80	**fileZilla**	0.80	0.89	0.84
twitter	0.74	0.69	0.71	**skype**	0.83	0.92	0.87
facebook	0.81	0.75	0.78	**gmail**	0.90	0.78	0.84
outlook	0.83	0.88	0.85	**netflix**	0.79	0.83	0.81
youtube	0.87	0.90	0.88	**yahoomail**	0.77	0.87	0.82
linkedin	0.82	0.89	0.85	**servu**	0.81	0.89	0.85
instagram	0.61	0.72	0.66	**hulu**	0.84	0.86	0.85
icloudmail	0.90	0.87	0.88	**foxmail**	0.85	0.80	0.82
oneDrive	0.85	0.89	0.87	**michat**	0.79	0.85	0.82

5.4　本章小结

本章考虑真实环境中用户的使用特点，设计出由数据获取、网络行为分割和精细化识别组成的原型系统。在训练和测试阶段，利用该系统的数据获取，获取带标注的隧道内混合网络行为流量。预测阶段，获取真实隧道内的混合网络行为流量。在网络行为分割阶段，利用 Burst 分割、网络行为转换检测、首尾段分割、

端到端分割识别和分割决策的多级分割架构，对隧道内混合网络行为进行分割，提取出单应用流量，使得不同的应用流量在不同的数据流。实验结果表明，该系统可以有效的处理隧道内多种类型的混合流量。

第 6 章　总结与展望

隧道具有封装和加密特性，使得网络通信更加安全和便捷。隧道的出现，也给网络流量管理和安全防护带来很大的挑战。一些攻击者通过隧道和加密技术绕过防火墙和入侵检测系统，实施恶意行为。因此准确识别隧道内承载的网络行为，对于维护网络安全具有重要的意义。与加密网络行为识别相比，隧道内网络行为更加复杂，存在混合和重叠的问题，给网络管理者也带来很大的挑战。

6.1　本文工作总结

由于隧道的封装和加密特性，隧道内承载的所有网络行为都具有相同的协议和五元组信息，因此使得隧道内承载的不同网络行为无法利用明文分割流量，隧道流量存在混合和重叠的问题。本文从隧道内网络行为识别出发，考虑隧道内存在流量混合和重叠的特点，开展了隧道内混合网络行为识别的研究，最终完成了隧道内混合网络行为识别的原型系统。

1. **隧道内混合网络行为数据集的生成和分割**。考虑到隧道内网络行为存在混合和重叠的问题，为了生成带标注的隧道内混合网络行为数据集，构建了隧道回放和混合流生成的混合数据集生成方法。隧道回放方法利用加密网络流量数据集，通过回放到隧道生成隧道内混合网络行为流量，实现隧道内混合网络行为的标注。混合流生成方法利用时间维度信息，基于累计时间间隔信息生成隧道内混合网络行为数据集，可以实现包级别的混合网络行为数据集的标注。最后，针对隧道内两两应用混合的数据集，提出了首尾段分割的方法，该方法可以实现快速、高效和准确的混合网络行为识别。

2. **隧道内混合网络行为的精细化识别**。针对前面产生的数据集，提出了基于分割决策的混合流分割方法。该方法通过检测出网络行为发生转换的时刻，实现隧道内混合网络行为数据集的分割。实验结果表明，该方法可以有效的处理正时间分离应用流量。此外，还提出了基于端到端的混合网络行为数据集识别模型，该模型将混合网络行为分隔与识别集成一个模型中，实现隧道内混合网络行为的精细化识别。在四种隧道的多种类型的数据集上进行了测试，实验结果表明该方法在多种隧道上均实现了很好的识别效果。而且在 Tor 公开数据集上，也取

得很好的识别效果。

3. **隧道内混合网络行为识别原型系统**。本文考虑用户在真实隧道中的使用，基于前面章节的研究，提出了隧道内混合网络行为识别的院系系统。该系统包括数据获取、网络行为分割和精细化识别三个阶段。数据获取阶段，在训练和测试的时候，产生带标注的隧道内混合网络行为数据集。在预测阶段，收集隧道内内混合网络行为数据集。网络行为分割阶段主要包括 Burst 分割、网络行为转换检测、首尾段分割、分割决策和端到端识别。通过多级的选择性处理实现隧道内混合网络行为数据集的有效分割。精细化识别阶段对分割出的单应用流量，提取特征后，输入分类器进行处理。

6.2 未来展望

本文对隧道内混合网络行为识别进行了深入分析，并提出了隧道内混合网络行为识别的原型系统。虽然该系统在测试实验环境下实现了很好的识别效果，但是依然还有很多可以做的工作：

1. **隧道内高重叠率混合流量识别问题**。隧道内混合网络行为识别的难度，随着重叠率的增加，变得越来越大。重叠率越高，导致隧道内可以利用的纯净流量越来越少，混合流量识别变得更加困难。

2. **单隧道多人使用场景流量识别问题**。目前的工作仅做了隧道内单人使用的场景的识别，没有分析隧道内多人使用场景下的识别。针对于隧道内多人使用的数据，网络行为的混合更加复杂，对隧道内网络行为识别提出了更高的要求。

3. **隧道内真实环境中包级别标注问题**。为了定性分析原型系统在真实环境中的识别效果，未来，需要研究实现隧道内混合网络行为数据集的包级别标注问题。

参考文献

[1] Lakbabi A, Orhanou G, El Hajji S. Vpn ipsec & ssl technology security and management point of view [C]//2012 Next Generation Networks and Services (NGNS). IEEE, 2012: 202-208.

[2] Narayan S, Brooking K, de Vere S. Network performance analysis of vpn protocols: An empirical comparison on different operating systems [C]//2009 International Conference on Networks Security, Wireless Communications and Trusted Computing: volume 1. IEEE, 2009: 645-648.

[3] 王延年. 隧道技术及其应用研究 [J]. 中国优秀博硕士学位论文全文数据库, 2001.

[4] 贾永杰, 周秋剑, 李刚. VPN 隧道协议比较与分析 [J]. 空军雷达学院学报, 2003, 17(2): 33-35.

[5] 黄德奇. 基于 L2TP 协议的虚拟专用网络设计 [J]. 电子技术与软件工程, 2018(8): 16-16.

[6] Kumar J, Kumar M, Pandey D K, et al. Encryption and authentication of data using the ipsec protocol [C]//Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems. Springer, 2021: 855-862.

[7] 王笛, 陈福玉. 基于 IPsec VPN 技术的应用与研究 [J]. 电脑知识与技术: 学术版, 2020, 16 (11): 17-19.

[8] KeHe W, Peng Z, WenChao C, et al. Tunneling ssl vpn based on pf_ring [C]//2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2019: 1-4.

[9] 全国信息安全标准化技术委员会. 网络安全态势感知技术标准化白皮书 [M]. 2020.

[10] Top G. Strategic technology trends for 2020(2019) [Z]. 2020.

[11] Shi Y, Biswas S. A deep-learning enabled traffic analysis engine for video source identification [C]//2019 11th International Conference on Communication Systems & Networks (COMSNETS). IEEE, 2019: 15-21.

[12] Shi Y, Feng D, Cheng Y, et al. A natural language-inspired multilabel video streaming source identification method based on deep neural networks [J]. Signal, Image and Video Processing, 2021, 15(6): 1161-1168.

[13] Shi Y, Biswas S. Using traffic analysis for simultaneous detection of bittorrent and streaming video traffic sources [C]//2017 9th International Conference on Communication Systems and Networks (COMSNETS). IEEE, 2017: 79-86.

[14] 赵双. 基于多分类器融合的移动网络流量识别方法研究 [D]. 国防科技大学, 2018.

[15] Aceto G, Ciuonzo D, Montieri A, et al. Mobile encrypted traffic classification using deep

learning [C]//2018 Network traffic measurement and analysis conference (TMA). IEEE, 2018: 1-8.

[16] Aceto G, Ciuonzo D, Montieri A, et al. Mimetic: Mobile encrypted traffic classification using multimodal deep learning [J]. Computer networks, 2019, 165: 106944.

[17] Roughan M, Sen S, Spatscheck O, et al. Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification [C]//Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. 2004: 135-148.

[18] Rowan T. Vpn technology: Ipsec vs ssl [J]. Network Security, 2007, 2007(12): 13-17.

[19] Adeyinka O. Analysis of problems associated with ipsec vpn technology [C]//2008 Canadian Conference on Electrical and Computer Engineering. IEEE, 2008: 001903-001908.

[20] Leinwand A, Fang K. Network management: a practical perspective [M]. Addison-Wesley Longman Publishing Co., Inc., 1993.

[21] Ahmed W, Shahzad F, Javed A R, et al. Whatsapp network forensics: Discovering the ip addresses of suspects [C]//2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS). IEEE, 2021: 1-7.

[22] 杜臻, 马立鹏, 孙国梓. 一种基于小波分析的网络流量异常检测方法 [J]. 计算机科学, 2019, 46(8): 178-182.

[23] Zhao J, Jing X, Yan Z, et al. Network traffic classification for data fusion: A survey [J]. Information Fusion, 2021.

[24] Lamport L. Document preparation system [M]. Addison-Wesley Reading, MA, 1986.

[25] Shi Y, Ross A, Biswas S. Source identification of encrypted video traffic in the presence of heterogeneous network traffic [J]. Computer Communications, 2018, 129: 101-110.

[26] Shi Y, Biswas S. Protocol-independent identification of encrypted video traffic sources using traffic analysis [C]//2016 IEEE International Conference on Communications (ICC). IEEE, 2016: 1-6.

[27] Shi Y, Biswas S. Detecting tunneled video streams using traffic analysis [C]//2015 7th International Conference on Communication Systems and Networks (COMSNETS). IEEE, 2015: 1-8.

[28] Shi Y, Biswas S. Characterization of traffic analysis based video stream source identification [C]//2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS). IEEE, 2015: 1-6.

[29] McCarthy C, Zincir-Heywood A N. An investigation on identifying ssl traffic [C]//2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA). IEEE, 2011: 115-122.

[30] Pang Y, Jin S, Li S, et al. Openvpn traffic identification using traffic fingerprints and statistical characteristics [C]//International Conference on Trustworthy Computing and Services. Springer, 2012: 443-449.

[31] He Y, Li W. Image-based encrypted traffic classification with convolution neural networks [C]//2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC). IEEE, 2020: 271-278.

[32] Liu J, Li S, Zhang Y, et al. Detecting dns tunnel through binary-classification based on behavior features [C]//2017 IEEE Trustcom/BigDataSE/ICESS. IEEE, 2017: 339-346.

[33] Lin H, Liu G, Yan Z. Detection of application-layer tunnels with rules and machine learning [C]//International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage. Springer, 2019: 441-455.

[34] Cheng J, He R, Yuepeng E, et al. Real-time encrypted traffic classification via lightweight neural networks [C]//GLOBECOM 2020-2020 IEEE Global Communications Conference. IEEE, 2020: 1-6.

[35] Davis J J, Foo E. Automated feature engineering for http tunnel detection [J]. computers & security, 2016, 59: 166-185.

[36] MontazeriShatoori M, Davidson L, Kaur G, et al. Detection of doh tunnels using time-series classification of encrypted traffic [C]//2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). IEEE, 2020: 63-70.

[37] 余红星, 申国伟, 郭春. 一种基于自动特征工程与压缩感知的网络隧道检测方法 [J]. 计算机与现代化, 2019(6): 1-8.

[38] Ishikura N, Kondo D, Vassiliades V, et al. Dns tunneling detection by cache-property-aware features [J/OL]. IEEE Transactions on Network and Service Management, 2021, 18(2): 1203-1217. DOI: 10.1109/TNSM.2021.3078428.

[39] Leroux S, Bohez S, Maenhaut P J, et al. Fingerprinting encrypted network traffic types using machine learning [C]//NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2018: 1-5.

[40] Shapira T, Shavitt Y. Flowpic: Encrypted internet traffic classification is as easy as image recognition [C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2019: 680-687.

[41] Li Y, Lu Y. Multimodality data analysis in information security etcc: Encrypted two-label classification using cnn [J]. Security and Communication Networks, 2021, 2021.

[42] Lin K, Xu X, Gao H. Tscrnn: A novel classification scheme of encrypted traffic based on flow

spatiotemporal features for efficient management of iiot [J]. Computer Networks, 2021, 190: 107974.

[43] Guo L, Wu Q, Liu S, et al. Deep learning-based real-time vpn encrypted traffic identification methods [J]. Journal of Real-Time Image Processing, 2020, 17(1): 103-114.

[44] Cui S, Jiang B, Cai Z, et al. A session-packets-based encrypted traffic classification using capsule neural networks [C]//2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019: 429-436.

[45] 陈昱彤. 隧道内网络行为识别技术研究 [M]. 2021.

[46] Zhang J, Li F, Wu H, et al. Autonomous model update scheme for deep learning based network traffic classifiers [C]//2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6.

[47] Yao H, Liu C, Zhang P, et al. Identification of encrypted traffic through attention mechanism based long short term memory [J]. IEEE Transactions on Big Data, 2019.

[48] Zheng W, Gou C, Yan L, et al. Learning to classify: A flow-based relation network for encrypted traffic classification [C]//Proceedings of The Web Conference 2020. 2020: 13-22.

[49] Sun B, Yang W, Yan M, et al. An encrypted traffic classification method combining graph convolutional network and autoencoder [C]//2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC). IEEE, 2020: 1-8.

[50] 和留勇. 基于深度学习的 SSH 隧道下应用的精细化识别研究和实现 [D]. 北京邮电大学, 2018.

[51] Korczyński M, Duda A. Markov chain fingerprinting to classify encrypted traffic [C]//IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014: 781-789.

[52] Shiming T, Feixiang G, Shuang M, et al. End-to-end encrypted network traffic classification method based on deep learning [J]. The Journal of China Universities of Posts and Telecommunications, 2020.

[53] Shi Y, Biswas S. Website fingerprinting using traffic analysis of dynamic webpages [C]//2014 IEEE Global Communications Conference. IEEE, 2014: 557-563.

[54] Juarez M, Afroz S, Acar G, et al. A critical evaluation of website fingerprinting attacks [C]// Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 263-274.

[55] Wang T, Goldberg I. On realistically attacking tor with website fingerprinting. [J]. Proc. Priv. Enhancing Technol., 2016, 2016(4): 21-36.

[56] Xu Y, Wang T, Li Q, et al. A multi-tab website fingerprinting attack [C]//Proceedings of the 34th Annual Computer Security Applications Conference. 2018: 327-341.

[57] Cui W, Chen T, Fields C, et al. Revisiting assumptions for website fingerprinting attacks [C]// Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. 2019: 328-339.

[58] Gu X, Yang M, Luo J. A novel website fingerprinting attack against multi-tab browsing behavior [C]//2015 IEEE 19th international conference on computer supported cooperative work in design (CSCWD). IEEE, 2015: 234-239.

[59] Guan Z, Xiong G, Gou G, et al. Bapm: Block attention profiling model for multi-tab website fingerprinting attacks on tor [C]//Annual Computer Security Applications Conference. 2021: 248-259.

[60] Taylor V F, Spolaor R, Conti M, et al. Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic [C]//2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016: 439-454.

[61] Panchenko A, Lanze F, Pennekamp J, et al. Website fingerprinting at internet scale. [C]//NDSS. 2016.

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历:

赵盼盼，男，河南省商丘市人，1996 年出生，中国科学院信息工程研究所硕士研究生。

2015 年 9 月-2019 年 6 月，在郑州大学信息工程学院就读，获得了学士学位。

2019 年 9 月-2022 年 6 月，在中国科学院信息工程研究所攻读硕士学位。

已发表（或正式接受）的学术论文:

1. **Panpan Zhao**, Gaopeng Gou, Chang Liu, Yangyang Guan, Mingxin Cui,Gang Xiong. TMT-RF: Tunnel Mixed Traffic Classification Based on Random Forest[C]// Security and Privacy in Communication Networks, 2021. (SecureComm 2021, CCF-C).
2. **Panpan Zhao**, Zhen Li, Mingxin Cui, Ji Lu, Gang Xiong, Gaopeng Gou. TunnelScanner: A Novel Approach For Tunnel Mixed Traffic Classification Using Machine Learning[C]// High Performance Computing and Communications, 2021. (HPCC 2021, CCF-C).

申请或已获得的专利:

1. 熊刚，管洋洋，赵盼盼，崔明鑫，苟高鹏，李镇.《一种基于随机森林的隧道混合流量分类方法及系统》，申请中

参加的研究项目:

1. 2019.3-2019.9，基于 wireshark 的网络协议深度解析系统设计与开发。通过 Lua 语言插件解析、源码编译等实现新型协议解析解析和隧道协议的数据流还原重组。

2. 2020.6-2020.10, 隧道回放。通过搭建隧道，使用 Scapy 库、Tcpreplay 等实现将加密应用流量回放进隧道生成隧道流量。

3. 2020.9-2021.9，隧道内混合网络行为识别。开发了隧道内混合网络行为识

别框架，首先将隧道内混合网络行为分割，提取单网络行为流量，然后再进行识别。

研究生期间获奖情况

- 2021 年 11 月，获中华人民共和国教育部颁发的国家奖学金
- 2021 年 6 月，被中国科学院大学授予三好学生荣誉称号
- 2020 年 6 月，被中国科学院大学授予优秀学生干部荣誉称号
- 2020 年 5 月，被中国科学院大学授予优秀共青团干部荣誉称号
- 2019 年 12 月，获中国科学院大学颁发的中国科学院大学大学生奖学金
- 2019 年 11 月，获北京市怀柔区团委颁发的志愿服务证书

致 谢

时光飞逝，光阴荏苒。当写到这部分的时候，研究生生涯也即将结束。回想起 2019 年入所做本科毕业设计，到如今做研究生毕业设计，已有三年的时间。这三年时间让我明白一个道理“世上无难事，只要肯攀登”。这三年时间让我从网络行为分析方向的小白，到如今掌握扎实的专业知识。在这三年中，很多人给予了我很大的帮助，仅以此部分对他们表示感谢。

首先感谢我的导师苟高鹏老师。在这三年里，苟高鹏老师在学习和生活上给予了我无限的指导和关怀。在科研学习方面。在刚入组时候，苟高鹏老师与我深刻的探讨研究内容。在做实验阶段，苟高鹏老师时刻关注我的实验进度，给予了很大的理论指导。在论文撰写阶段，通宵达旦的指导我修改学术论文。读书期间，父亲生病，家庭压力巨大。苟高鹏老师时刻关心我的家庭情况，及时做好我的心理疏导，给予了我很大的帮助。在此，向苟高鹏老师说一声：您辛苦了！同时也要感谢我的指导老师熊刚老师。熊刚老师在我做我本科毕业设计的时候，给与了无限的帮助。熊刚老师兢兢业业的工作态度，也激励着我不断努力，实现自我超越。最后还要感谢刘畅老师、崔明鑫老师、李镇老师和侯承尚老师对我的帮助和支持，在此向各位老师表示最诚挚的敬意与感谢。

其次要感谢组内的陈曼菁、李思佳、王晨成同学，我们在一起做了本科毕业设计和研究生毕业设计，感谢您们的陪伴。同时要感谢陆杰、谭曰文、杨琛同学。在被人误解的时候，是你们始终默默的支持我，及时做好我的心理疏导，使我身心没受太大的影响。特别要感谢我的室友金泽文同学，在我忙着跑实验的时候，帮我承担宿舍的卫生清理等工作。感谢管中师兄、王宇师兄、田婧师姐和陈昱彤师姐，感谢你们在学习和生活上的帮助。最后还要感谢顾哲媛、夏耀华、郭敬宇师妹，感谢你们在学习上的帮助和支持。

然后要感谢父母，你们是我坚强的物质支持和精神支撑。你们身体不好，但仍然奋斗在工作一线，起早贪黑，全力支持我读研究生，没有你们的支持，我不能走到今天。感谢我的姥姥，您时刻教诲我生活学习的方方面面，您的“人生世间，时刻要怀三心”更是让我受用终身：一曰孝心，二曰善心，三曰平心。还要特别感谢我的姐姐：赵楠楠。我自幼性格内向，成绩落后，内心也比较自卑。自

从中学和老姐相处以来，性格慢慢的从内向走向外向、内心慢慢的从自卑走向自信、成绩也慢慢的提上来了，最终考上大学，再到保送研究生。而且老姐总是时刻能够注意到我情绪的变化，能够及时开导我，让我能够全身心投入到学习中去，很幸运遇见老姐。感谢我的弟弟，你的到来，给我小时候的带来了一些陪伴，让我也感受到兄弟姐妹的关心，给我带来了很多快乐。

最后，感谢我的爷爷奶奶。初中期间，家中生活水平不好，为了不让我在晚自习后不饿肚子，爷爷奶奶捡破烂给我买方便面吃，你们大早上在垃圾坑前捡垃圾的场景，我还历历在目。你们的鼓舞和激励，使我从差生（班级后几名）慢慢的变成班级前几名。你们在生命最后一刻叮嘱孙子的话语，孙子必定牢牢永记：好好学习，日日上进。虽然您们没有陪着我走到研究生生涯，但您们的教诲永远使我能够迎难而上，奋勇向前，感谢我的爷爷奶奶。感谢我的老爷爷和老奶奶（爸爸的爷爷、奶奶）。我是最幸运的人，从小在你们的关心和爱护下长大。你们陪伴我走到了大学，你们八十多年来的相濡以沫、互敬互爱的爱情故事，给我们后辈树立了一个好的榜样。您们面对一切困难的豁达态度、心纳大海的胸怀气度和长寿的养生之道（老奶奶 100 岁、老爷爷 97 岁），更是我以后需要努力学习的。感谢我最亲爱的芳芳姐，您虽然很早就离开了我，去了另一个世界。但是小时候您的照顾关爱和留下的书信，一直是我长大后面对各种困难最强大的力量。

“长风破浪会有时，直挂云帆济沧海”。在您们的帮助和支持下，我即将完成研究生的学业，走向社会，走向工作岗位。在将来的工作和学习中，我会时刻牢记你们的帮助和支持，继续艰苦奋斗、努力拼搏、行稳致远，做一个对民族、对国家、对社会有用的人才。