西安交通大學

硕士学位论文

基于流量认知分析的无线设备指纹识别技术研究与实现

学位申请人：余天文

指导教师：沈超　副教授

学科名称：控制科学与工程

2018 年 05 月

A Wireless Device Identification System Based on Traffic Analysis and Cognition

A thesis submitted to
Xi'an Jiaotong University
in partial fulfillment of the requirements
for the degree of
Master of Engineering Science

By
Tianwen Yu
Supervisor: Associate Prof. Chao Shen
Control Science and Enginnering
May 2018

论文题目：基于流量认知分析的无线设备指纹识别技术研究与实现
学科名称：控制科学与工程
学位申请人：余天文
指导教师：沈超　副教授

摘　要

随着无线网络的飞速发展和智能移动设备的日益普及，无线网络技术在给人们日常生活带来方便的同时，也存在着许多安全隐患。访问控制策略是网络安全防范和保护的主要策略，其目的是保证网络资源不被非法使用和非法访问。对无线网络设备进行精确识别在网络访问控制和网络攻击防御中起着至关重要的作用。当前的网络访问控制机制大多使用数字证书或标识符来识别设备的身份，如用户名和密码、MAC 地址、SSL 证书等。然而用户名和密码存在着泄露的风险，MAC 地址容易被篡改和伪造，SSL 证书也有被破解的可能，一旦设备身份被破坏，将会给用户带来巨大的损失。针对上述问题，本文提出了基于流量认知分析的无线设备指纹识别技术，可以在不影响设备正常使用的情况下，实现对无线设备的精确识别。

文章中首先深入分析了无线数据帧与设备个体之间的相关性，提出了基于流量认知分析的无线设备指纹识别方案。接着，捕获到包含个人 PC、智能手机、平板电脑在内的 23 台无线设备的流量数据，创建了设备指纹识别的数据集。本文从 TCP 流量数据帧中提取出帧时间间隔、帧大小和传输速率三种参数，对每种参数的数据进行降噪和归一化处理，将数据转换成易于提取特征的格式。本文提出了两种特征指纹生成方法：基于概率密度的特征指纹和基于多特征融合的特征指纹。基于概率密度的特征指纹是统计每种参数的概率密度，将其作为特征，三个参数分别形成三类特征指纹。多特征融合的特征指纹是将三个独立的特征指纹组合在一起，以形成能够更加完整表征设备身份信息的特征指纹。

针对无线设备识别问题，本文使用随机森林（RF）、支持向量机（SVM）、K 最近邻（KNN）、朴素贝叶斯（NBC）四种分类器进行设备身份模型的构建与评估，并采用准确率（precision）、召回率（recall）和 F1 值来评估识别的效果。本文分别对两种特征指纹和四种分类器的识别性能进行比较，实验结果表明：融合特征在标识设备身份方面表现更好，在各种分类器下均能取得较好的效果；在四种分类器中，随机森林更适合无线设备指纹识别问题，针对每种特征指纹均能得到较好的识别精度。当使用融合特征指纹时，随机森林分类器的 precision、recall 和 F1 值分别为 0.9930、0.976 和 0.9783。本文还探讨了特征空间的变化对识别效果的影响，即更改特征指纹生成过程中的部分参数，比较各分类器的效果和两种特征指纹的识别性能。实验结果表明，随机森林的分类效果比较稳定，始终能保持较好的识别结果；相比于基于概率密度的特征指纹，融合特征指纹的鲁棒性更好，当特征指纹生成过程中的参数变化时，融合特征仍能较好地

标识设备的身份。

基于上述提出的无线设备指纹识别方法，本文设计并实现了基于 B/S 架构的无线设备识别原型系统。原型系统包含流量数据捕获、特征指纹生成、无线设备识别和指纹库更新四个功能模块。其中流量数据捕获模块用于实时捕获流量数据；特征指纹生成模块负责特征参数的提取、数据降噪和归一化，并提取出相应的特征指纹；设备识别模块将设备的特征指纹与指纹库中的指纹进行比对，识别出设备类型；指纹库更新模块根据识别的结果将新设备指纹加入指纹库中。

关 键 词：网络流量分析；无线设备识别；设备指纹；分类器

论文类型：应用研究

Title: A Wireless Device Identification System Based on Traffic Analysis and Cognition
Discipline: Control Science and Engineering
Applicant: Tianwen Yu
Supervisor: Associate Prof. Chao Shen

ABSTRACT

With the rapid development of wireless network and the increasing popularity of smartphone, wireless network technology brings convenience to people's daily life, but there are many security risks. Access control policy is the main strategy for network security protection， its purpose is to ensure that network resources are not used illegally. Accurate identification of wireless network devices plays a crucial role in network access control and cyber attack defense. Current network access control mechanisms rely exclusively on the use of digital tokens or identifiers to recognize the device's identity, such as usernames and passwords, MAC address, SSL certificates and so on. However, there is a risk of leakage of usernames and passwords, MAC address is easily tampered and forged, SSL certificate may also be cracked. Once the identity of the device is destroyed, the user will suffer huge losses. In order to strengthen the security performance of the wireless network, a novel wireless fingerprint identification technology based on the network traffic is proposed. It can recognize the identity of the device without affecting the equipment under the condition of normal use.

Firstly, we analysed the correlation between wireless data frame and device individual, then developed a wireless device fingerprint recognition scheme based on traffic recognition. In order to verify the feasibility of the scheme, we captured the traffic data of 23 wireless devices including personal PC, smartphone and iPad, and created the data set of device fingerprint identification. Interval-arrive time, frame size and transmission rate were extracted from the TCP frame, and these data were converted into the form of easy-to-extract after noise reduction and normalization. Two characteristic fingerprints generation methods were proposed in this paper: probability density based fingerprint and multi-characteristic fusion based fingerprint. The probability denisty based fingerprint generated three types of feature, each feature was obtained by counting the probality of the parameter. The multi-characteristic fusion based fingerprint is the combination of three independent characteristic which can more fully characterize the device identity information.

For the wireless devices identification problem, this paper used four classification algorithms to construct and evaluate the device recogntion model: random forest (RF), support vector machine (SVM), K nearest neighbor (KNN), naïve bayes (NBC). Precision, recall and F1 value were used to assess the effectiveness of the recognition approach. We compared the performance of two characteristic fingerprint and four classifier, and the experimental results showed that the fusion fingerprint performed better in identifying device and achieves better

results under four classifiers. Among four classifiers, random forest was more suitable for identification of wireless devices, the precision, recall and F1 values of the random forest classifier were 0.9930, 0.976 and 0.9783 when the fusion fingerprint was used. The influence of the change of feature space on the recognition effect was also discussed, that is, change some parameters in fingerprint generation process and compare the effects of each classifier and the performance of two fingerprints. The experimental results show that random forest can get stable performance with high precision. In comparison with probability density based fingerprint, the robust of fusion based fingerprint was better, when the parameters changed, the fusion feature can still identify the device well.

Based on the proposed wireless device fingerprint identification approach, this paper designed and implemented a wireless device recognition prototype system based on B/S architecture. The prototype system included four functional modules: traffic data capture module, fingerprint generation module, wireless device identification module, and fingerprint library update module. The traffic data capture module was used to capture traffic data in real time; the feature fingerprint generation module is responsible for the extraction of feature parameters, data denoising and normalization, and extracting corresponding fingerprints; the device identification module identified the identity of the device by classification algorithm; the fingerprint library update module added the new device fingerprint to the fingerprint library according to the result of the recognition.

KEY WORDS: Network Traffic Analysis; Wireless Device Identification; Device Fingerprinting; Classifier

TYPE OF THESIS: Application Research

目 录

CONTENTS

1 绪论

1.1 选题意义与应用背景

随着互联网的普及和 Wifi 技术的发展，无线/移动通信网络已经成为当今社会的基础设施，广泛地应用于政府、金融、军事等领域。统计显示，截止到 2017 年 12 月，中国网民规模达 7.72 亿，互联网普及率达到 55.8%；与此同时，我国手机网民规模达到 7.53 亿，网民中使用手机上网的人群占比由 2016 年的 95.1%提升至 97.5%[1]。无线通信在日常生活中发挥着不可替代的作用，已成为现代社会不可或缺的一部分。然而，无线网络的普及在给人们生活带来便利的同时，也存在很多安全隐患。相比于有线网络，无线网络由于其开放性更容易遭受到恶意攻击[2]。隐私泄露、病毒入侵、网络欺骗、假冒攻击等问题时刻威胁着网络用户的安全。

传统的保护无线网络安全的方法通常是基于比特层面的(即 OSI 七层模型中物理层以上的层次)，通过设计基于密码机制的安全协议来实现对数据完整性和机密性的保护以及提供通信双方身份的认证[2]。然而，无线网络安全协议通常会存在安全漏洞[3]。例如，IEEE 820.11 无线局域网(WLAN)最初的有线等效加密(WEP)协议易受统计分析攻击[4]，虽然此后升级为 WPA 和 WPA2，但其口令句可以被恢复，仍然存在着各种各样的安全问题[5]。此外，密码机制也存在着密钥泄露的风险，攻击者会通过多种方式获取用户的 ID 和密码，然后伪装成用户进行登录，给用户造成极大的损失。因此，人们急需寻找一种能够有效识别授权用户和非授权用户的安全机制,从而降低来自恶意用户的潜在威胁[2]。值得庆幸的是，攻击者可以通过多种方式获取用户的账号密码，甚至模拟用户的操作，但却很难伪造出与用户相似的设备特征。

在过去的十几年里，无线网络设备指纹识别问题得到了国内外广泛的关注，研究者们希望在无法识别无线设备用户的情况下，从无线设备入手，通过分析无线设备独特的行为模式特征，准确识别设备。MAC 地址常被用于设备的标识，MAC 地址就像用户的身份证号码一样，对每台设备都是独一无二的，因此有很多研究者提出可以根据 MAC 地址对设备进行标识。根据 MAC 地址识别设备的缺陷在于，攻击者很容易通过网络攻击工具篡改或伪造 MAC 地址，将自己伪装成合法的设备接入网络，且不易被检测和发现。

基于以上缺陷，研究者们尝试寻找设备独一无二的属性，这些属性就如同生物技术中的基因或者指纹一样，可以唯一地标识设备，且独立于用户在设备上使用的网络协议和应用类型，不随时间变化、不易被篡改或者伪造。分析流量的生成和传输机理后可以发现，网络流量的产生需要多个设备组件的配合，如网络适配器、处理器、DMA 控制器和网卡等。不同的厂商、不同的设备类型在这些硬件上都会存在差异，而这种差异会反映在网络流量中。通过分析不同设备在网络流量中的差异可以寻找到类似生物指纹一样的设备指纹，用于设备独一无二的标识。

基于网络流量认知的无线设备识别技术通过捕获设备的网络流量，并从中提取出设备指纹用于设备的标识。整个过程是隐式进行的，不会干扰设备的正常使用，也无需用户的参与。选择这一研究方向的实际意义在于寻找一种被动式的无线设备指纹识别方案，通过被动地捕获网络流量，从中提取出每个设备的特征指纹，进而实现对网络设备的精确识别。无线网络设备指纹识别技术可应用于网络接入控制、设备真伪鉴别、网络罪犯追踪等领域，可以作为传统保障网络安全技术手段的加强，提高无线网络的安全性能。

1.2 设备识别研究现状

“指纹”最早是指标识个体的生物特征识别技术[6,7]。这一概念早在上世纪 60 年代就被应用于设备识别，当时研究者们开发出通过观测信号特征来区分雷达的系统[8]。此后类似的技术被用于蜂窝网络中的发射器识别[9,10]。近年来，随着对无线网络设备研究的深入，对无线/移动网络设备“指纹”的分析技术也越来越受到研究者的重视，展现了广阔的应用前景[11]。

设备指纹是指可用于唯一标识出该设备的设备特征或者独特的设备标识。目前较普遍的技术是针对设备提供的某些信息，从中提取特征生成近似唯一的设备指纹，与存储的可信信息匹配辨识，从而对设备的身份进行认证和辨识。其原理是不同厂商生产的设备（即便是同一厂商生产的设备）会由于硬件或软件的不一致在某些特定的方面存在些许差异。当前研究的技术手段可大致分为基于硬件的识别和基于软件的指纹识别。基于软件的识别通常是研究从软件信息中提取的设备指纹，如根据 MAC 地址和浏览器等进行识别，而基于硬件的识别往往是探究设备身份与硬件因素之间的关系，最常见的是研究设备与时钟偏移之间的关系。

1.2.1 基于软件的设备识别

802.11 MAC 帧的格式和内容经常被用于辨识无线设备。MAC 地址是 Medium/Media Access Control 地址的简称，表示互联网上每一个站点的标识符。无线网络中对 AP 的识别通常利用网络协议中常见的标识符，例如网络名称、MAC 地址或者 IP 地址，但是这些标识符很容易被伪造、拦截或者更改。

Guo 等人根据 MAC 帧头的信息追踪设备[12]。顾杨等人利用从 MAC 层中的管理帧提取的特征作为区分真假无线接入点（Access point，AP）的特征指纹[13]。他们在文章中提出两种提取特征的方法：一种是扫描无线网络获得的，这种特征指纹的提取是被动的，即捕获到网络流量后，从信标帧和探测响应帧中获取 MAC 地址，构成无线 AP 的特征指纹；第二种指纹提取方式是基于主动式的，即发送请求刺激无线 AP，从 AP 的响应中提取特征指纹。作者在 3 种网络环境下测试第一种特征指纹的性能，又从 10 个 AP 中提取第二种特征指纹进行实验，实验的结果表明两种方式提取的特征指纹可以识别出大部分的 AP。但是，MAC 地址是与网卡对应的，如果一个设备拥有不止一个网卡，这个设备就会有相应数目的 MAC 地址，此时 MAC 就不能用来唯一地标识设

备；另外，如同居民的身份证容易被人盗用或者伪装，设备的 MAC 地址也容易被篡改或者伪装。

鉴于使用 MAC 的识别技术中存在上述缺陷，研究者们提出了一系列根据其他特征构建指纹的方法。Desmond 等人仅通过研究不同设备对 802.11 请求探测帧响应的时序特性，从中提取相应的指纹可识别出连接在同一无线接入点上的设备[14]。Pang 等人也能从流量中提取出特征指纹，进行设备的认证与标识[15]。Seika[16]等人向 802.11 设备发送一个信号，捕获响应帧的到达时间并用时序分析研究其规律，他们用 SVM 分类器建立目标设备模型，取得了较好的结果。

Gao 等人[17]提出了一种用于确定接入到网络的 AP 的类型的基于黑盒的被动式识别技术。作者进行了大量的实验（收集超过 60GB 的数据）以对 6 种 AP 进行分类，在至少 100000 个数据包的数据基础上取得了较高的分类精度。

此外，一些开源的工具如 Nmap[18]和 Xprobe[19]可以通过 TCP/IP 协议栈的响应识别设备的操作系统，据此进行设备识别。其他基于软件的指纹识别技术则是根据设备上的应用如浏览器来确定设备，不过基于浏览器的智能终端识别的研究主要集中在桌面浏览器上，对移动设备并不适用。Eckersley 在 2010 年最早研究了浏览器指纹，他设计的浏览器特征可从物理层、应用层和用户层数据中提取[20]。作者通过分析大量数据证实了浏览器指纹的有效性，同时也指出无线移动端设备（手机、iPad 等）的浏览器识别较为困难。

在 Eckersley 之后，也有许多关于桌面浏览器识别的研究。Yen 等人[21]通过分析从 Hotmail 网络邮件服务和 Bing 搜索引擎收集的长达一个月的日志，进行用户的追踪。Mowery[22]等人提出了两种形成浏览器指纹的方式：第一种方法根据每个浏览器 JavaScript 引擎的固有性能构成设备签名，即使传统形式的系统标识被修改或隐藏，这种特征指纹依然可以检测出浏览器版本和设备的操作系统；第二种方式通过确定用户的白名单中是否存在特定的域，从而有选择地启用网页的脚本权限以增加隐私。作者在亚马逊土耳其机器人平台上验证了该指纹识别技术的有效性。Acar 等[23]人设计了基于 Web 的指纹识别框架，他们通过对浏览器字体的研究来进行指纹的提取。此外，浏览器的浏览历史也被用于追踪 Web 端的用户[24]。

1.2.2 基于硬件的设备识别

Bratus 等人[25]提出了一种主动式设备指纹识别方法,该方法通过向 802.11 无线设备发送一系列的经过特殊构造的某种错误格式的数据帧来观察设备的回应，通过这些回应的差异可以区分设备在芯片、固件或者驱动上的差异，以此达到设备识别的目的。同时，该方法可以作为已有指纹识别技术的补充，用于区分合法的 AP 和钓鱼 AP。

Radhakrishnan 等人设计了一种名为 GTID 的识别框架，该技术采用主动和被动结合的方式，不仅可识别物理设备，还可以判断出相应的设备类型[26]。GTID 实现的主要依据是设备之间的异构性，他们认为不同的设备的处理器、DMA 控制器和时钟偏移等都存在差异，而这些差异可以被用于识别设备和设备类型。GTID 通过被动地抓取网络

流量，从中提取出相应的指纹特征并使用人工神经网络（Artificial Neural Networ，ANN）的算法进行训练和测试，取得了良好的效果。他们的技术适用于各种网络协议，且不需要进行深度包检测，但是由于该技术依赖于细粒度包时间，在路由器和交换机的缓冲区有时间丢失，故该技术的应用还有待进一步的研究。

基于硬件的指纹识别技术依赖于某些稳定的特性，论文[27]中的研究表明网络设备往往具有稳定的时钟偏移。Kohno 和 Cristea 等人的研究便是基于此，他们通过分析 TCP 和 ICMP 中微小的时钟偏移，从而构建设备的特征指纹[28,29]。该技术无需对设备做任何修改，即便目标设备通过多种方式接入互联网，或是测量设备与目标设备相距数千英里，仍能达到较好的识别结果。

然而，时钟偏移率很大程度上依赖于实验环境[30]。在论文[31]中作者讨论了基于时钟偏差的无线设备识别的局限性，他们利用无线接入点在信标帧中定期发送的时间戳为依据进行识别，消除了测量设备对时钟偏移的影响。此外，他们还进行了大量的评估，以探讨不同接入点和测量装置之间时钟偏差的分布及稳定性，发现时钟偏差的波动仅为 1ppm。该算法能够消除测量装置对实验结果的影响，使不同设备生成的指纹可比较且具有区分性。

与仅使用时钟偏斜作为特征指纹不同的是，Neumann C 等人[32]分别以传输速率、帧大小、介质访问时间、传输时间、包内间隔时间等特征作为特征指纹进行 802.11 设备指纹识别，并比较这些特征在设备识别中的性能差异，实验结果表明传输时间和包内间隔时间的性能要优于其他参数，为其他的设备指纹识别工作提供了很好的借鉴。

Franklin 等人提出了一种指纹识别技术，可以快速准确地找到无线设备驱动程序在真实无线网络环境下的指纹 [33]。作者指出不同的无线网卡在扫描无线网络时发出的探测帧会有所不同，因为 IEEE 802.11 协议中并没有规定扫描的算法，这主要取决于无线网卡的驱动程序，因此可以通过分析终端传输信标帧的间隔时间提取设备指纹。这种方法的缺陷在于数据获取困难，通常情况下一个无线网卡加入网络时只发送极少数的请求，想要获取足够的数据量需要大量的时间。

Gerdes 等人提出了基于模拟信号的终端设备识别技术，通过分析由于设备制造好硬件组件不一致造成的模拟信号的变化，可以唯一地标识和追踪以太网设备[34]。该技术仅需少量的数据即可识别出设备，虽然实验结果较好，但他们的研究仅是针对有线设备而言，这种方法对无线网络设备是否可行还存在疑问。而且这种方法依赖于如模数转换器和数字取样示波器等昂贵的设备，因此实用性不高。

此外，还有许多人研究了利用设备的辐射测量量作为指纹的技术[35,36,37,38]。其原理是天线、功率放大器、ADC、DAC 等硬件在生产过程中不可能完全相同，每个设备因此会具有一系列独一无二的辐射信号，如振幅、频率、相位等，从中可提取出特征指纹。由于这些辐射测量量在设备生产后无法改变，所以使用起来安全可靠。这种方法最大的局限性在于数据采集困难，只能在有限的范围内获取到设备的指纹，无法进行远程追踪或认证。

Anupam Das[39]等人认为厂商在制造智能手机的麦克风和扬声器时存在不同程度的缺陷，通过分析不同智能手机上的麦克风和扬声器的声学特征可以从中提取到设备相应的指纹。他们的方法虽然在 50 个安卓手机上达到了 98%的精度，但是必须在手机的麦克风或扬声器发声的时候才能进行测试，实现复杂且无法远程监控。

在论文[40]中，作者认为智能手机和平板电脑中的加速度计具有独特的指纹，可以用于设备识别。他们在 25 个安卓手机和 2 个平板电脑上验证这一方法，识别精度达到 96%。这种方法虽然精度高，但是要求设备上必需要有加速度传感器，且需要某种形式的外部刺激/震动来捕获加速度计的数据，局限性较大。相比而言，本文的工作只需远程捕获网络流量数据，不会被用户发现，操作简单，可实现性高。

1.3 本文的研究内容

本文以无线网络安全分析为研究背景、以网络中最为基础的流量数据为研究对象，提出了一种基于流量认知分析的无线设备指纹识别技术，用于加强无线网络的安全性能。这种技术通过捕获无线设备的网络流量，从中提取出能够表征设备身份信息的特征指纹，并使用模式识别的方法对特征指纹的有效性进行评估。基于流量认知分析的无线设备指纹识别技术无需用户的参与，可以在不被用户发现的情况下实现对无线设备身份的隐式识别。同时流量中提取的特征指纹不易被伪造和修改，安全性高且识别结果精确可靠。基于流量认知的无线设备识别方法流程如图 1-1 所示。

本文首先对基于流量认知分析的无线设备指纹识别原理进行讨论，从网络流量帧的生成和传输过程出发，分析网络流量与设备的硬件组成及控制算法之间的相关性，在此基础上设计了基于流量认知分析的无线设备指纹识别框架，该框架主要分为数据采集、数据预处理、特征指纹生成和无线设备识别几个部分。

我们首先搭建了一个无线局域网，用 Wireshark 捕获接入到局域网内的无线设备网络流量。我们共捕获到包含个人 PC、智能手机、平板电脑在内的 23 台无线设备的流量数据，每台设备的数据帧数目都达到了 50 万条以上，这 23 台设备的流量数据构成了本文研究的数据集。

捕获到的网络流量不能直接用于构建特征指纹，需要先进行预处理操作，将数据转换成易于提取特征的格式。根据设备指纹的特点，本文从数据包中过滤出 TCP 流量帧的数据，并从每条 TCP 协议数据帧中提取出帧间隔时间（IAT）、帧大小（FrameSize）和传输速率（TransRate）三个参数的数据，用概率密度（probability density function，PDF）曲线来查看每种参数数据的分布情况。观察 PDF 曲线可以发现，每种参数的数据中均含有一定的噪声数据，会干扰设备识别的结果。采用自定义区间的方法对数据进行降噪，可以过滤掉绝大部分的噪声数据，仅保留取值合理的特征参数。此外，提取到的三个参数量纲差异极大，降噪后的 IAT 的取值在 0 到 0.2 之间，而传输速率的取值则在 0 到1.2×10^{6}之间。防止量级影响数据分析的结果，我们使用 min-max 标准化方法对数据进行归一化处理，使得三种参数的数据向量均落在[0,1]区间内。

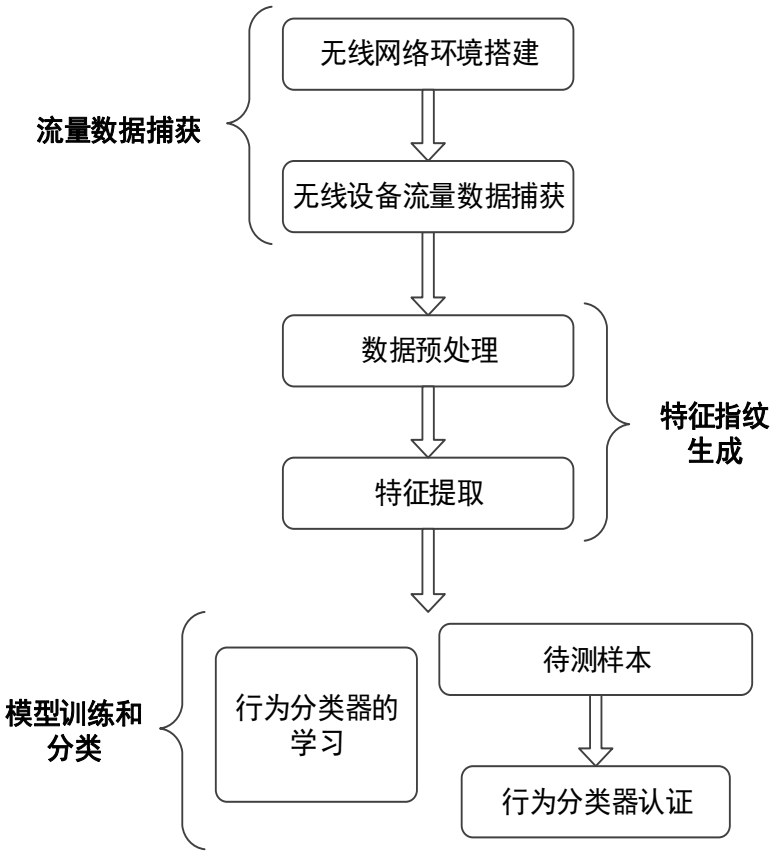

图 1-1　基于流量分析认知的无线设备识别流程

本文提出两种特征指纹生成方法：基于概率密度的特征指纹和基于多特征融合的特征指纹，两种特征指纹均可独立使用。基于概率密度的特征指纹是统计每种参数的概率密度作为特征，三个参数分别形成三类特征指纹。考虑到一种特征也许不能完全表征设备的身份，独立的特征可能仅包含了代表设备身份某一方面的信息，我们提出将三个独立的特征指纹结合在一起，形成基于多特征融合的特征指纹。

针对无线设备识别问题，本文使用随机森林（RF）、支持向量机（SVM）、K 最近邻（KNN）、朴素贝叶斯（NBC）四种分类器进行设备身份模型的构建与评估，并采用准确率（precision）、召回率（recall）和 F1 值来评估识别的效果。横向和纵向对比实验结果可以发现：相比于基于概率密度的特征指纹，融合特征指纹在标识设备身份方面表现更好，在各种分类器下均能取得较好的效果；在四种分类器中，随机森林更适合无线设备指纹识别问题，无论使用哪种特征指纹均能得到较好的识别精度。当使用融合特征指纹时，随机森林分类器的 precision、recall 和 F1 值分别为 0.9930、0.976 和 0.9783。本文还探讨了特征空间的变化对识别效果的影响，即更改特征指纹生成过程中的部分参数，比较各分类器的效果和两种特征指纹的性能。实验结果表明，随机森林的分类效果比较稳定，始终能保持较好的识别结果；相比于基于概率密度的特征指纹，融合特征指纹的鲁棒性更好，当特征指纹生成过程中的参数变化时，融合特征仍能较好地标识设备的身份。

基于上述提出的无线设备指纹识别技术，本文设计并实现了基 B/S 架构的无线设

备识别原型系统，该原型系统集成了无线设备指纹识别方案中的各个模块，可在线捕获移动设备的网络流量并对其进行身份识别。此外，原型系统还设计了良好的可视化界面，对数据处理过程中的每一步都以图形的形式展示结果，方便用户实时掌握数据情况。原型系统的实现使移动设备的身份识别从学术研究的角度向工程实用的角度逐步过渡。

1.4 论文的组织结构

本文共有六个章节，每个章节的内容如下：

第一章首先介绍了本课题的研究背景及意义，明确了课题的研究方向。接着阐述无线设备指纹识别技术的研究现状，最后介绍本文的主要研究工作和论文组织结构。

第二章介绍基于流量认知的无线设备识别原理和整体框架。通过分析网络数据帧的生成和传输过程，讨论无线网络流量与设备的内部硬件组成、控制算法等因素的相关性。然后介绍基于流量认知的无线设备指纹识别框架，对框架中每一个模块的作用和使用的关键技术进行阐述。

第三章对面向设备指纹的流量进行分析。首先结合 Wireshark 抓取的流量对数据帧的结构进行介绍。紧接着介绍本文用于研究的数据集，分别从无线网络环境、数据采集方案和数据集几个方面进行阐述。接下来根据采集的数据集介绍预处理的关键技术，以将流量数据转换成易于提取特征的形式。最后提出两种特征指纹生成方法：基于概率密度的特征指纹和基于多特征融合的特征指纹。

第四章阐述无线设备指纹识别的步骤。首先描述了无线设备指纹识别的应用场景。然后介绍本文用于设备识别的四种分类器：随机森林（RF）、支持向量机（SVM）、K 最近邻（KNN）和朴素贝叶斯（NBC），对这四种分类器的原理和优缺点分别进行阐述。接着设计设备识别的评估方法，介绍了训练和测试过程以及评估指标，本文使用准确率（precision）、召回率（recall）和 F1 值来评价识别的效果。最后是对实验结果的展示和分析，先在现有数据集上进行设备识别实验，然后讨论了特征空间变化对识别结果的影响。

第五章阐述无线设备指纹识别原型系统的开发与实现。在对原型系统需求分析的基础上设计出系统架构，然后介绍系统实现的关键步骤，最后对原型系统的功能进行测试。

第六章对本文的研究工作进行总结，并分析识别技术和方案的不足之处，最后是对未来工作的展望。

2 基于流量认知的无线设备指纹识别原理与框架

2.1 无线数据帧与设备个体的相关性

图 2-1 展示了网络流量帧的生成过程，从图上可以看出，网络数据帧的产生需要操作系统和设备内部多个硬件组件共同协作完成。其中涉及到的硬件组件有多级存储器（L1/L2 Cache、主存、硬盘）、CPU、PCI 总线、DMA 控制器和网卡等。流量帧的产生从取值开始，先从存储器结构中取出相应的指令集发送给 CPU 执行；接着在操作系统（OS）的指示下，CPU 产生一个或多个包含起始存储地址和流量占用存储长度的缓冲区描述符；如果流量包在存储器中的存储是不连续的，那么 CPU 就会产生多个缓冲区描述符，操作系统也会相应地指示 CPU 生成新的缓冲区描述符，该描述符用于存储寄存器的映射信息；这些信息的发送需要经过前端总线和 PCI 总线；接着网卡会启动一个或多个 DMA 传输来检索描述符；网卡也会初始化一个或多个 DMA 传输把确切的流量数据帧从主存转移到网卡的传输缓冲区，这些数据由前端总线离开，经由北桥和 PCI 总线传输到网卡中。最后，网卡通知操作系统和 CPU 描述符已经处理完成，相应的流量包也产生成功发送到网络中。

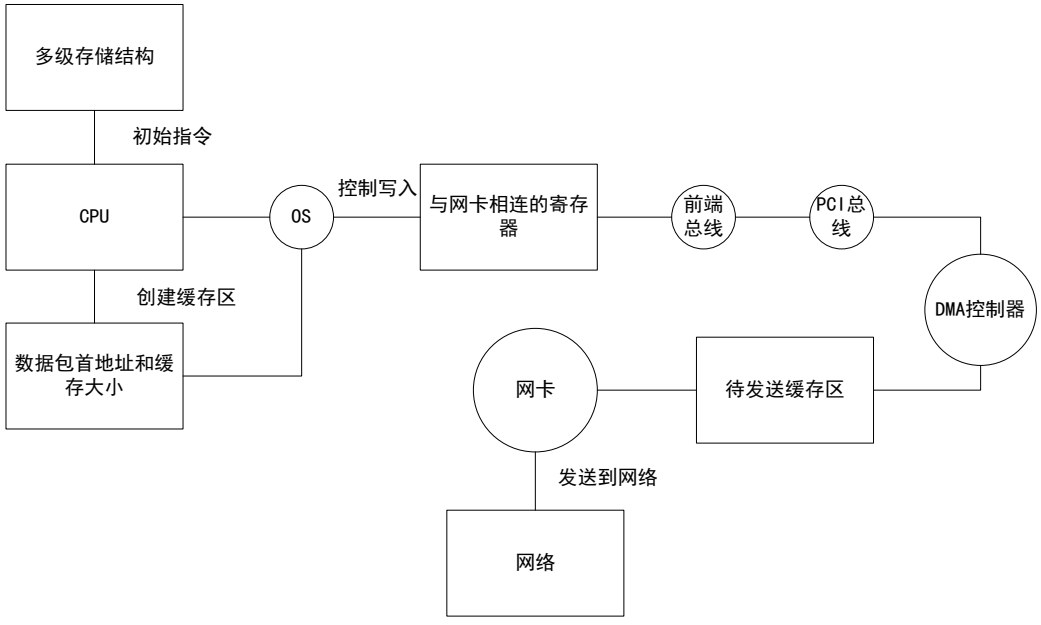

图 2-1 网络数据帧的生成过程

从网络流量包的产生过程可以看出，网络数据帧的生成需要多个硬件组件和操作系统的协同工作，所以网络流量包中有可能会包含能体现出发送设备身份的相关信息，比如设备所用的操作系统、设备 CPU 的配置、处理器所使用的主要算法、设备每个硬件的使用时限等。这种相关性不仅会体现在不同厂商、不同类型的设备中，也会体现在相同型号的设备中，因为即使是同一型号的设备，设备内部所采用的硬件部件、硬件部件的使用时间、处理器的时钟频率这些信息也不可能完全相同，犹如两个双胞胎也不

会有完全相同的指纹特征和基因图谱。正是网络数据帧在设备层次和组件层次上的差异为本文的研究提供了理论依据和一种全新的思路。

2.2 基于流量认知的无线设备指纹识别框架

基于流量认知的无线设备指纹识别框架如图 2-2 所示，主要包含流量数据采集、数据预处理、特征指纹生成和无线设备指纹识别几个部分。

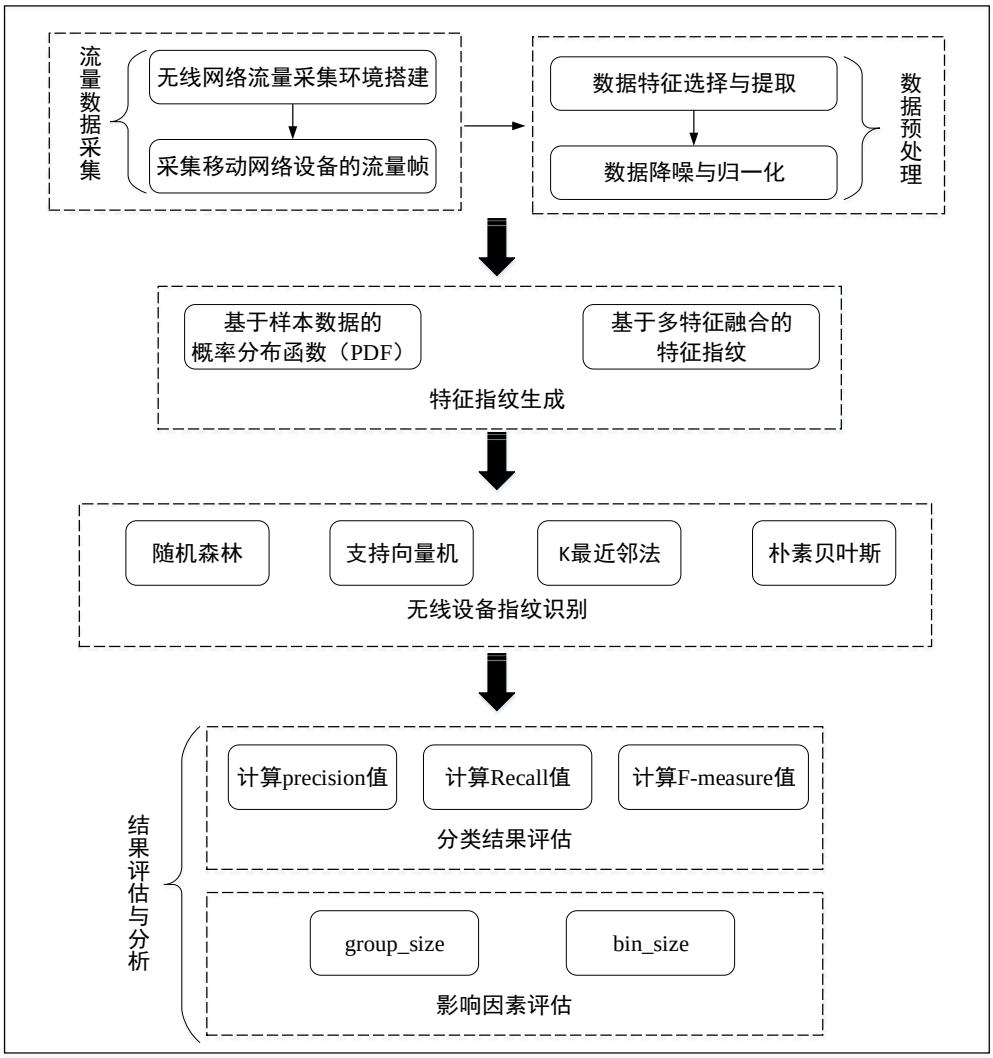

图 2-2　无线网络设备识别方案

在流量数据采集阶段，我们搭建了一个小型的无线局域网，用于捕获接入到网络的设备在访问互联网时发送的数据帧，建立本次实验的数据集。流量数据捕获的过程对用户来说是透明的，即不会对设备的正常使用造成干扰，也不会被用户发现。

数据预处理的目的是对采集到的流量数据进行处理，将其转化成便于提取特征的格式。采集到的流量数据中包含各种协议的数据帧，本文只使用 TCP 协议的数据帧进行相关实验。我们从数据包中过滤出 TCP 协议数据帧，从中提取出帧时间间隔、帧大小和传输速率这三种参数的数据，并分别对每种参数的数据进行降噪和归一化处理。

对处理后的数据，在特征指纹生成阶段提取出相应的设备指纹。本文提出了两种特征指纹的生成方法：基于概率密度的特征指纹和基于多特征融合的特征指纹。其中基于概率密度的特征指纹生成方法是将每种参数的数据划分为若干个组，计算每组数据的概率分布，分别形成基于每种参数的特征指纹。考虑到从一个参数中提取的特征指纹也许并不能完全表征设备的身份属性，独立的特征可能仅能表征设备在一个方面的属性，因此我们考虑将三种独立的特征融合在一起生成基于多特征融合的特征指纹。

在设备的训练和测试阶段用分类算法建立无线设备身份模型，并在现有的数据集上评估模型的有效性。做法是将设备指纹输入到分类器中，用十折交叉验证的方法进行训练和测试，计算出精确度（precision）、召回率（recall）和 F1 三个评估指标。本文用到的分类算法有：随机森林、支持向量机、K 最近邻和朴素贝叶斯。

最后本文还对影响设备识别效果的因素进行了讨论，探究了特征指纹生成过程中相关参数的改变对识别效果的影响，进一步论证了基于网络流量认知的无线设备指纹识别技术的有效性。

2.3 本章小结

本章介绍了网络数据帧的生成和传输过程，并从该角度分析了数据帧与无线设备个体之间的相关性，从原理上说明了基于网络流量的设备识别技术的可行性。紧接着介绍基于流量认知的无线设备指纹识别技术的整体框架。

3　面向设备指纹的流量分析与认知

本章首先介绍了本文用于研究的数据集，分别从无线网络环境、数据采集方案和数据集几个方面进行阐述。接下来介绍预处理的关键技术，以将流量数据转换成易于提取特征的形式。最后提出两种特征指纹生成方法：基于概率密度的特征指纹和基于多特征融合的特征指纹。

3.1 流量数据采集

3.1.1 无线网络环境

无线网络是指利用无线通信技术搭建的网络，通常分为借助公众移动通信网实现的无线网络（如 4G、3G 或 GPRS）和无线局域网（Wireless Local Area Networks，WLAN）两种方式。由于借助公众移动通信网实现的无线网络环境下的数据采集需要运营商的配合，数据获取困难，因此我们在实验中采用的是第二种方式，即自己设置 WiFi 热点，搭建无线局域网。为了探讨移动设备在远程网络下的流量行为，同时研究流量数据帧在网络中的传输是否会对识别造成影响，我们在实验中搭建了一个具有两个节点路由器的无线网络，其设计模型如图 3-3 所示。

该网络包含一个镜像交换机、两个路由器和若干个无线终端设备。一般来说，交换机或者路由器只能接收与自身相连的设备流量数据，若要捕获其他无线设备的数据，就需要用到镜像交换机。镜像交换机就是具有端口镜像功能的交换机或路由器，它将源端口的数据拷贝一份，经由目的端口发出，再用电脑进行捕获。这样既可以分析网络流量，同时不影响原来的数据发送。两个路由器一个负责数据的接收与转发，另一个则用于搭建 WiFi 热点。无线设备接入到局域网后，产生的流量经由路由器转发传输到镜像交换机，在装有 Wireshark 的笔记本电脑上即可抓取流量包。

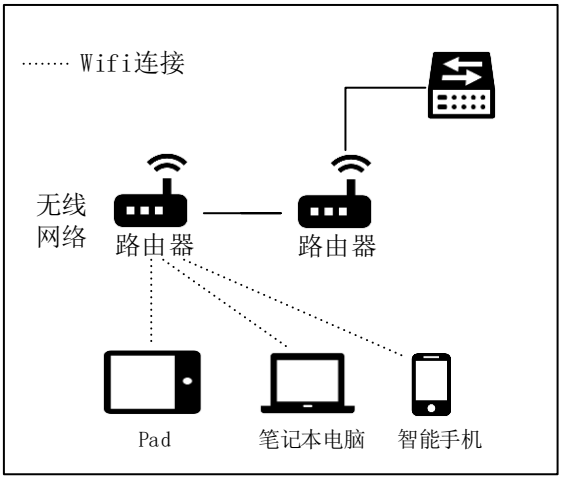

图 3-1　无线网络环境模型

根据上述网络模型，我们在西安交通大学彭康楼内搭建如图 3-4 所示的网络，在该

网络环境下完成数据采集。

搭建该无线网络用到的硬件设备主要有一台笔记本电脑，两个路由器和一个镜像交换机，这些设备的信息如表 3-1 所示。图中左下角是端口镜像交换机；白色 TP-Link 路由器为中间节点路由器，仅负责数据的接收与转发；右上角的华为路由器是终端路由器，由它建立 WiFi。图上还有一部黑色的手机，它接入局域网后产生的流量经过两个路由器的转发传输到镜像交换机，在笔记本电脑可抓取设备流量。

表 3-1　数据采集设备硬件信息

名称	主要参数	描述
Dell 笔记本	型号：Ispiron 灵越 15 5000 CPU 主频：2.5GHz 内存：4GB	装有 Windows10 系统，WireShark 软件，用于网络监控，捕获数据流量
TP-LINk 路由器	型号：TL-WR842N 无线传输速率：300Mbps 无线频段：2.4GHz	支持 IEEE 802.11b/g/n，具有良好的无线性能和连接稳定性，用于无线帧的发送与接收
HUAWEI 路由器	型号：WS550 最高传输速率：450Mbps 无线频段：2.4GHz	支持 IEEE 802.11n，兼容 IEEE 802.11b/g 用于建立 Wifi 热点
TP-LINK 端口镜像交换机	型号：TL-SF2005 5 个 10/100Mbps RJ45 端口	支持端口镜像功能，用于监控网络流量

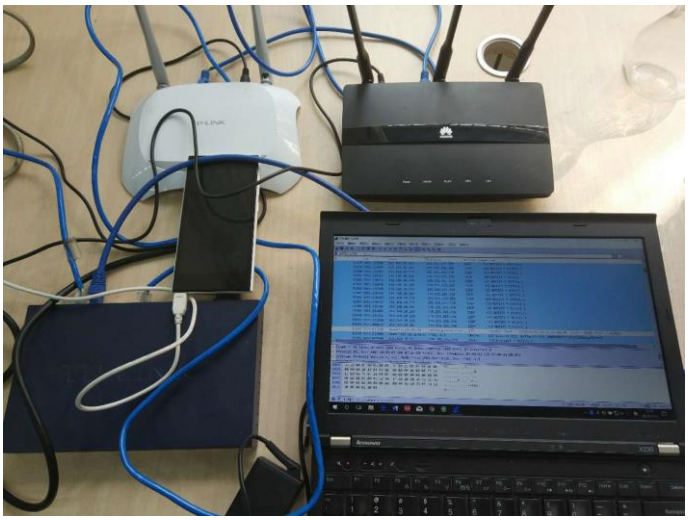

图 3-2　数据采集环境

本文使用 Wireshark 捕获接入到无线网的设备网络流量，Wireshark 是一款常见的网络数据包分析工具，具有以下几个特性：

1) 支持 Unix 和 Windows 平台；
2) 可以在线截取各种网络封包，显示网络封包的详细信息，也可以分析已有的报文数据，包括 http、TCP、UDP 等网络协议包；

3)　提供多种过滤规则，进行报文过滤；

4)　可进行多种统计分析；

需要注意的是 Wireshark 只能抓取和查看封包，而不能修改封包，也无法发送封包。图 3-5 展示了 Wireshark 的主界面。

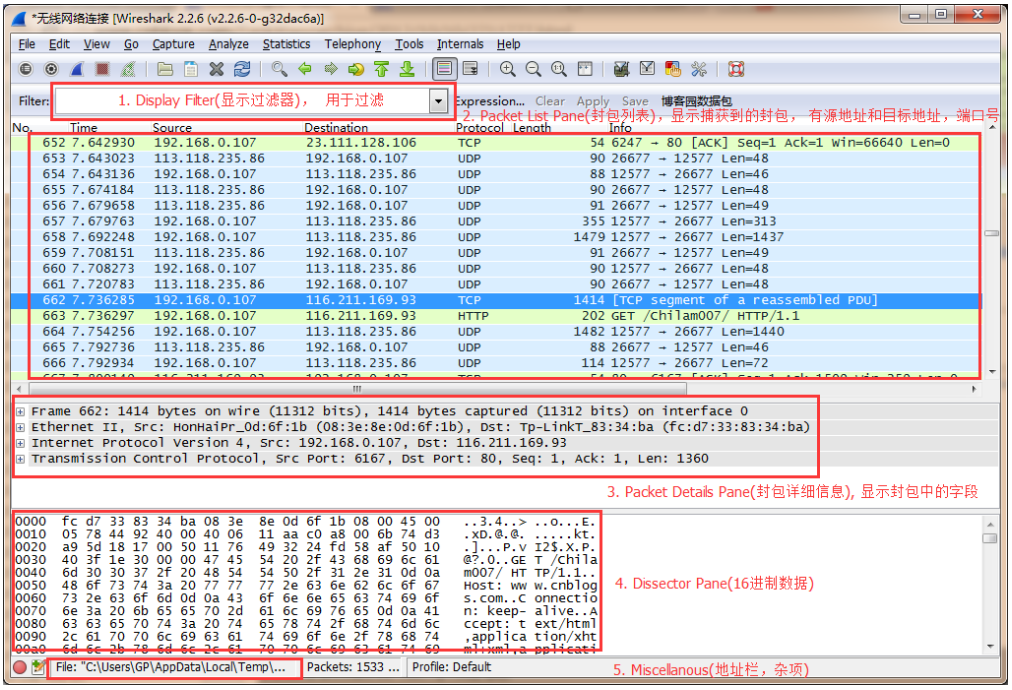

图 3-3　Wireshark 主界面

在 Wireshark 上设置“ip.src == 192.168.20.109”的过滤条件，通过“导出特定分组”便可以将无线设备的网络流量保存为 pcap 格式的文件。其中 192.168.20.109 为中间路由器 WAN 口的 IP 地址，由于终端路由器的局域网中一次仅有一台测试设备接入，而路由器在稳定的网络环境下不会主动发送 TCP 流量，因此可以认为这个无线网络中的 TCP 流量都是由局域网中的移动设备发出的。

3.1.2 数据采集方案

由于实验中是两跳网络，用了两个路由器，终端路由器开启的 WiFi 热点会形成一个小型局域网，移动设备连接这个局域网之后，路由器会分配一个临时的 IP 地址给这个设备（例如局域网网关为 192.168.1.1/24，设备被分配的 IP 地址将会是 192.168.1.X），但是设备被分配的这个 IP 地址仅在这个局域网内有效，流量数据帧经由路由器中转之后，数据帧中的 IP 地址便会转变为路由器 WAN 口的 IP 地址。如果路由器 WAN 口的 IP 地址为 202.117.14.191，那么设备发出的数据帧经过终端路由器转发后数据帧的源 IP 就会由局域网内的 192.168.1.X 变为 202.117.14.191。在这种情况下，局域网内所有移动设备的 IP 地址都会表现为 202.117.14.191。同样的道理，设备产生的数据帧也经由路由器转发后数据帧中的源 MAC 地址也更改为上一跳网口的 MAC 地址，即转发路由器的 MAC 地址，同一区域网的不同设备也无法通过 MAC 地址进行标识。

基于以上原因，此局域网中不同设备的网络数据帧不能通过 IP 地址或 MAC 地址

来区分，因此每台设备数据的采集都是单独进行的，即一台设备采集结束，数据保存后再进行下一台设备的采集。

TCP（Transmission Control Protocol，传输控制协议）是面向连接的、可靠的传输层通信协议，许多需要高度可靠的面向连接的服务都使用了 TCP，移动终端上的大部分应用程序都是基于 TCP 协议的，用户只需要使用手机的网络应用程序即可捕获到 TCP 数据流量。但是诸如微信、淘宝等应用程序产生的 TCP 的流量较少，采集足够的数据所花费的时间过长，因此我们让每个终端设备播放视频，可在短时间内产生大量的流量。每个无线终端大约持续播放视频 2 到 3 小时即可捕获到足够的数据。

基于 3.2.1 搭建的无线局域网，本文设计了如下数据采集方案：

第一步：无线设备接入 WiFi 热点，播放网络视频。

第二步：打开 Wireshark，开始捕获流量。

第三步：当流量足够后停止捕获，将流量数据保存为 pcap 格式的文件，文件名为设备的型号；当有相同型号的设备时，再后缀一个设备编号加以区分。

第四步：断开无线设备的网络连接，将其从局域网内移除，以免影响其它设备的采集。

对每台设备都严格执行上述采集方案，直到采集到足够的设备流量数据。

3.1.3 数据集介绍

本文一共采集了包括 PC、智能手机、平板电脑等 23 台移动设备的流量数据帧，它们的硬件及软件配置信息如表 3-2 所示。数据采集时间为 2017 年 1 月 25 日至 2017 年 2 月 16 日，在这 23 天中 23 台设备的采集工作独立进行，即一台设备采集结束再开始采集下一台设备。每台设备的数据采集规模都达到了 600MB 以上，数据帧数目至少 50 万条，持续时间在两个小时到四个小时之间，具体时间视数据采集时的网络状况而定。

表 3-2　采集设备及其配置信息

设备编号	设备型号	操作系统	CPU	RAM
1	小米 3	Android 4.4	Quad-core 2.3GHz	2GB
2	小米 4	Android 6.0	Quad-core Max 2.5GHz	2GB
3	小米 5S	Android 6.0	Quad-core Max 2.15GHz	4GB
4	iphone5	ios 6.0	苹果 A6 1.0GHz	1GB
5	华为 honor7	Android 5.0	Hisilicon Kirin 935	3GB
6	魅蓝 note	Android 4.4	Flyme0S4.2.0.4A	2GB
7	DELL 7420	Windows7	Intel Corei5-3230M 2.6GHz	8GB
8	DELL Vostro 3550	Windows7	Intel Corei5-2410M 2.3GHz	4GB
9	Thinkpad X240	Windows10	Intel Corei5-4200U 2.29GHz	4GB
10	iPad Air	iOS 8.4.1	苹果 A7+M7 处理器	1GB
11	华为 honor7	Android 5.0	Hisilicon Kirin 935	3GB

表 3-2（续）　采集设备及其配置信息

设备编号	设备型号	操作系统	CPU	RAM
12	ThinkPadT450	Windows7	Intel 酷睿 i5 5200U	4GB
13	ThinkPadT450	Windows7	Intel 酷睿 i5 5200U	4GB
14	iPad mini 4	iOS9	苹果 A8+M8 处理器	2GB
15	iPhone8	iOS11	苹果 A11+M11 处理器	3GB
16	华为 nova2	Android7.0	Hisilicon Kirin 659	4GB
17	华为 honor7	Android 5.0	Hisilicon Kirin 935	3GB
18	iPad Pro	iOS9	苹果 A9X+M9 处理器	4GB
19	iphone7	iOS10	苹果 A10+M10 协处理器	2GB
20	小米 MIX2	Android 7.1	高通 MSM8998	6GB
21	vivo X6	Android 5.1	Mali-T760	4GB
22	华为 honor7	Android 5.0	Hisilicon Kirin 935	3GB
23	OPPO A77	Android 7.1.1	Snapdragon 625 2.0Ghz	3GB

表 3-2 中每台设备捕获的数据帧数目如图 3-6 所示。图中横轴代表设备编号，与表中设备编号对应，纵轴为数据帧的数目。从图上可以看出数据帧的数量在 60 万到 200 万之间，大部分都在 100 万条左右。由于设备上网行为会产生多种协议的流量帧，如 TCP、UDP、HTTP 协议等，而我们在实验中只用到了其中 TCP 协议的流量帧数据，因此在后续处理中还需要过滤出 TCP 协议的数据。100 万条数据帧中大约可过滤出 30 万条左右的 TCP 协议数据帧。

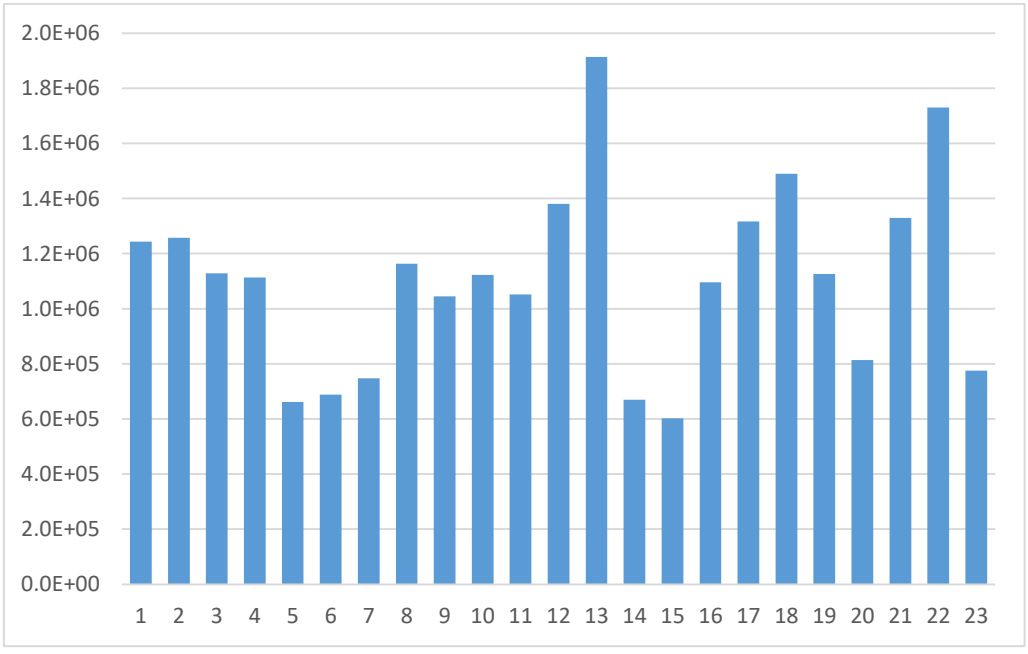

图 3-4　数据帧数目统计图

3.2 数据预处理

数据预处理模块的功能是对捕获的流量帧数据进行解析，从中提取出构建特征指纹所需参数，并对这些参数进行降噪和归一化处理，将其转化为便于提取特征的形式。

数据预处理的流程如图 3-7 所示。先打开 pcap 文件，对每一条数据帧提取帧号、帧间隔时间、帧大小和传输速率这四项参数；对除帧号外的参数进行降噪处理；再将降噪后的数据归一化，处理完成后保存结果。其中，解析 pcap 文件并提取参数是基于 Java 平台的，数据降噪和归一化处理用的是 Python 语言。下面逐一介绍每一个过程使用的关键技术和具体步骤。

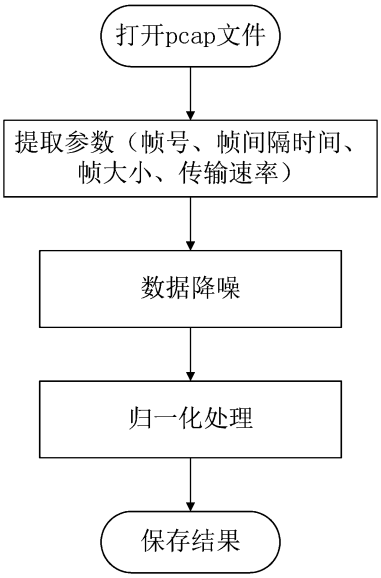

图 3-5　数据预处理流程图

3.2.1 数据帧参数提取

本文使用的数据集为 23 台无线设备的网络流量数据，这些数据是 pcap 格式的文件。pcap 格式的文件无法直接使用，需要从中提取出固定的参数才能用于构建特征指纹。虽然 Wireshark 也具有流量数据包解析的功能，但只能手动打开每一条流量数据查看详细信息再记录参数，我们实验中对每一台无线设备都抓取了上百万条流量，使用 Wireshark 解析数据并提取特征是一项不可能完成的任务。本文使用 Java 的 jNetPcap 库进行 pcap 文件的处理，jNetPcap 是一个开源的 Java 类库，主要功能是捕获和分析数据包。由于 Java 平台本身不支持底层网络操作，因此需要用第三方包封装不同系统的 C 库以提供 Java 的上层接口。Libpcap 是一个开源且功能齐全的处理网络流量的函数库，提供了流量包的捕获与解析功能，许多网络监控软件都是基于 Libpcap 进行开发的。jNetPcap 是基于 Java 平台的 Libpcap 库的封装，提供与 Libpcap 完全相同的功能。jNetPcap 主要有下面几个特点：

1）提供几乎所有 libpcap 类的封装；

2）可实时解码所捕获的数据包；

3）提供广泛的网络协议库（核心协议）；

4）用户可以使用 Java SDK 轻松添加自己的协议定义；

5）jNetPcap 可以使用本机和 Java 混合实现最佳的分组解码性能。

由于我们的实验是在 Windows 系统下进行的，使用 jNetPcap 还必须要安装 WinPcap，以提供 jNetPcap 所需要的链接库。pcap 类是 jNetPcap 中最为核心的类，是一个对 libpcap 中方法的 Java 直接映射，提供了获取网卡设备列表、设置过滤器、数据包分析等必须的功能。

本文利用 jNetPcap 开源库对捕获的网络流量进行解析，使用 jNetPcap 类库提供的接口打开 pcap 文件，对其中每一条数据帧记录，从帧头提取如下几个参数：

1）帧号（Frame No）：每一个数据帧在文件中的编号，这一参数主要是用于计算帧间隔时间。无线设备的网络流量中除 TCP 协议数据帧外，还有其他各种协议的数据帧，如 UDP 协议、FTP 协议等。因此数据包中的 TCP 协议数据帧之间可能还有其它协议的数据帧，故而提取的帧号有可能不连续。

2）帧间隔时间（Interval-arrival time，IAT）：两帧到达时间的间隔，由于我们用于实验的数据帧不一定是连续的，单纯用当前帧的到达时间减去上一帧的到达时间来计算帧间隔时间是不合理的。我们采用的做法是用两帧到达时间之差除以两帧的帧号之差，计算公式如下：

$$IAT = \frac{t_2 - t_1}{F_2 - F_1} \tag{3-1}$$

其中：t_1 和 t_2 分别为两帧的到达时间，F_1 和 F_2 分别为两帧的帧号。

3）帧大小（FrameSize）：每一帧的大小。

4）传输速率（TransRate）：网卡进行数据传输的速率，单位为 Mbps（兆位/秒），反映了网卡每秒钟接受或者发送数据的能力，计算公式为：

$$TransRate = FrameSize/IAT \tag{3-2}$$

经过上述操作，从每个设备中提取的数据帧参数格式如下表：

表 3-3　数据帧参数格式

帧号	帧间隔时间	帧大小	传输速率
1	0.038666	130	3362.126933
2	0.018444	54	2927.781392
5	0.057298	60	1047.156969
…	…	…	…
N	0.061937	60	968.726286

其中帧号仅用于计算帧间隔时间，后文中其它工作均不需要这一参数。

3.2.2 数据降噪

由于网络环境的复杂性，网络数据的传输会受到网络带宽和网络延时等因素的影响，例如帧间隔时间就会受到数据包延迟的影响而变得异常大。这种不和其他数据相

一致的数据我们称之为噪声数据。噪声数据会干扰设备识别的结果，需要对其进行处理。本文使用自定义区间降噪的方法进行数据降噪，自定义区间降噪的做法是：用户根据需要自定义区间，只查看特定范围内的数据。这种方法的关键在于阈值的选取，只要选出合适的阈值，即可过滤出在阈值范围内的正常数据。

我们通过每个参数的概率密度（probability density function，PDF）曲线来查看噪声数据的分布情况。PDF 曲线反映了随机变量在某个确定取值点附近可能性的大小，而随机变量的取值落在某个区间内的概率则为概率密度曲线在该区间上的面积。如果随机变量在某一点附近的概率密度值过小，则说明该范围内的数据量非常少，即为我们所提到的异常数据，也就是噪声数据。

我们用 MATLAB 的 ksdensity 函数来估计样本向量的概率密度。该函数首先统计样本向量 X 在各个区间的概率，再自动选择 X_1，计算对应的 X_i 点的概率密度。ksdensity 绘制的是连续随机变量的概率密度，而连续型随机变量的概率密度函数是可以大于 1 的。由于概率密度函数的积分为 1，因此对于横坐标的值小于 1 的点，其纵坐标很可能大于 1。

以帧时间间隔为例，随机选取一台无线设备的数据，绘制其帧时间间隔的 PDF 曲线如图 3-8 所示。图中横轴代表 IAT 取值，纵坐标代表在某一点上取值的概率密度。从图中可以看到该设备帧时间间隔的取值范围为[0,1.4]，同时观察到曲线呈非常明显的脉冲状，在[0,0.01]范围内的纵坐标值特别大，说明 IAT 取值落在该区间内的数据非常多；曲线在[0.2,1.4]区间内的概率密度值无限接近于零，说明该范围内的数据很少，可以将其作为噪声数据滤除。

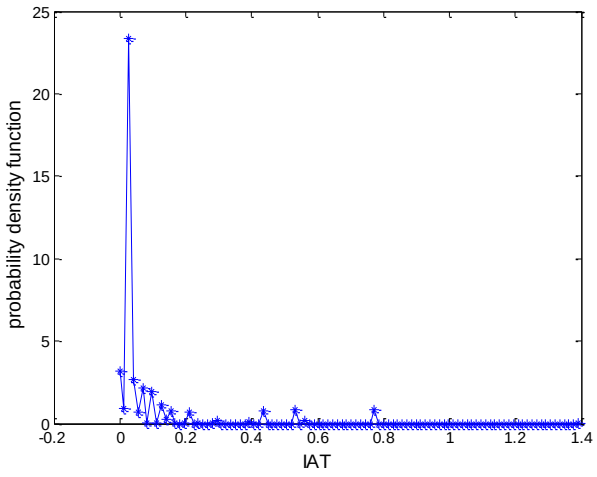

图 3-6　帧间隔时间的 PDF 曲线

为了选取合适的阈值，我们对比了不同阈值的过滤效果，绘制不同阈值下的散点图，如图 3-9 所示。图中左上角的子图为原始数据的散点图，右上角是以 0.6 为阈值过滤数据的散点图，左下角和右下角分别为以 0.4 和 0.2 为阈值过滤噪声的散点图。从图

上可以看到，当阈值为 0.2 时，数据分布已经较为集中，说明绝大多数的噪声数据已经被成功滤除。

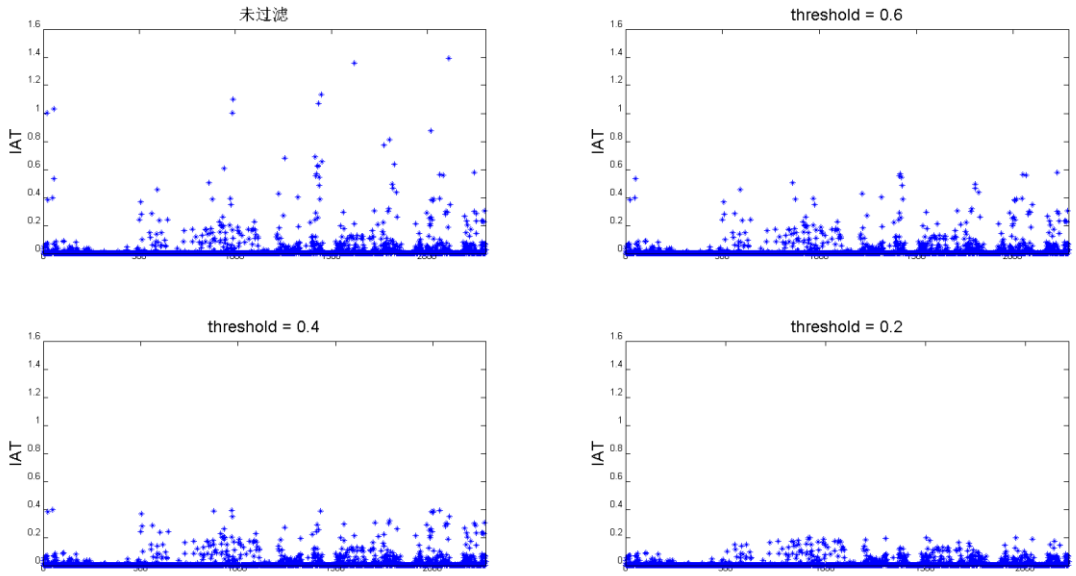

图 3-7　不同阈值的降噪效果对比（IAT）

我们用同样的方法对帧大小和传输速率的噪声分布情况进行分析，最终确定出合适的阈值。帧大小和传输速率的阈值分别为 100 和1.2×10^{6}，降噪后的数据散点图与原始数据散点图的对比分别如图 3-10 和图 3-11 所示。从图中可以看到帧大小主要集中在 0 到 100 之间，在 100 到 1500 之间零散地分布着一些数据。而传输速率的噪声数据较少，放大坐标系后可以看到数据集中分布在 0 到1.2×10^{6}之间。

需要指出的是，虽然对于三种参数的阈值分析是分开进行的，但是过滤操作却是一起进行的。即如果一组数据（包含 IAT、FrameSize、TransRate）中有一个参数大于相应阈值，那么这一组数据均会被删除。如表 3-3 中第一组数据，三个参数值分别为 0.038666、130、3362.126933，虽然帧间隔时间和传输速率均在正常范围内，但是帧大小却超出阈值范围，因此该组数据将被删除。

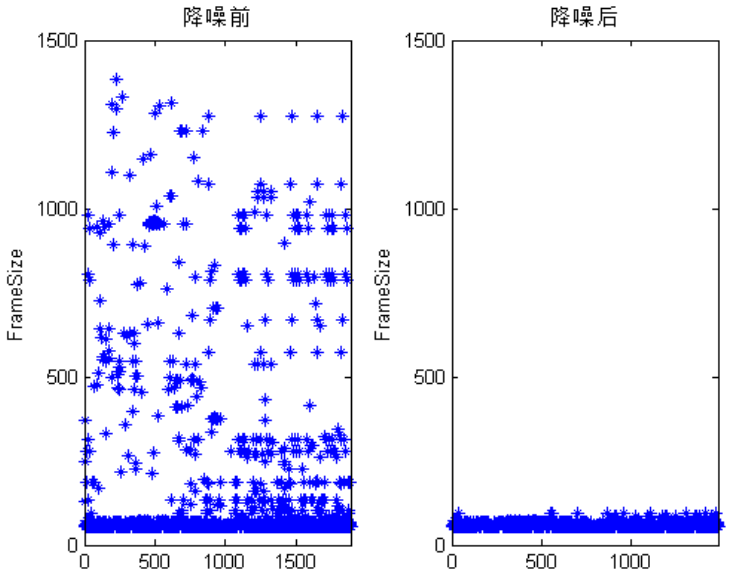

图 3-8　帧大小降噪效果对比

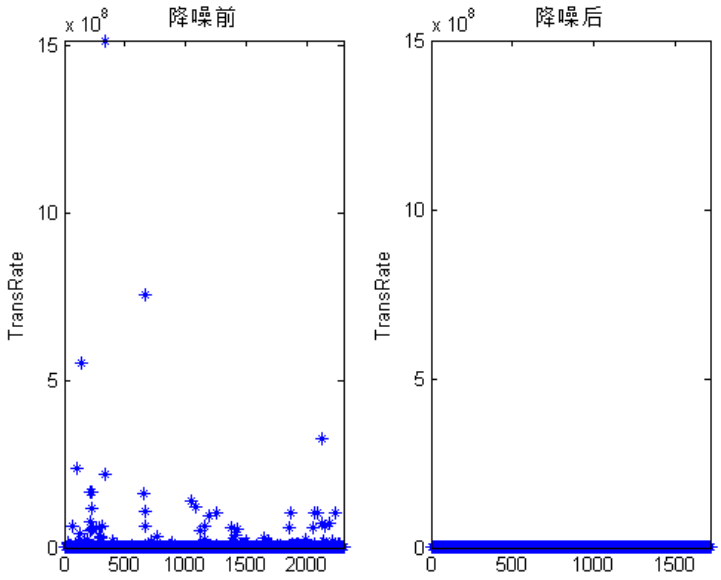

图 3-9　传输速率降噪效果对比

3.2.3 数据归一化

根据 3.3.2 的分析可以看出，同属性的数据往往具有不同的量纲或量级，降噪后的 IAT 的取值在 0 到 0.2 之间，而传输速率的取值则在 0 到1.2×10^{6}之间，数据的量纲差异非常大。为了避免量级的差异对实验结果造成影响，本文对降噪后的数据进行了归一化处理。归一化对后续的处理十分必要，很有可能会提高实验精度。

数据归一化的目的是将数据落按照一定的比例进行缩放，使其落在特定的范围内。本文使用 min-max 归一化方法，计算公式为：

$$y = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3-3}$$

其中：$x_{\min}$ 为某个属性的最小值，$x_{\max}$ 为某个属性数据的最大值，x 表示待归一化处理的数据，y 表示归一化处理之后的数据。

从公式中可以看出，min-max 标准化的结果是将数据向量落在[0,1]区间内。

3.3 特征指纹生成

本文提出了两种特征指纹生成方法：基于概率密度密度的特征指纹、基于多特征融合的特征指纹，两种指纹可独立使用。

3.3.1 基于概率密度的特征指纹

为了初步度量提取的各项参数能否用于区分不同的设备，我们绘制出同一参数在不同设备上的 PDF 曲线。如果不同设备某个参数的 PDF 曲线有较为明显的差异，我们可以认为不同设备在该参数上的分布不同，也意味着该参数可以作为设备识别的依据。

以传输速率（TransRate）为例，我们随机挑选了三台无线设备，绘制其传输速率的 PDF 曲线，如图 3-12 所示。从图上可以看出，三条曲线在 IAT 取值较小时完全不重合，因此我们有理由相信，帧时间间隔对于移动设备具有很好的区分性。同样的，帧时间间隔和包大小在无线设备识别中也表现出了良好的性能。

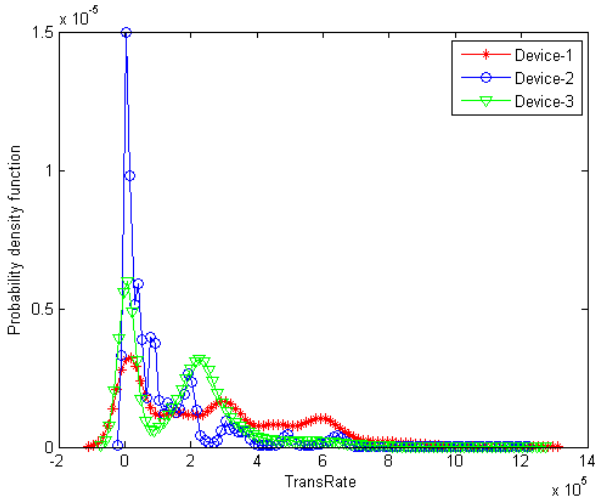

图 3-10　三台设备传输速率的 PDF 曲线

通过上述分析可以看出，每个参数的概率密度曲线可以很好地区分不同的设备。基于这个思路，我们提出基于概率密度的特征指纹构建方法。由于每个参数服从的概率密度函数未知，在样本量较大的前提下，可以用频率近似替代概率。特征指纹生成过程如图 3-12 所示。

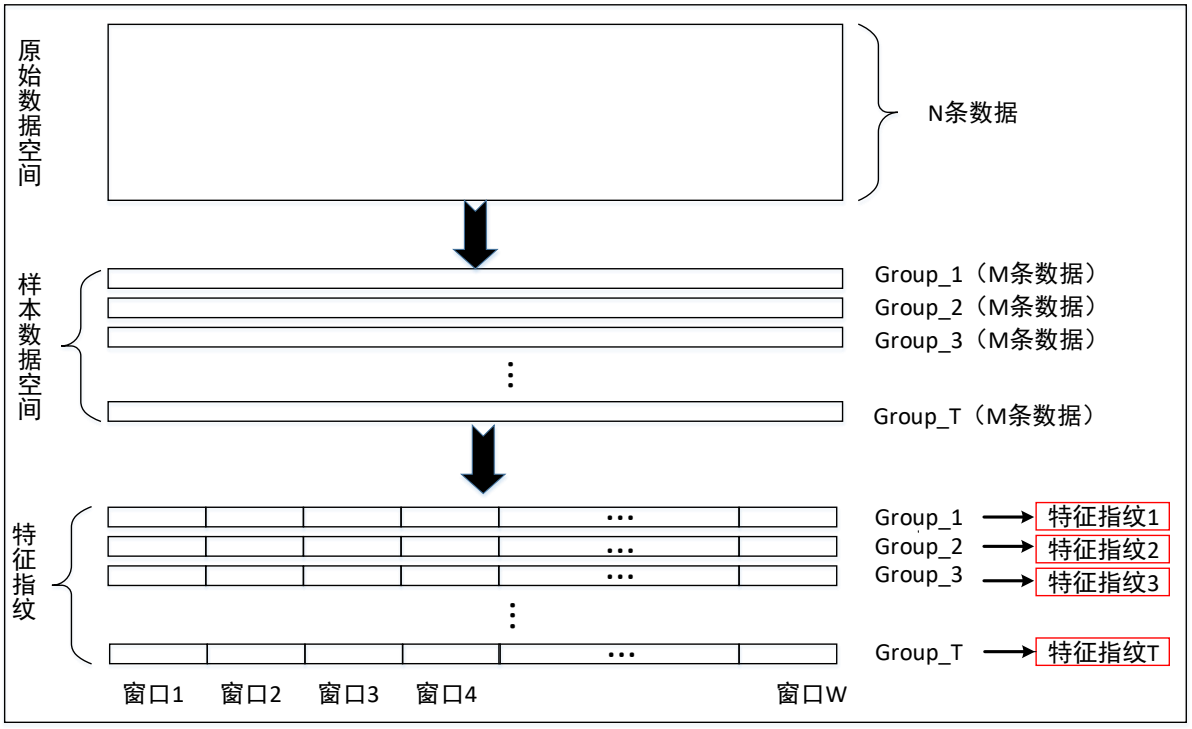

图 3-11 特征指纹生成过程

为了详细解释特征指纹的生成过程，不妨假设每个参数有 N 条数据，为 $x_1, x_2, \ldots, x_N$，基于概率密度的特征指纹构建操作步骤如下：

第一步：对数据进行分组。将 N 条数据划分为若干组（group），假设每组有 M 个样本，即每组的样本量（$group\_size$）为 M。则原始数据被划分为 T 组，其中 $T = N/M$。这 T 组数据分别用 $G_1, G_2, \ldots, G_T$ 表示，G_i（1<i< T ）的形式如下：

$$G_i = [X_{M*(i-1)+1}, X_{M*(i-1)+2}, \ldots, X_{M*i}] \tag{3-4}$$

第二步：统计 G_i 的频率分布情况。记 G_i 的最小值为 $g_{\min}$，最大值为 $g_{\max}$，将区间 $[g_{\min}, g_{\max}]$ 等分为 W 个窗口：$[g_{\min}, g_1], (g_1, g_2], \ldots (g_{W-1}, g_{\max}]$。统计 G_i 中的数据落在每个窗口内的频率，形成 $1 * W$ 的特征向量：

$$f = [freq_1, freq_2, \ldots, freq_W] \tag{3-5}$$

其中 $freq_j (1 < j < W)$ 为样本数据落在第 i 个窗口内的频率。

第三步：生成特征指纹。将 T 组数据的特征向量结合在一起，构成特征矩阵。该特征矩阵的维度为 $T * W$，其中 T 为分组个数，W 为窗口数目。这个特征矩阵就是设备的特征指纹。

实验中，我们设定每组的数据量 M =300（即 $group\_size$=300），窗口大小 W =20（即 $bin\_size = 20$）。我们从每个设备的流量数据中提取出三个参数：帧间隔时间（IAT）、帧大小（FrameSize）、传输速率（TransRate），按照上述操作步骤造作，将会形成三个特征矩阵，每个特征矩阵都可以作为设备的特征指纹来验证设备的身份。

3.3.2 基于多特征融合的特征指纹

考虑到从一个参数中提取的特征指纹也许并不能完全表征设备的身份属性，独立的特征可能仅能表征设备在一个方面的属性，因此我们考虑将三种独立的特征融合在一起生成特征指纹。在式 3-5 的基础上，将从三个参数中提取出的特征向量合成一个向量，如公式 3-6 所示：

$$f = [IAT_1, \ldots, IAT_W, FS_1, \ldots, FS_W, TR_1, \ldots, TR_W] \tag{3-6}$$

其中 $IAT_j(1 < j < W)$ 指帧间隔时间在样本内的概率分布，$FS_j(1 < j < W)$ 指帧大小在样本内的概率分布，$TR_j(1 < j < W)$ 指传输速率在样本内的概率分布。

最后将所有的特征向量（T 个特征向量）拼接为一个特征矩阵，特征矩阵的维度为 $T * (W * 3)$，这个特征矩阵即为该设备基于特征融合的特征指纹。

3.4 本章小结

本章首先介绍了 TCP/IP 参考模型和 Wireshark 抓取的 TCP 数据帧格式，为后文的研究提供基础。紧接着介绍搭建的无线网络环境和数据采集方案，采集到包括个人 PC、智能手机、iPad 等在内的 23 台设备的流量数据，建立了本次实验的数据集。然后从每个设备的流量中过滤出 TCP 协议的数据帧，从中提取出帧时间间隔、帧大小和传输速率三个参数的数据，再进行数据降噪和归一化的处理，将其转化为易于提取特征的格式。最后从处理后的数据中提取出每个设备的特征指纹，本文提出了两种特征指纹生成方法：基于概率密度的特征指纹和基于多特征融合的指纹，两种指纹均被用于标识无线设备的身份。

4 无线设备指纹识别

4.1 无线设备指纹识别概述

从模式分类的角度来看，无线设备的识别是一项有挑战性的任务，是一个多分类问题。在基于流量认知分析的无线设备识别场景下，用户从捕获无线设备的网络流量中提取出设备指纹，然后将该设备指纹与指纹库中的指纹进行比对，识别出这是指纹库中的哪一个设备，或者为未知的新设备。鉴于很多分类算法在原理推导上都是基于二分类的假设，它们在多分类的情况下精度较低，本文采取的策略是将多分类问题转化为二分类问题。具体的做法是从数据集中随机挑选两个设备，将其中一个指定为正类，另一个指定为负类，给出相应的标签后将特征指纹输入到分类器中进行训练和测试，计算出相应的评估指标来测试该特征指纹的有效性。

本文使用了四种设备识别分类器：随机森林、支持向量机、K 最近邻和朴素贝叶斯，首先分别在这四种分类器上进行设备识别实验，验证两种指纹的有效性，紧接着验证了特征空间变化对于识别效果的影响。

4.2 设备识别分类器

4.2.1 随机森林

随机森林（Random Forests，RF）是一种比较新的机器学习模型。上世纪八十年代 Breiman 等人提出分类与回归树的算法（Classification and regression tree，CART）[41]，通过反复二分数据进行分类或回归，将输入空间即特征空间划分为有限个单元，并在这些单元上确定预测的概率分布[42]。

Breiman 等人在 2001 年又提出将多个决策树分类树组合在一起，形成随机森林[43]。随机森林中包含的决策树有一个先决条件，即决策树之间互相没有关联（如果决策树之间有关联的话，决策树的总数将会减少）。形成森林之后，当有一个新的输入样本进入的时候，每个决策树测试待分类项中相应的特征属性（这些属性可以是分类项不同的属性），并按照判定条件将输入样本归类到决策树不同的分支，直到最后到达叶节点，叶节点即为决策树预测的类。图 4-1 是一棵简单的决策树示例，用于判定用户的身份。主要根据三个属性进行判断：年龄大小、是否为学生、信用评级，每一个节点都表示根据一个属性进行判断，叶子结点给出判定的结果，即是否为目标用户。每棵决策树根据属性给出一个判定结果，随机森林根据所有决策树的结果进行投票，得票多的即为最终结果。随机森林的算法步骤如下：1）从训练样本池中随机有放回地选择 n 个样本；2）从输入特征中随机选择 m 个特征进行训练，建立决策树；3）将步骤 1 和步骤 2 重复 k 次，建立 k 个决策树；4）每棵决策树对新样本进行分类，随机森林综合 k 个决策树的结果给出样本最终类别。

随机森林的“随机”体现在：1）在训练每棵树时，从训练样本池（N 个样本）中选取的训练子集（n 个样本，n<N）是随机有放回地选取的。决策树的训练阶段在一定程度上避免了过拟合现象，因为每次进行训练的数据都只是 N 个样本中的一部分。2）在决策树训练中的条件判断节点中，从输入特征（M 个特征）中随机选取 m 个输入特征（m<M）,然后从这 m 个特征里选一个最好的进行分裂。这种分类机制显著地提升了分类精度，且不会带来额外的计算开支。

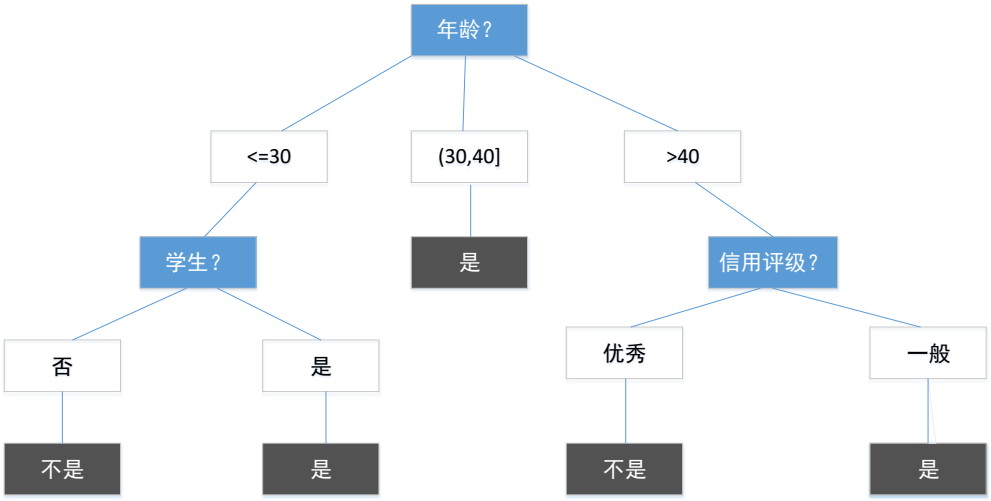

图 4-1　决策树示例

随机森林有许多的优点：1）随机森林在训练的过程中对特征子集的选取是随机进行的，因此它在处理多特征的样本时，无需额外进行特征选择；2）随机森林所包含的决策树具有独立性，相互之间并无关联，因此可以采用并行处理的方式进行训练，大大节省了训练所需的时间；3）随机森林使用了无偏估计泛化误差，因此泛化能力比其他模型更强；4）随机森林在处理不平衡数据集时表现效果良好；5）即便数据集中有大部分的特征遗失，仍可以保持较高的准确度。

4.2.2 支持向量机

支持向量机（Support Vector Machine，SVM）是 Cortes 和 Vapnik 在 1995 年提出的经典分类算法[44]，由于在解决小样本、非线性及高维模式识别中表现出许多特有的优势[45]，很快成为机器学习的主流技术，并直接掀起了“统计学习”（statistical learning）在 2000 年前后的高潮[46]。

SVM 是一种经典的分类器模型，其主要思想是将在低维空间中线性不可分的数据通过映射函数映射到高维空间中，使其在高维中可以准确地分类（如图 4-2）。SVM 在高维空间中寻找一个超平面，根据这个超平面对正负类数据进行划分，SVM 算法要求正负类到超平面的距离之和尽可能大。在测试阶段，新的样本同样会被映射到高维空间，通过超平面对新样本的数据进行类别划分。

选择一个恰当的核函数不仅可以在很大程度上降低 SVM 分类的时间复杂度，还可

以提高 SVM 的分类精度，如何根据实际的训练数据选择恰当的核函数是 SVM 在应用中亟待解决的一个关键问题。本文使用的核函数是线性核函数，即 $K(X, X_i) = X * X_i$，线性核函数计算简单，所需时间开销小。

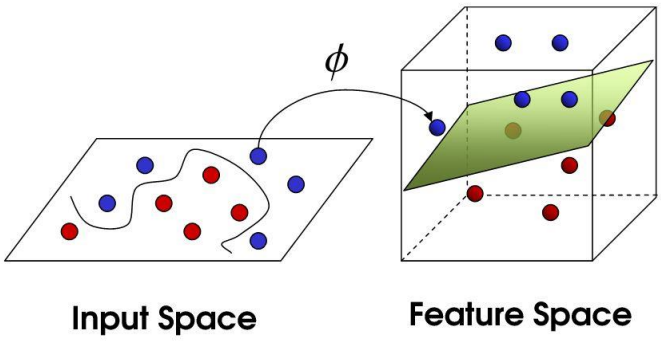

图 4-2　支持向量机原理

支持向量机是一种有着坚实理论基础的小样本学习方法。从原理上看，它并未涉及到概率测度和大数定律等传统的统计学方法，这在很大程度上解决和简化了回归和分类问题的模型。与此同时，SVM 模型的复杂度主要取决于支持向量机的数目，由于进行了空间映射，样本空间的维度对算法的计算复杂度并不会产生很大的影响，因此对于高维数据的分类问题，使用 SVM 算法不会产生很大的计算代价。此外，SVM 实现较为简单，且具有良好的鲁棒性。但是支持向量机算法在处理大规模训练样本和多分类问题时存在一定的困难。

4.2.3 K 最近邻

K 最近邻（K-Nearest Neighbor，KNN）分类算法是最简单的监督式算法之一[47]，它的分类思想是从训练数据集中找到与待测样本最靠近的 k 条数据，这 k 条数据中的大多数样本属于哪个类别，待测样本就被判定为哪个类别。图 4-3 展示了 K 最近邻的原理，图中与 X_u 最近的 5 个点中有 4 个是 ω_1 类的，1 个属于 ω_2，因此 X_u 将会被标记到 ω_1 类中。KNN 的计算步骤为：1）计算距离：给定待测样本，计算该样本与数据集中每个数据的距离；2）寻找邻居：对计算得到的距离进行排序，寻找与待测样本距离最近的 k 个样本作为该数据的最近邻；3）类别判定：确定这 k 个最近邻所属的主要类别，以此作为判定待测样本所属类别的依据。

KNN 算法与其他有监督分类算法最大的区别在于其训练过程时间开销为零，它不需要对训练进行显式的训练，当有待测样本时才进行处理。KNN 算法简单，易于理解和实现，且在处理多分类问题时有着比其它算法更好的表现。KNN 算法中关键的问题在于 k 值的选取和度量方式的选择。当 k 取不同值时，分类结果会有显著差异。如果 k 值较小，噪声数据将会影响分类结果；如果 k 值较大，k 个近邻中样本点的类别太多，会影响结果的判断。此外，距离度量方式的选取对分类结果也有很大影响，采用不同的

距离度量计算方式，找出的“近邻”可能会有显著差异，从而导致分类结果也不相同。本文的实验中，我们使用欧式距离作为距离度量方式，同时将 k 取值设置为 20。

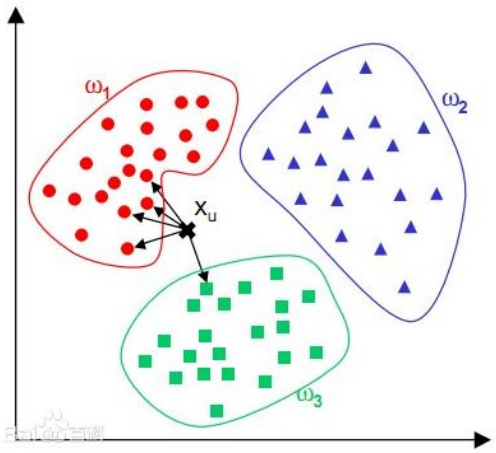

图 4-3　K 最近邻原理

4.2.4 朴素贝叶斯

贝叶斯分类是一类分类算法的总称，这类算法均以贝叶斯定理为基础，故统称为贝叶斯分类。而朴素贝叶斯分类(Naïve Bayes Classifier，NBC)是贝叶斯分类中最简单却应用最为广泛的一种方法[48]，其中朴素是指对于模型中各个特征有强独立性的假设。根据贝叶斯定理，给定样本特征 x ,该样本属于类别 y 的概率如式(4-1)所示：

$$p(y|x) = \frac{P(x|y)p(y)}{p(x)} \tag{4-1}$$

式中，x 是一个特征向量，假设其维度为 M 。由于朴素贝叶斯算法假设样本特征之间相互独立，式(4-1)可以用式(4-2)表示：

$$p(y = c_k|\mathrm{x}) = \frac{\prod_{i=1}^{M} p(x^i|c = c_k)p(y = c_k)}{\sum_k p(y = c_k)\prod_{i=1}^{M} p(x^i|c = c_k)} \tag{4-2}$$

通过公式 4-2，求样本特征 x 属于类别 y 的概率就转化为统计类别 y 的先验概率问题。在给定已知的训练样本集合以及类别标签的条件下，计算出未知的样本属于每一类别的条件概率，取概率最大的一类为最终分类结果。

朴素贝叶斯算法的思想是基于古典数学理论，数学基础坚实。算法的逻辑简单，便于理解，实现起来较为容易，分类效率也比较稳定。理论上，朴素贝叶斯算法的误差率比其它分类算法更小，但在实际情况中却并非如此。因为朴素贝叶斯模型假设特征之间相互条件独立，这个假设在实际应用中很多时候并不成立[49]。因此在特征数目较多或特征之间相关性较大时，分类效果不好；而在属性相关性较小时，朴素贝叶斯分类的性能最为良好。

4.3 评估方法

4.3.1 数据集

如章节 3.2 所述，对接入到无线网络内的设备流量数据进行捕获。本文一共采集到 23 个设备的数据，每个设备的流量帧数目都达到了 50 万条以上。对采集到的流量帧，根据 3.3 所述步骤进行参数提取、数据降噪和归一化处理，将其转化为易于提取特征的形式，然后再按照 3.3 中所述方法提取出特征指纹，输入到分类器中进行训练和测试。

4.3.2 训练和测试过程

在 4.1 所述的设备识别场景下，我们用十折交叉验证的方法进行训练和测试。交叉验证（Cross Validation），也被称作循环估计（Rotation Estimation），是一种统计学上将数据样本切割成较小子集的实用方法[50]。其基本思想是将样本划分为若干组，其中一部分用于训练，另一部分则用于模型的测试。在分类问题中，分类器先根据训练样本建立合法用户的身份模型，再用验证样本测试训练模型的有效性。图 4-4 展示了交叉验证的过程。

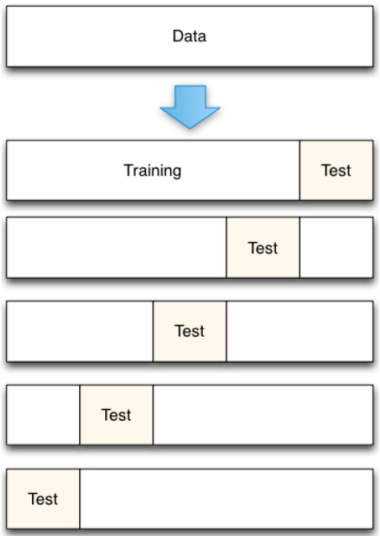

图 4-4 交叉验证过程示意图

十折交叉（10-fold cross validation）是最常用的交叉验证方法。做法是将原始数据随机分割成规模相同的 10 个子样本，将其中 9 个子样本作为训练数据（Train set），一个子样本作为测试数据（Test set），交叉验证重复 10 次，最后取 10 次测试结果的平均值。十折交叉法随机地产生训练和测试所用的子样本，使所有的样本得到充分的利用，同时可以避免由训练数据不足导致的欠拟合现象。

在 4.1 所述的设备识别场景下，按照如下的步骤训练并且测试分类器对无线设备的识别能力：

第一步：从 23 台设备中随机选择两台设备，将其中一个设备当做正例，样本标记为+1，另一台设备的样本标记为-1；

第二步：将正例数据和负例数据分别随机分割成规模相同的 10 个子样本，划分的

10 个子样本中，每个子样本内各个类别数据的比例和原始数据集中各个类别的比例相同；

第三步：将其中的 9 个子样本分别作为正例和负例的训练样本，建立相应的认证模型；

第四步：将剩下的 1 个子样本作为测试数据，测试分类器对于目标设备和其他设备的识别能力；

由于在训练过程中设备的选择和训练样本的生成均是随机进行的，为了避免随机因素对实验结果造成影响，本文将上述训练和测试过程重复了 50 次，每次分别计算出相应的 precision、recall 和 F1 值，最后对 50 次实验结果取平均值。

4.3.3 评估指标

precision 和 recall 是模式识别问题中常用的两个评估指标。precision（准确率，又称查准率）和 recall（召回率，又称查全率）的概念源于信息检索系统，是用来衡量某一类文献检索系统的信号噪声比的指标，后来也用于评估模式识别中分类算法的性能。准确率通常用于评价结果的质量，而召回率则用于评价结果的完整性，其计算方法如下：

$$precision = \frac{TP}{TP + FP} \tag{4-3}$$

$$recall = \frac{TP}{TP + FN} \tag{4-4}$$

其中：

FN — False Negative，被判定为负样本，但事实上是正样本；

FP — False Positive，被判定为正样本，但事实上是负样本；

TN — True Negative，被判定为负样本，事实上也是负样本；

TP — True Positive，被判定为正样本，事实上也是正样本。

由上述公式可以看出：准确率计算的是所有“正确被判定为正的样本”占所有“标记为正的样本”的比重；召回率反映了“正确被判定为正的样本”占所有“实际为正的样本”的比重。一般来说，不能仅依靠准确率或召回率来评价分类器的效果，两个指标分别反映了分类器性能的两个方面。

precision 和 recall 值都是越高越好，最好的情况是 precision 和 recall 值均为 1，说明所有的样本都被正确分类了。但是事实上这两者在某些情况下是有矛盾的。为了兼顾精确率和召回率，我们使用综合评价指标 F-measure。F-measure 的定义公式如下：

$$F_\beta = \frac{(\beta^2 + 1)P * R}{\beta^2(P + R)} \tag{4-5}$$

其中 β 是参数，P 是精确率（precision），R 是召回率（recall），当参数 β=1 时，就是最常见的 F1-measure，即认为准确率和召回率同等重要。本文即使用 F1 作为识别效果的综合评估指标。

$$F1 = \frac{2 * P * R}{P + R} \tag{4-6}$$

4.4 实验结果与分析

本文通过对多个实验结果的对比分析验证设备指纹识别技术的有效性。首先进行了设备识别实验，在已有的数据集上对比了各特征指纹对表征设备身份的有效性和各分类器的识别性能。此外，本文还讨论了特征空间变化对识别效果的影响。

4.4.1 实验 1：设备识别实验结果与分析

我们基于 3.3 节中提到的特征指纹生成方法，从帧间隔时间、帧大小和传输速率三个参数中提取了相应的特征指纹，并分别在随机森林、支持向量机、k 最近邻法和朴素贝叶斯五种分类算法下进行实验。我们将 4.3.2 中所述的训练和测试过程重复了 50 次，最后对这 50 次实验结果取平均值，结果如表 4-1 所示。

表 4-1　设备识别实验结果

分类器	评估指标	帧时间间隔	帧大小	传输速率	融合特征
随机森林	precision	0.9301	0.9740	0.9234	0.9930
	recall	0.9314	0.9798	0.9300	0.9670
	F1	0.9277	0.9757	0.9221	0.9783
支持向量机	precision	0.8792	0.9636	0.9103	0.9899
	recall	0.9416	0.9737	0.9411	0.9522
	F1	0.9035	0.9655	0.9197	0.9678
K 最近邻	precision	0.8730	0.9600	0.8915	0.9741
	recall	0.9238	0.9581	0.9434	0.9247
	F1	0.8942	0.9566	0.9112	0.9450
朴素贝叶斯	precision	0.8493	0.9433	0.8673	0.9055
	recall	0.8825	0.9408	0.8930	0.9143
	F1	0.8600	0.9386	0.8753	0.9044

从表 4-1 中可以看出，最好的识别结果为 0.9930 的准确率、0.976 的召回率和 0.9783 的 F1。该结果是十分令人鼓舞的，它证实了网络流量中确实存在能够标识设备身份的信息。

对比各种分类器的效果可以看出，随机森林得到的设备识别效果优于其它三种分类器，无论使用哪个特征指纹，随机森林得到的准确率、召回率和 F1 值均在 0.9 以上。其原因是随机森林使用了无偏估计处理泛化误差，因此泛化能力较其他模型强。我们拟将随机森林作为设备识别的主分类器，其他三种分类器的识别结果作为辅助判断。

其它三种分类器在设备指纹识别问题中效果也不错，F1 值都在 0.9 左右。我们还注意到大部分情况下召回率比准确率略高一点，这说明分类器将较多的负例样本划分为正例，导致准确率略低。

对比各种特征来看，基于多特征融合的特征指纹在标识设备身份时更加有效，四种分类器的 F1 都在 0.9 以上，高于其他三种特征指纹。其原因在于基于概率密度的三种特征指纹都仅代表了设备在某一方面的属性，并不能完全表征设备的身份属性，将三种特征指纹融合形成的特征指纹可以很好地弥补单一特征指纹的不足，能更加完善地表征设备的身份属性。但整体上看，其他三种特征指纹的识别效果都比较好，识别的精确度都在 0.9 左右，其中从帧大小中提取的特征指纹在使用随机森林分类器时 F1 值为 0.9757，与融合特征指纹的识别效果十分接近。

此外，需要指出的是无线设备指纹识别技术的平均识别时间为 12.5 秒，该时长完全能够满足现实应用的要求。这表明从网络流量中提取的特征指纹能够在短时间内识别出设备，能够满足大部分设备识别场景的即时性需求。其原因在于我们采用一系列的数据处理和特征指纹提取手段，可以在保证识别精度的同时大幅度降低消耗的时间。

4.4.2 实验 2：特征空间变化对识别效果的影响

本实验检验了指纹生成过程中相关参数的变化对识别结果的影响。在 3.3 中所述的特征指纹生成技术中，我们先将数据划分为大小相同的若干组，然后将每组数据划分为若干个窗口，统计组内数据落在每个窗口内的频率。在 4.4.1 节的实验中，我们设定每组的样本量为 300，即 group_size=300；窗口的数目为 10，即 bin_size=10，识别结果如表 4-1 所示，本节中我们将探讨 group_size 和 bin_size 的变化对设备识别的影响。

1）分组变化对识别效果的影响

在设备识别实验中，我们固定每组的样本量为 300。本实验中我们令 group_size 从 100 到 500 以步长为 50 变化，bin_size 仍固定为 10。此外，在设备识别实验中对设备的选取是随机的，本实验中为了排除因设备不同对结果造成的影响，我们固定地选取两个设备数据进行实验。图 4-5 至图 4-8 分别展示了四种分类器在不同的特征指纹上的识别效果，为了避免图中线条过多影响阅读，我们仅绘制了 F1 的变化曲线，图中横坐标为 group_size，纵坐标为 F1 的值。

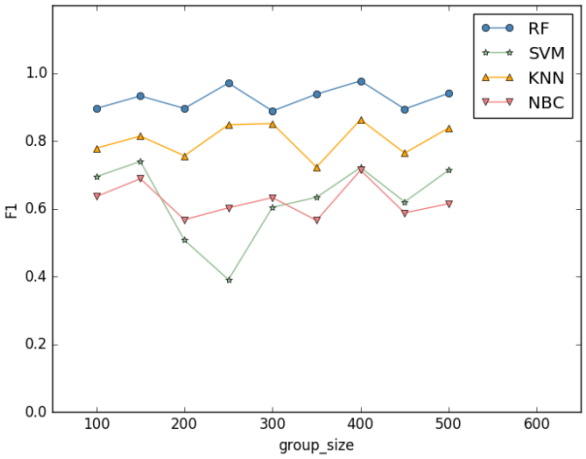

图 4-5　group_size 变化对识别效果的影响（IAT）

图 4-5 为基于帧间隔时间（IAT）的概率密度特征指纹识别结果，图中结果表明 group_size 的变化对于各个分类器的识别效果影响不大。其中随机森林、K 最近邻和朴素贝叶斯得到的 F1 波动很小，但是当 group_size 为 200 和 250 时，支持向量机的结果较差。综合各个分类器的效果来看，对于基于帧时间间隔的特征指纹，最佳 group_size 为 400，此时随机森林的 F1 接近 1，而其它三种分类器的 F1 在 0.8 左右。

图 4-6 为基于帧大小（FrameSize）的概率密度特征指纹识别结果，从图中可以看出每组的样本量变化对四种分类器的识别效果几乎无影响，曲线波动非常小。但是在 group_size 为 200 时，随机森林和 KNN 的 F1 值低于其它情况，降到了 0.8 左右。整体上来看，对于基于帧大小的特征指纹，将 group_size 设为 400 到 500 之间均可，应当避免将 group_size 设置为 200。

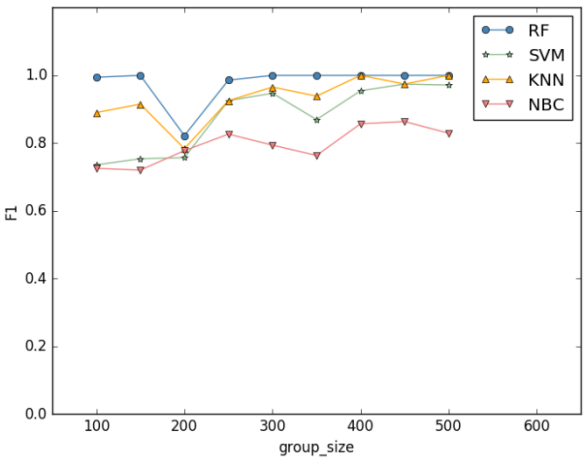

图 4-6　group_size 变化对识别效果的影响（FrameSize）

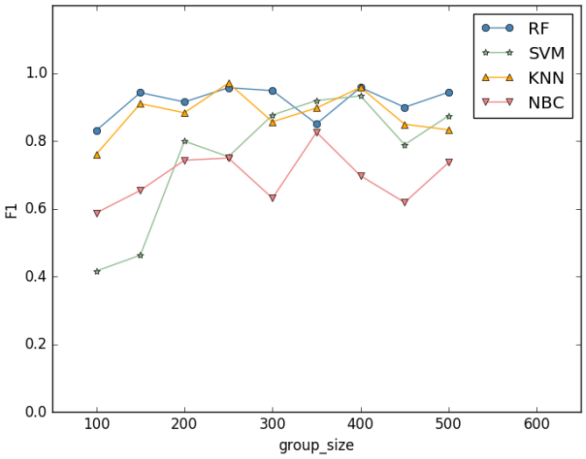

图 4-7　group_size 变化对识别效果的影响（TransRate）

图 4-7 展示了基于传输速率（TransRate）的特征指纹识别效果。图中带下三角和星号标记的线波动比较大，说明 group_size 的变化对支持向量机和朴素贝叶斯的影响较大；而图中蓝色和黄色的线波动很小，说明随机森林和 K 最近邻对于 group_size 的变化不敏感。当 group_size 为 400 或 500 时，四种分类器的 F1 值都比较高。

基于融合特征的识别结果如图 4-8 所示。从图上可以看到，除了朴素贝叶斯的识别结果略有波动外，各个分类器的 F1 值变化都不大，识别效果都比较稳定。且相比于基于概率密度的特征指纹，在各个 group_size 下融合特征的识别精度都要略高一些。这一结果表明融合特征指纹对 group_size 的变化具有一定的鲁棒性。

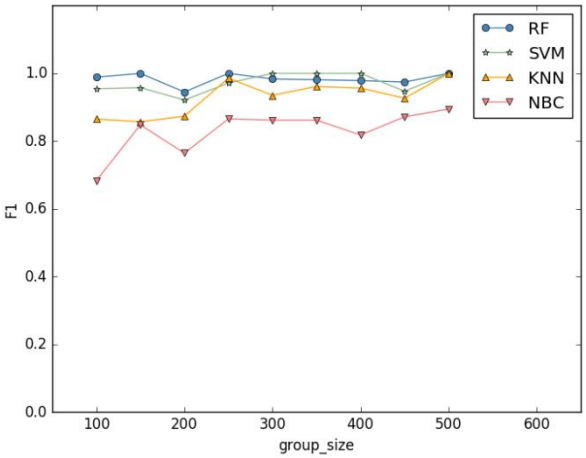

图 4-8　group_size 变化对识别效果的影响（融合特征）

2）分窗变化对识别效果的影响

在设备识别实验中，窗大小为 10（bin_size=10），即将每组数据划分为 10 个窗口，统计落在每个窗口内数据的频率。在本实验中，我们令 bin_size 从 5 到 15 以 1 为步长变化。为了排除其他因素对实验结果的影响，我们将 group_size 设为 300，且固定地选取两个设备的样本进行实验。图 4-9 至图 4-12 分别展示了四种分类器在不同的特征指

纹上的识别效果，为了避免图中线条过多影响阅读，我们仅绘制了 F1 的变化曲线，图中横坐标为 bin_size，纵坐标为 F1 的值。

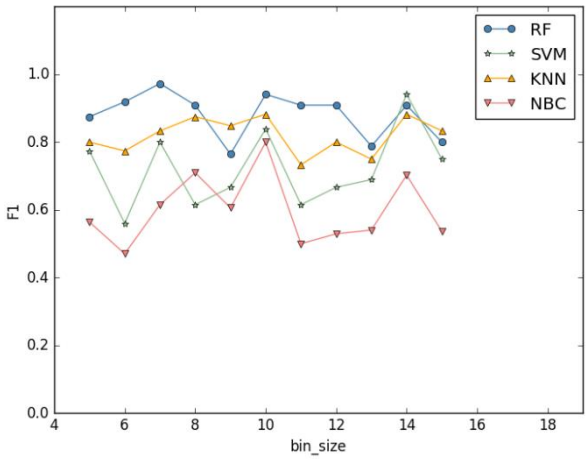

图 4-9　bin_size 变化对识别效果的影响（IAT）

图 4-9 展示了以帧时间间隔为特征参数时，bin_size 的变化对识别结果的影响。图中四条曲线的波动都比较明显，说明从帧时间间隔中提取的特征指纹鲁棒性较差，对 bin_size 的变化敏感。对比图中各点，可以看出当 bin_size 为 10 时识别效果最好。

图 4-10 是基于帧大小（FrameSize）的特征指纹的识别结果。相比于图 4-9（IAT）中曲线较大的波动性，以帧大小作为特征参数的设备指纹鲁棒性更好。当 bin_size 变化时，随机森林、支持向量机和 K 最近邻的识别效果都比较稳定，朴素贝叶斯分类器则略有波动。当使用随机森林、支持向量机和 K 最近邻三种分类器时，bin_size 值可以为 5 到 15 之间的任意一个。

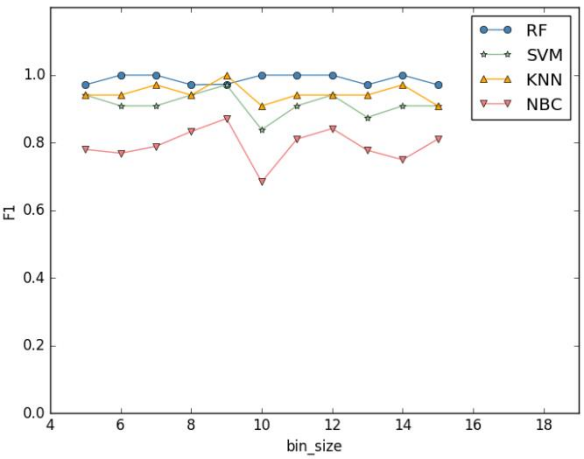

图 4-10　bin_size 变化对识别效果的影响（FrameSize）

下图是 bin_size 变化时，以传输速率作为特征参数的实验结果。图中红色和绿色的曲线波动较大，说明 SVM 和朴素贝叶斯对于 bin_size 的变化敏感。观察图中各曲线可以看出，当 bin_size 为 10 时，各个分类器的识别效果都比较好，F1 均在 0.8 左右。

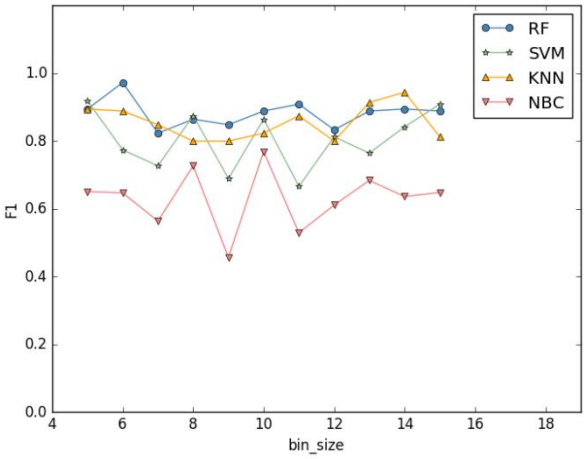

图 4-11　bin_size 变化对识别效果的影响（TransRate）

图 4-12 是融合特征的识别结果。当 bin_size 从 5 到 15 变化时，除朴素贝叶斯的识别结果略有波动外，其余三种分类器的 F1 均在 0.9 左右。与基于概率密度的特征指纹相比，融合特征指纹不仅精度高，且鲁棒性好，因此融合特征指纹更加适合用于表征设备的身份信息。

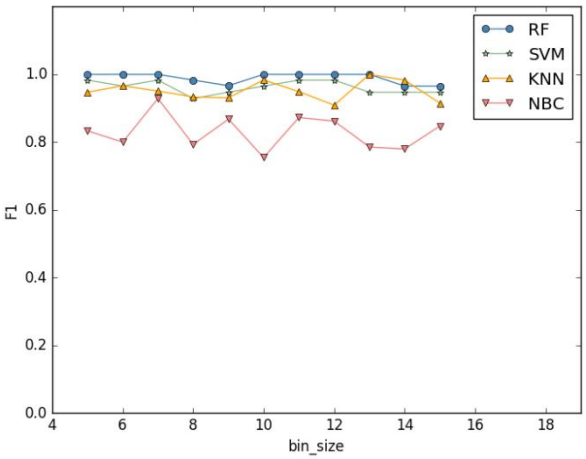

图 4-12　bin_size 变化对识别效果的影响（融合特征）

对上述实验结果进行对比分析后不难发现：在四种分类器中随机森林的对于无线设备识别问题有着更好的表现，精度高且鲁棒性好；相比于单独的特征指纹，融合特征指纹能更加全面地表征设备身份信息。

4.5 本章小结

本章首先介绍了无线设备指纹识别的整体流程，紧接着介绍了实验中用到的四种分类器：随机森林、支持向量机、K 最近邻和朴素贝叶斯。然后将从数据集中提取的特征指纹输入到上述各个分类器中进行训练和测试，计算出准确率（precision）、召回率

（recall）和 F1 这三个评估指标，并使用 F1 来综合评估分类器的性能。从实验结果中可以看出：使用随机森林分类器并采用多特征融合的特征指纹时，识别效果最佳，准确率为 0.9930、召回率为 0.9670、F1 值为 0.9783；对比各个分类器来看，随机森林的识别效果最好，在每种特征指纹上 F1 值均在 0.9 以上；对比各个特征指纹来看，融合特征指纹在标识设备身份时更加有效，在每个分类器下得到的 F1 值均在 0.9 左右。此外，我们还讨论指纹生成过程中相关参数的变化对识别效果的影响，实验结果表明：随机森林的分类效果比较稳定，始终能保持较好的识别结果；融合特征指纹对于相关参数的变化不敏感，无论特征指纹生成过程中的参数如何变化，融合特征指纹均能很好的体现设备身份信息。

5　无线设备指纹识别原型系统的开发与实现

前几章的研究和实验结果表明，网络流量中确有包含设备身份信息的特征指纹，可用于设备的识别与认证。基于此，我们设计并开发出无线网络设备识别原型系统，本章从需求分析、架构设计、系统实现以及功能测试四个方面对原型系统进行介绍。

5.1 原型系统需求分析

原型系统的开发依托于课题组和电子信息控制重点实验室的合作项目“基于行为特征认知的无线网络目标指纹识别技术研究”，该原型系统主要用于在该单位内部网络中对网络内连接的各种设备进行不间断地、被动式、实时性身份识别及认证，在设备正常使用的同时，网络管理人员可以自发性完成设备网络流量数据捕获、设备特征指纹的形成、无线设备指纹生成、设备身份识别认证等功能。此原型系统也可以应用于其他对安全性要求比较高的政府、企业等单位内部网络。

本文在对无线设备指纹识别技术研究的基础上，开发了一套完整的原型系统。原型系统是基于 B/S（Browser/Server）架构进行搭建的，前端页面（Browser）上展示系统的功能，而这些功能的实现则主要由服务器端（Server）完成。在这种结构下，用户只需要一台安装了浏览器的主机即可访问系统。B/S 架构的优点在于几乎所有的功能都在服务器上实现，当系统更新时，只需修改服务器上的代码即可，无需用户重新下载安装应用软件。但是由于浏览器（Browser）种类众多，同一个页面元素在不同的浏览器上显示的效果可能不同，甚至无法显示，因此在系统开发时应当考虑各种浏览器的兼容性，使系统页面在任何浏览器下的显示效果都是统一的。本文开发的无线设备指纹识别原型系统服务器端代码是基于 Python 平台开发的，使用 Python 的 Django 架构搭建 Web 服务。此外，此原型系统的设计与开发由作者独立完成，兼顾各种浏览器的兼容性问题难度较大，作者在进行浏览器端页面设计时仅考虑了 Chrome 浏览器。

原型系统用例图如图 5-1 所示，用例图主要描述了用户与系统之间如何交互。用户进入系统，捕获接入到局域网的无线设备网络流量；从特征参数中提取数据再进行预处理，按照 3.3 节中所述方法提取特征指纹；形成特征指纹后将其输入到分类器中训练和测试，根据测试结果判断是否为新设备，如果是指纹库中已知设备则识别出是哪一个；如果是未知设备则由用户添加设备 ID，并将设备指纹上传到指纹库中，更新指纹库。

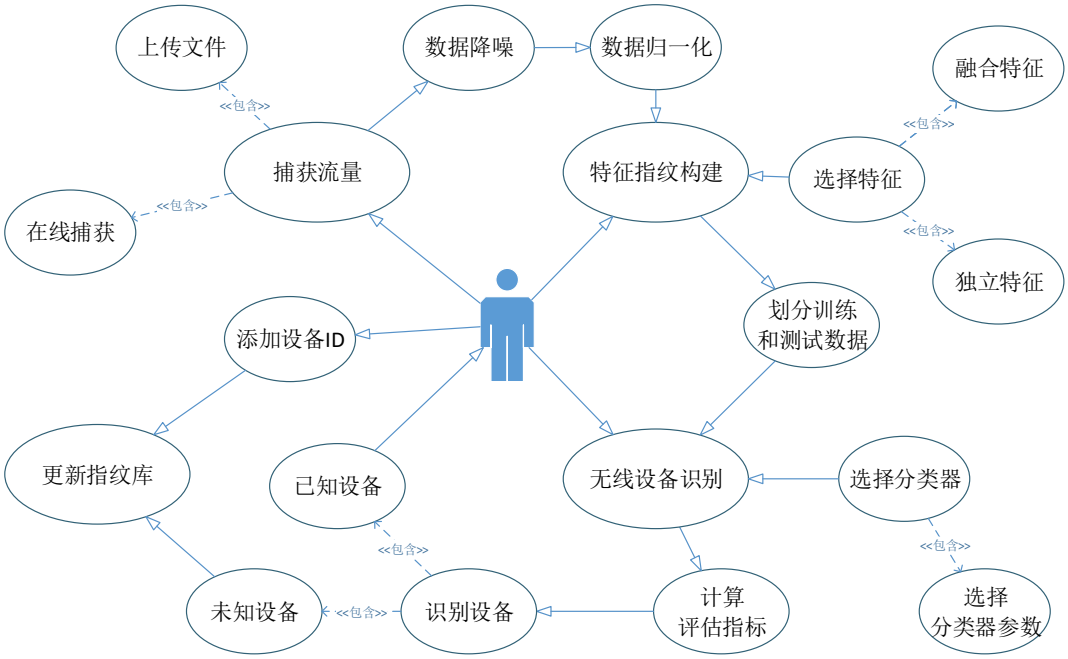

图 5-1　原型系统用例图

原型系统主要分为网络流量捕获、特征指纹的形成、无线网络设备识别和指纹库的更新四个模块。每个模块的需求如下：

1）流量数据捕获：流量数据的捕获是在线进行的，用户也可以在系统中实时采集连接入网络的设备数据，这样用户不仅可以实时观察设备的流量数据，也可以在数据规模达到一定程度时将数据保存成文件进行设备识别认证。通过开发第三方 Python 工具库，系统可以捕获接入到局域网的无线设备网络流量，并将其上传至后台服务器；捕获流量的过程对于对用户是透明的，不会干扰设备的正常使用；用户可在 Web 页面控制什么时候开始捕获流量，什么时候采集截止；用户可随时查看数据采集的情况，如采集的设备数目和数据量等；当数据量足够时，系统提示可以结束采集，数据量过少时也应给出相应提示。

2）特征指纹的形成：后端处理捕获到的流量数据，提取帧间隔时间、帧大小和传输速率三个参数的数据，对其进行降噪和归一化处理，并从中提取特征形成设备指纹；前端可实时查看参数提取、数据降噪和归一化的结果，结果以散点图的形式展示此模块中用户可控制的输入有：1）降噪的阈值；2）特征指纹的构建方法，有基于概率密度的特征指纹和基于特征融合的特征指纹，如果是前者，还需确定使用三种参数（帧间隔时间、帧大小和传输速率）的哪一个；3）每组样本的大小（group_size）和窗大小（bin_size）；前端获取用户的各项输入，传递给后端；后端接收参数形成特征指纹。

3）无线网络设备识别：将待识别的设备指纹输入到分类器中，完成无线设备模型的构建与评估，计算 precision，recall 和 F1 三项评估指标；通过设置评估阈值判断设备身份，若与指纹库中某个设备指纹比对结果的 F1 值（或 precision/recall）大于该阈值，则认为这是指纹库中已有设备，返回设备 ID；若 F1（或 precision/recall）小于该阈

值，则认为这是一个新的设备，用户可以选择将其加入指纹库中；用户可以选择使用哪一个分类器（随机森林、支持向量机、K 最近邻和朴素贝叶斯），也可以选择使用多个分类器，系统会展示多个分类器的识别结果；此外用户还需要选择使用的特征指纹，是基于概率密度的特征指纹还是融合特征指纹；前端获取用户的输入，传给后端；后端接收参数给出识别结果再传给前端；前端接收结果并展示在页面上。

4）指纹库的构建：我们为系统建立了一个基准指纹库，包含 3.2.3 中所述 23 台设备的特征指纹；当系统检测到有未知设备时，用户可以选择是否将该设备指纹加入到指纹库中进行更新，此时用户需要设置未知设备的 ID；这部分功能在后台服务器上完成。

5.2 原型系统架构设计

根据 5.1 中对原型系统的需求分析，我们设计了如图 5-2 的框架图和功能模块设计表（表 5-1）。整体来说，原型系统采用 B/S 架构，服务器端完成数据采集、预处理和设备识别，负责系统功能的实现，而前台负责用户与系统的交互以及结果的展示。

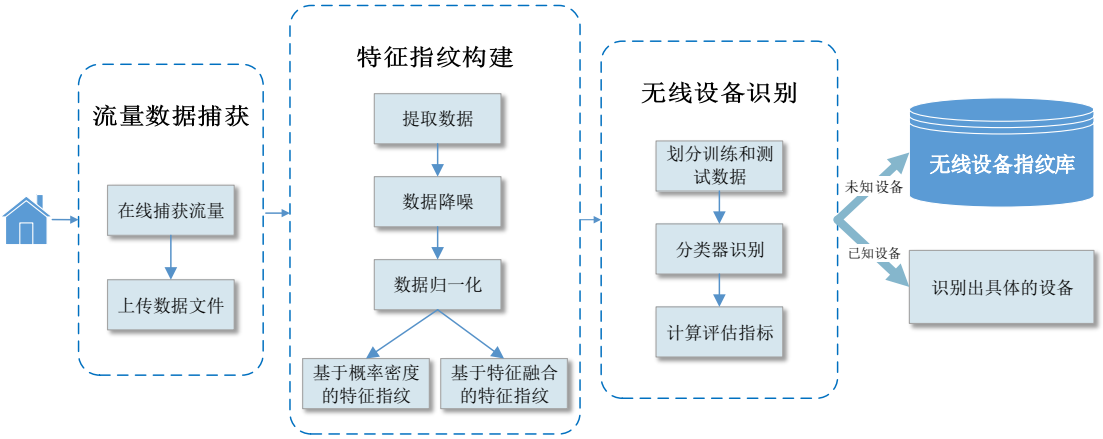

图 5-2　原型系统框架图

表 5-1　原型系统功能模块表

模块名称	功能介绍
流量数据捕获模块	搭建无线网络环境，编写程序实时采集连接到网络中的设备发出的流量，将生成的文件保存在系统后台；从前台可视化模块读取数据采集参数，在界面上显示数据相关信息
特征指纹生成模块	读取捕获的网络流量文件，从中提取各项参数；完成数据降噪和归一化；根据用户选择的特征构建方法形成设备指纹并保存为文件形式
无线网络设备识别模块	验证系统的核心模块，获取用户选择的分类器及各项参数，对设备指纹进行训练和测试，计算评估参数
指纹库更新	根据识别的结果，将未知设备的指纹加入指纹库中

5.3 原型系统实现

原型系统采用 B/S 架构的模式，通过 Python 的 Django 架构搭建 Web 服务。前端和服务器端的数据交换基于 JSON（JavaScript Object Notation，JS 对象标记）数据格式，使用 Python 自带的函数可方便地进行 JSON 格式数据的转换与解析。JSON 以键值对的形式保存数据，字符串、数字、对象、数组等类型的数据均可通过 JSON 来表示，是一种理想的数据交换语言。下面分别阐述原型系统各模块的设计与实现。

5.3.1 流量数据捕获模块

第 3 章的实验中无线网络环境是一个包含两个路由器的网络，每次只能捕获一台设备的流量。原型系统设计时为了能够同时采集多台设备的数据，搭建的无线局域网中只包含一个路由器。数据流量捕获模块的时序图如图 5-3 所示。

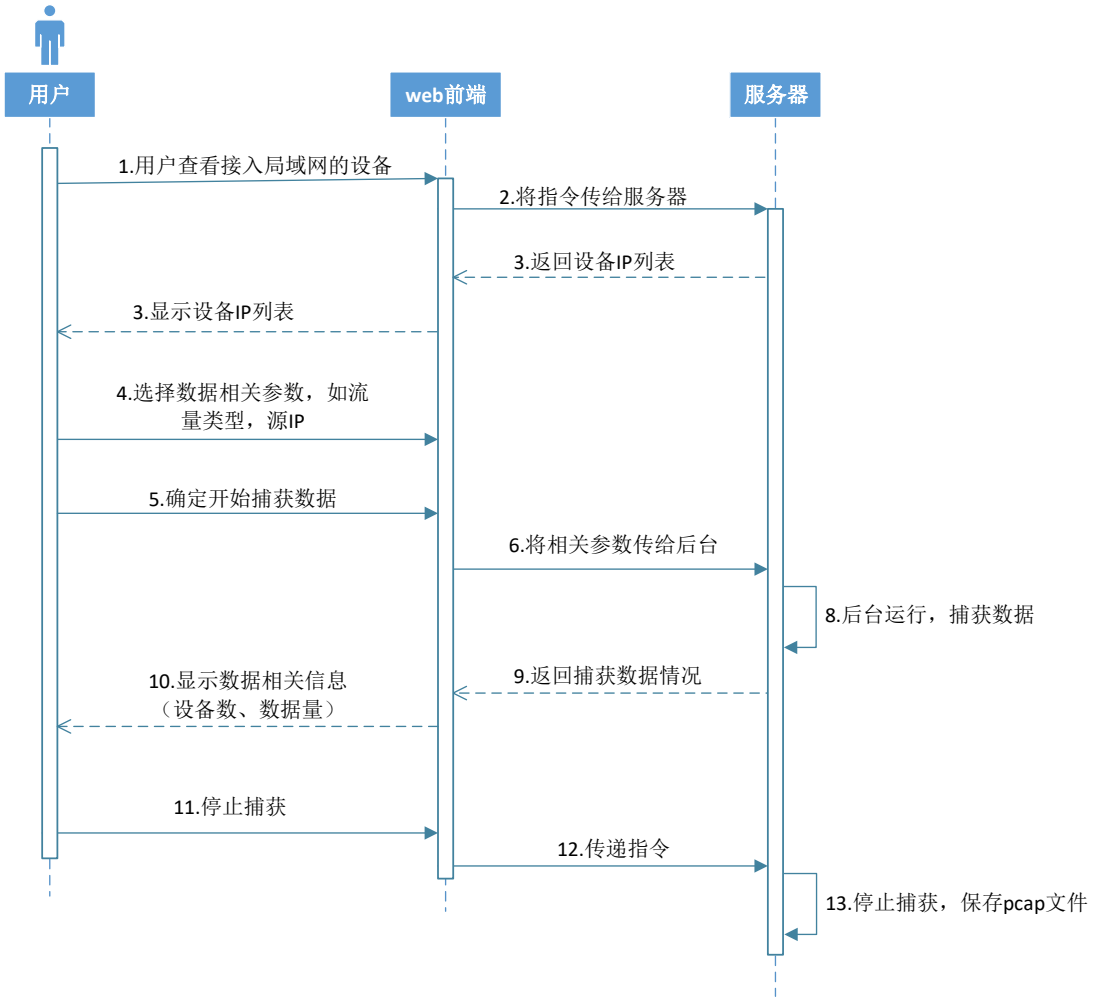

图 5-3　流量数据捕获模块时序图

Web 页面上显示接入局域网中的设备数目，用户可以选择其中的一个或多个进行抓包。抓包主要依赖 Python 的 scapy 库，scapy 是一个可用于网络嗅探的非常强大的第

三方库，可以方便地抓包与解析包。用户可以实时查看每台设备当前捕获的数据量，当数据量足够时可随时停止，若数据量不够系统则会给出提示。

停止采集后，系统将采集到的数据保存为 pcap 文件，文件名为设备 IP。

5.3.2 特征指纹形成模块

特征指纹形成模块基于 3.3 节介绍的指纹提取方法开发，用户与系统交互的时序图如 5-4 所示。特征指纹形成模块大致可分为流量参数的提取、数据降噪、数据归一化、特征指纹的形成几个步骤，下面详细介绍每个步骤的前端和后端设计。

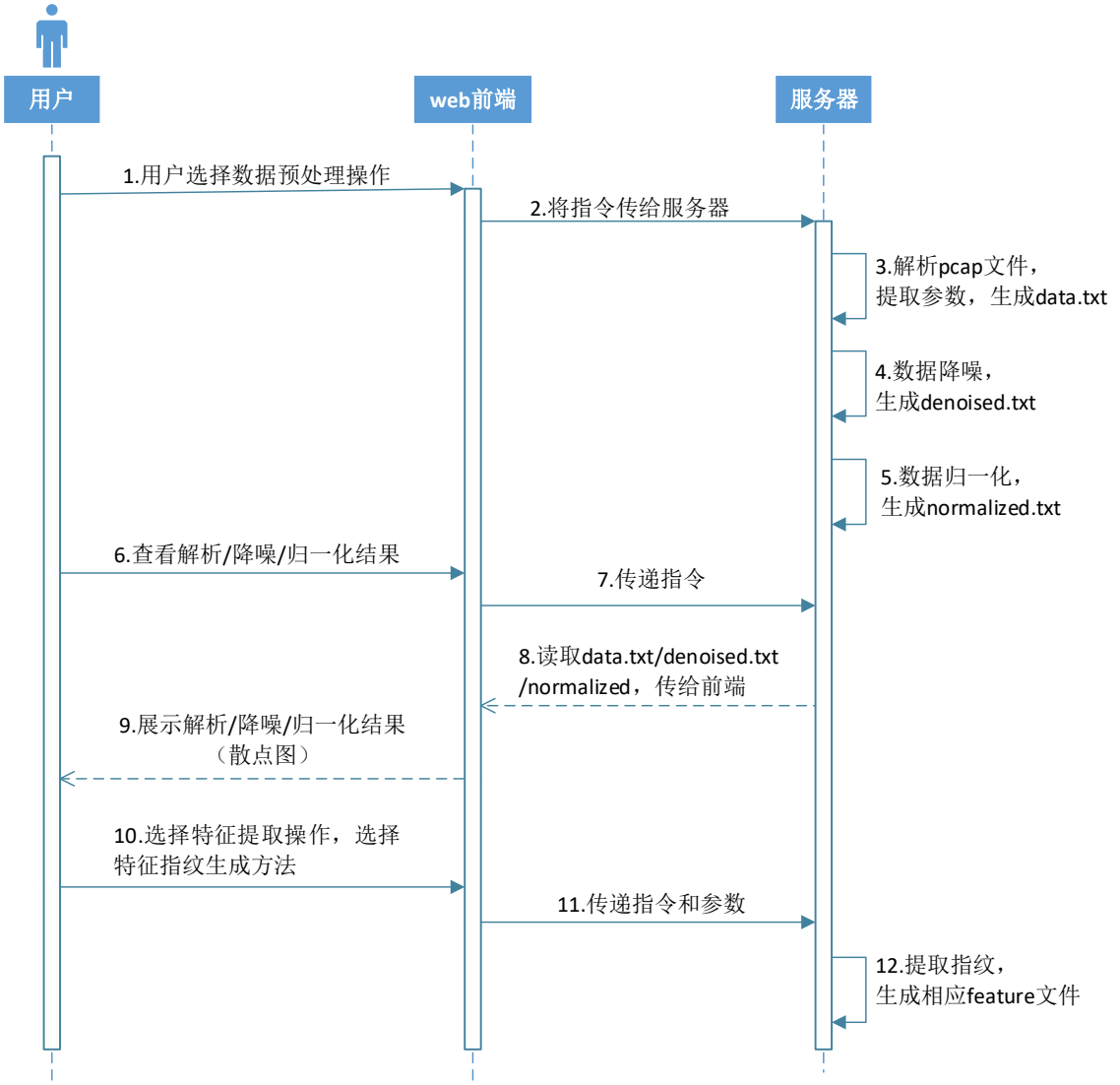

图 5-4　特征指纹生成模块时序图

1.　流量参数的提取

流量数据捕获模块的结果是将实时捕获或上传的流量文件以 pcap 形式保存在服务器上。在本模块中，后台读取 pcap 格式的流量文件，从中提取出每条数据帧的帧时间间隔、帧大小和传输速率三个参数，并保存为易于处理的 txt 格式的文件，文件名为 data.txt。数据格式如式 5-1：

$$X = [X_{IAT}, X_{FS}, X_{TR}] \tag{5-1}$$

其中 X_{IAT} 为数据帧的帧时间间隔，为 X_{FS} 帧大小，为 X_{TR} 传输速率。

Web 页面上可以显示各项参数的散点图，系统后台在用 Python 绘制好图形并保存到固定的目录下，前端读取相应的图片文件并进行显示。

2. 数据降噪

后台读取 data.txt 文件，根据各项参数的阈值对其进行过滤，注意降噪结果只保留三个参数均小于相应阈值的样本，即便样本只有一个参数大于相应阈值，该样本也会被舍弃，结果保存为 denoised.txt。同样地，系统也会绘制降噪结果的散点图，前端读取图片文件并显示在 Web 页面上。

3. 数据归一化

后端读取 denoised.txt 文件，采用 min-max 方法对每项参数的数据（即 denoised.txt 文件中的每一列）归一化处理，归一化结果保存为 normalized.txt 文件。系统绘制相应的散点图并展示在 Web 页面上。

4. 特征指纹的构建

前端获取用户设置的各项参数，包括：特征指纹提取方法（基于概率密度的特征指纹/基于特征融合的特征指纹）、分组大小（group_size）和窗大小(bin_size)，以字符串的格式传给后端。

后端接收前端传递的数据，读取 normalized.txt 文件，根据 3.3 节中所述方法提取相应特征。对于基于概率密度的特征指纹提取方法，本阶段生成三个文件：IAT_feature.txt，FrameSize_feature.txt，TransRate_feature.txt，分别存储每个参数生成的特征；对于融合特征指纹，将生成一个 mix_feature.txt 文件，存储融合的特征指纹。

值得指出的是，为了避免随着用户使用系统次数的增加，系统数据文件越来越多占用空间的问题，上述过程中生成的 data.txt、denoised.txt 和 normalized.txt 文件均为临时文件，在用户退出系统后均会被删除。且若此设备被为指纹库中已有的设备，则四个 feature.txt 文件也会被删除；如果为新设备，则将四个 feature.txt 文件上传到指纹库中。

5.3.3 无线设备识别模块

无线设备指纹识别模块的功能是根据特征指纹形成模块中生成的设备指纹，识别出该设备是指纹库中的哪一个设备，或者该设备为未知的新设备。该模块依赖于特征指纹生成模块的结果，需要将特征指纹输入到分类器中建立相应的设备模型，再计算出 precision、recall 和 F1 这三个指标。由于指纹库中有多个设备的指纹，我们仍然是将多分类问题转化为双分类问题：每次从指纹库选取一个设备的数据，将待测设备指纹与其输入到分类器中进行比对，直到找到与待测设备相似的指纹，或是指纹库中所有的指纹均被比对过。

无线设备识别模块的时序图如图 5-5 所示，用户在 Web 页面上选择要使用的分类器和相关参数，服务器读取待识别设备的指纹以及指纹库中的设备指纹，按照十折交

叉划分数据并输入到分类器中进行训练和测试，再将识别结果结果返回给用户。

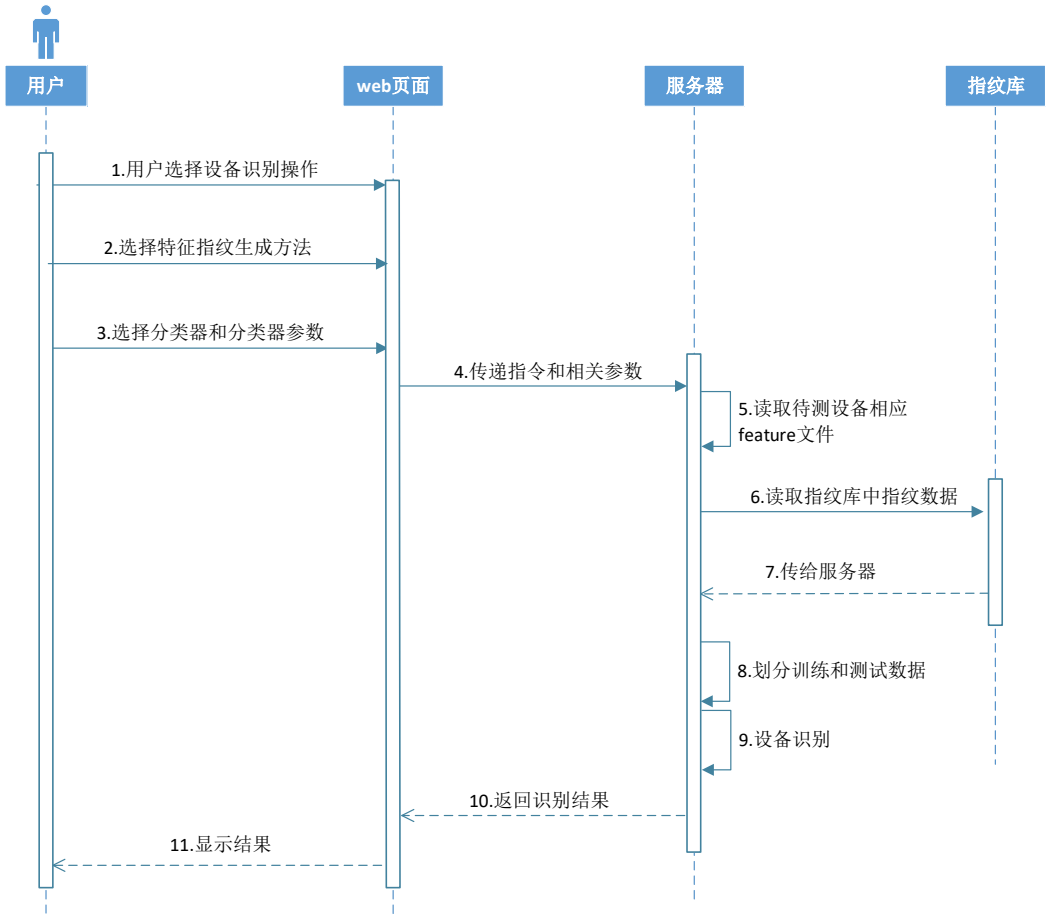

图 5-5　无线设备识别模块时序图

1.　十折交叉划分训练数据和测试数据

根据用户选择的特征提取方法，读取待识别设备和指纹库中的设备指纹的。根据每台设备的数据量，按比例将其随机划分为十份，其中九份用于训练，一份用于测试。训练和测试数据以及其相对应的类别标签分别存储在列表 train、test、train_label、test_label 中。这一部分无需前端参与。

2.　四种分类器的实现及性能评估

用户在页面上选择要使用的分类器（随机森林、支持向量机、K 最近邻、朴素贝叶斯）和特征指纹，前端将用户选择的分类器传给后端。

后端使用 Sklearn 中的接口计算各个分类器的识别结果。Sklearn 是 Python 的重要机器学习库，其中封装了大量的机器学习算法。后端调用 Sklearn 中的随机森林、支持向量机、朴素贝叶斯、K 最近邻这四种算法的接口，完成无线设备指纹的训练和测试。十折交叉划分的数据中 train 和 train_label 用于无线设备模型的构建，test 和 test_label 则用于测试。分类器输出 predict_label，即对于测试样本的预测标签，根据预测标签可以计算出 precision、recall 和 F1 值。如果待测指纹与指纹库中某个指纹比对得到的

precision、recall 和 F1 均大于 0.5，则说明分类器可明显地区别出两个设备，即说明两个设备不是同一类；反之，若分类器得到的 precision、recall 和 F1 值都在 0.5 左右，说明分类器无法明显区分两个设备，我们可以认为这是同一台设备，后端记录指纹库中相应设备的 ID。

后端把 precision、recall、F1 值及设备 ID 传给前端，前端进行显示。

5.3.4 指纹库更新模块

本文所构建的指纹库不仅为设备识别提供依据，还将作为公开的数据集，供其他的研究者使用。指纹库的构建包含基准指纹库和指纹库的更新两个部分。我们先将若干设备的指纹放入指纹库中作为基准指纹库，得到新的设备指纹后再将其加入指纹库中，指纹库更新的时序图如图 5-6 所示。

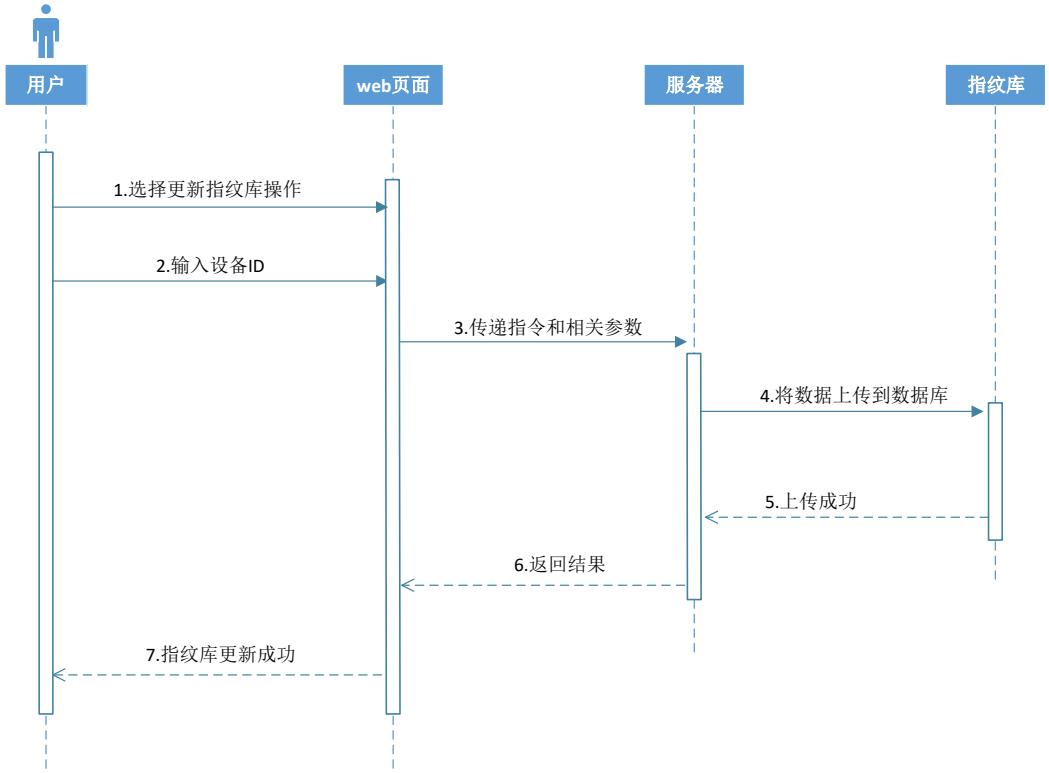

图 5-6　指纹库更新时序图

1. 基准指纹库

基准指纹库中包含了本文采集的 23 台设备的指纹，这 23 台设备的详细信息见 3.2.3 小节。为了标识每台设备，每个设备都有自己的 ID，设备 ID 一般是设备型号和设备编号组成的字符串，如“iphone7_1”。每个设备指纹包含 3 项参数的原始数据和 4 个 feature 文件，以 ID 为“iphone7_1”的设备为例，其四个指纹文件分别为 iphone7_1_IAT.txt，iphone7_1_FrameSize.txt，iphone7_1_TransRate.txt，iphone7_1_mix_feature.txt.

2. 指纹库的更新

指纹库更新的时序图如图 5-6 所示。在设备识别的基础上，如果结果显示该设备为未知的新设备，系统询问用户是否需要保存该设备指纹，若需保存则用户还需为设备命名，系统根据用户的命名再加上编号即为该设备 ID。若用户在之前的步骤中已生成了设备的四个特征文件，则将文件名分别改成设备 ID 加上特征名，上传到指纹库，若特征文件不足，则自动补足后上传至指纹库，完成指纹库的更新。

5.3.5 Web 服务器端开发

服务器端的开发采用 Python 语言，具体使用 Python 的 Django 框架搭建 Web 服务。Django 采用 MTV 框架，其中 M 代表模型（model），负责业务对象和数据库的关系映射；T 是模板（Tamplate），功能是向用户展示系统的页面；V 为视图（View），负责系统的业务逻辑，并根据功能的需要调用 Model 和 Tamplate。除了上述三层之外，Django 还设置了一个 URL 分发器，它的功能是将 HTTP 请求分发给相应的视图。当服务器接收到来自前端的 HTTP 请求后，首先判断是来自哪个 URL 的请求，然后根据 URL 配置文件找到相应的视图，视图处理结束后将响应传给模板，模板根据响应重新渲染页面再展示给用户。其处理流程如图 5-7 所示。

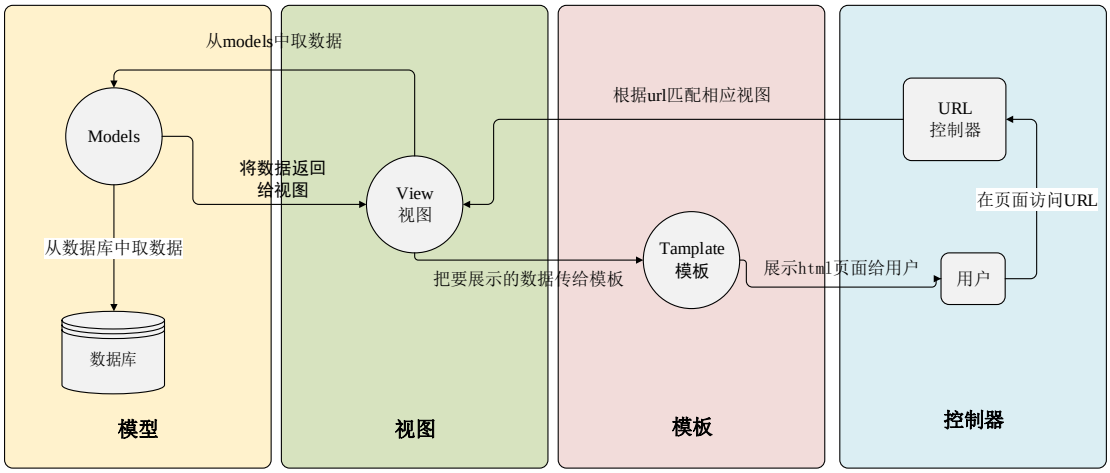

图 5-7　Django 处理流程图

Django 的 MTV 框架逻辑清晰，易于使用，各个组件之间是松耦合的，修改其中一个组件不会对其他的部分造成影响，便于维护和开发。下面分别从 MTV 的三个方面介绍具体实现。

1. 模型（model）

模型是一个抽象层，用来构建和操作 Web 应用中的数据。模型中包含所存储数据的必要字段和行为。Django 对各种数据库提供了很好的支持，本论文采用 SQLite 作为服务器端数据存储系统。无线设备指纹库需要 7 张表，分别为 device_list、IAT、FS、TR、IAT_FP、FS_FP 和 TR_FP。

device_list 存储设备指纹清单，其对应的数据结构如表 5-2 所示。主键为设备编号，从 1 开始按照顺序递增，设备 ID 是设备名（用户设置）和设备编号组成的字符串，其余各列分别存储该设备的数据清单，即对该设备，存储了哪些数据。

表 5-2 Device_list 表字段信息

序号	名称	字段名	数据类型
1	设备编号	No	INT
2	设备 ID	device_ID	VARCHAR
3	帧间隔时间	IAT	VARCHAR
4	包大小	FS	VARCHAR
5	传输速率	TR	VARCHAR
6	帧间隔时间指纹	IAT_FP	VARCHAR
7	包大小指纹	FS_FP	VARCHAR
8	传输速率指纹	TR_FP	VARCHAR

IAT、FS 和 TR 这三张分别存储从原始设备流量数据中提取的帧间隔时间、包大小和传输速率几个参数的数据。这三张表的数据结构相同，只有第 3 列存储的数据不同，主键为数据序号，由系统自动生成；第 2 列为设备 ID；IAT、FS 和 TR 三张表的第三列分别为从 TCP 流量帧中提取的帧时间间隔、帧大小和传输速率三种参数的数据。表 5-3 中展示了 IAT 表的数据结构。

表 5-3 IAT 表字段信息

序号	名称	字段名	数据类型
1	数据序号	No	INT
2	设备 ID	device_ID	VARCHAR
3	帧间隔时间	IAT	DOUBLE

IAT_FP、FS_FP 和 TR_FP 这三张表分别存储从各参数中提取的特征指纹。这三张表的数据结构相同，区别只在于第 3 列存储的特征不同。主键为数据序号，由系统自动生成；第 2 列为设备 ID；第三列为从三项参数中提取的特征指纹，IAT_FP、FS_FP 和 TR_FP 的第三列分别为从帧时间间隔、帧大小、传输速率中提取的特征指纹。IAT_FP 表的数据结构如表 5-4 所示。

表 5-4 IAT_FP 表字段信息

序号	名称	字段名	数据类型
1	数据序号	No	INT
2	设备 ID	device_ID	VARVHAR
3	帧间隔时间特征	IAT_FP	DOUBLE

需要指出的是，我们在设计数据库时并未建立基于多特征融合的特征指纹数据表，原因是融合特征指纹可以从 IAT_FP、FS_FP 和 TR_FP 三张表中获得，可以不用专门建立数据表。

上述七张表之间的关系如图 5-8 所示。

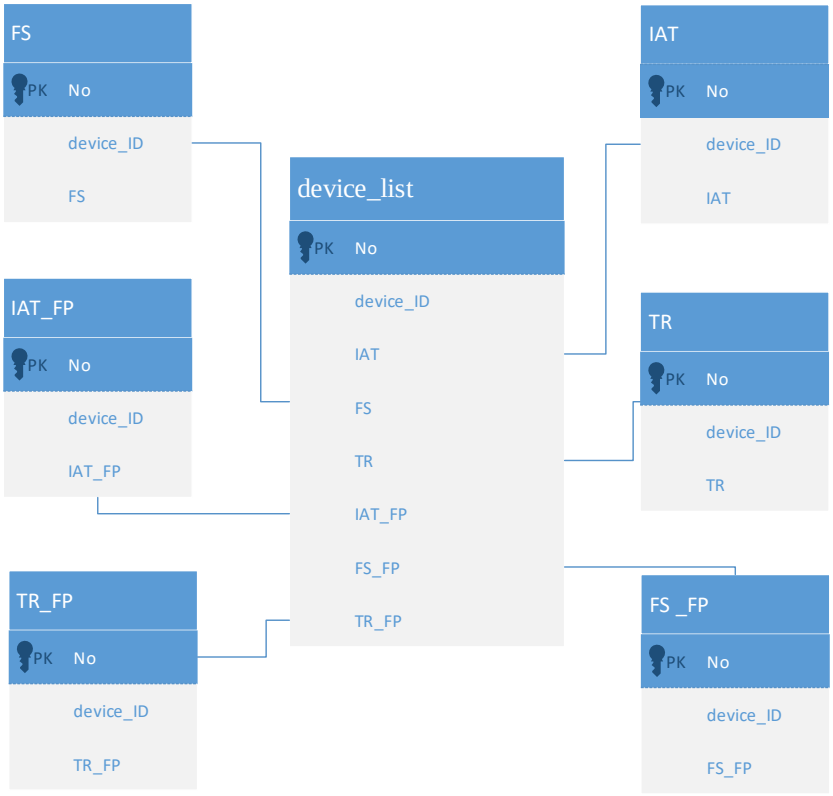

图 5-8　数据库表之间的关系

2.　模板（Tamplate）

模板的作用是在服务器端把变量嵌入到 HTML 中渲染后，返回给浏览器来达到前后端代码分离，页面动态显示的目的。模板层提供了设计友好的语法来展示信息给用户。使用模板方法可以动态地生成 HTML。Django 自带的模板语言包含 HTML 代码和逻辑控制代码，其语法主要分为两种：{{变量}}和{% Tag %}，{{变量}}主要用于和视图变量做替换，{% tag %}主要用于做逻辑判断和实现某些功能，正因有了数据+逻辑才构成了模板语言。

模板系统中，除 HTML 外，还包含图片、CSS 文件、字体文件、JS 文件等静态文件（static files）。CSS 是层叠样式表（Cascading Style Sheets），它定义了如何显示 HTML 元素。JS（JavaScript）是一种轻量级的编程语言，JS 与 CSS 配合可给页面增加动态效果，此外，JS 有时也用于实现简单的业务逻辑。Django 中是以 URL 的形式访问这些静态文件的，因此需要像配置 URL 一样配置静态文件的访问路径。

3.　视图（View）

视图用于接收 Web 请求并返回 Web 响应，一个视图就是一个 Python 函数。当服务器接收到用户的请求后，会根据请求的内容创建 HTTPRequest 对象，这个 HTTPRequest 对象即为视图函数的输入。处理完相应的请求后，视图函数通过创建一个 HTTPResponse 对象返回生成的响应，前端根据此相应重新渲染页面。每个视图函数都必须返回一个 HTTPResponse 对象。

前端和后端的数据交互基于 JSON 格式，使用 Python 自带的函数可随意进行 JSON

格式数据的解析与转换。

除上述三层外，Django 还需配置 URLconf，这就是系统的网站目录，当用户访问某个 URL 时，系统根据 URLconf 中的配置调用相应的视图函数。原型系统共有 4 个 URL：capture、extractFP、recognize 和 update，分别对应系统的 4 个功能模块，相应地，系统有 4 个视图函数，与每个 URL 一一对应。

5.4 原型系统功能测试

本节将对本文实现的无线设备指纹识别系统进行功能测试。针对原型系统中的每一个功能模块，分别描述其操作流程和相应的结果。

5.4.1 流量数据捕获模块功能测试

进入原型系统后会默认进入流量采集界面。流量数据捕获功能又分为两个子功能模块：数据采集和数据详情显示，具体的页面设计如图 5-9。

数据采集模块显示接入网络中的设备 IP，用户可以选择其中的一个或者多个捕获流量。点击 IP 后的“开始”按钮，即开始采集该设备数据，此时“开始”按钮会变为红色，且为不可点击状态。正在采集的设备其状态会变为“采集中”，未开始或已结束的设备状态均为“无”。点击“停止”按钮即可停止采集，完成采集后系统后台会对每一个设备生成一个文件夹，该文件夹存放该设备的各种数据文件，名为设备的 IP 地址。完成采集后会将设备的流量数据文件已 pcap 的格式保存，放在该文件夹下。

采集结束后可查看数据详情，数据详情会显示该设备采集的开始时间、持续时间（单位：分钟）和数据量。点击“数据详情”旁边的“刷新”按钮，会重新从后台加载数据并显示。

图 5-9　流量数据捕获页面设计

5.4.2 特征指纹构建模块功能测试

点击左侧菜单栏的“特征指纹提取”可进入特征指纹提取页面，如图 5-10。特征指纹提取页面主要被划分为四个模块：特征指纹提取、查看参数、查看降噪结果和查看归一化结果，点击左侧菜单栏中的按钮即可跳转到相应位置。

特征指纹提取子模块会显示系统中已有的数据文件，用户可以选中其中的一个或多个同时提取特征。窗大小和分组大小这两个参数可由用户选择，分组大小的范围为 100 至 1000，步长为 100，默认值为 100；窗大小范围为 10 到 50，步长为 10，默认值为 10。选完参数和文件后，点击下方的“一键提取特征”即会完成数据预处理和特征提取过程，过程中生成的参数文件（data.txt）、降噪数据文件（denoised.txt）、归一化数据文件（normalized.txt）和四个特征文件（IAT_feature.txt、FrameSize_feature.txt、TransRate_feature.txt 和 mix_feature.txt）均存放在相应的设备文件夹下。

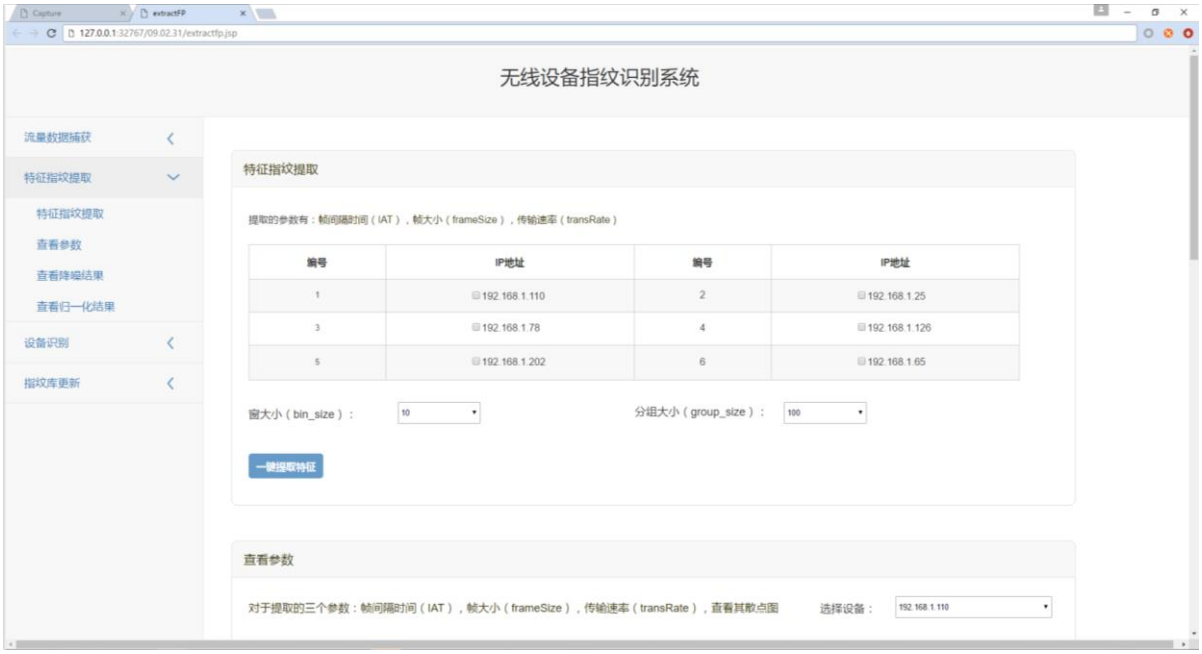

图 5-10　特征指纹提取页面（1）

查看参数子模块显示从流量数据文件中提取的各项参数的散点图。如图 5-11 中三幅图从左到右依次显示的是 IAT、FrameSize 和 TransRate 三项参数的散点图。

查看降噪结果子模块显示各项参数的降噪结果，仍以散点图的形式呈现。图 5-12 中从左到右依次为 IAT、FrameSize 和 TransRate 三项参数的降噪结果散点图。

查看归一化结果子模块显示各项参数的归一化结果，仍以散点图的形式呈现。图 5-13 中从左到右依次为 IAT、FrameSize 和 TransRate 三项参数的归一化结果散点图。

图 5-11　特征指纹构建页面（2）

图 5-12　特征指纹构建页面（3）

图 5-13　特征指纹构建页面（4）

5.4.3 无线设备识别模块功能测试

点击左侧菜单栏的“设备识别”按钮，会跳转到设备识别页面，该页面如图 5-14 所示。页面上主要包含两个子模块：参数选择和查看结果。

图 5-14　设备识别页面（1）

选择参数子模块的主要功能是让用户选择设备识别所需参数，包括：待测设备、使用的分类器和特征指纹。待测设备是一个下拉的单选框，显示设备的 IP 地址。可供选择的分类器有：随机森林（RF）、支持向量机（SVM）、朴素贝叶斯（NBC）和 K 最近邻（KNN），用户可选择其中的一个或多个。可供选择的特征指纹有三个基于概率密度

的特征指纹：帧时间间隔（IAT）、帧大小（FrameSize）、传输速率（TransRate），还有基于特征融合的特征指纹。特征指纹的选择同样是复选框的形式，用户可选择其中的一个或多个进行设备识别。每一项参数的选择都是必要的，用户选择完各项参数后，点击下方的“确定”按钮即可。

查看结果子模块显示识别结果。如果指纹库中没有与该设备指纹匹配的设备，则显示结果为未知设备。如果在指纹库中找到了与待测设备匹配的指纹，则会显示该设备指纹的 ID，以及两个设备的匹配结果。匹配结果以表格的形式展示，针对每一个分类器生成一个表格，表格中包含使用的特征指纹，以及针对每项特征指纹计算出的准确率、召回率和 F1。表格的最后一列显示各项指标的平均值。若 F1 的平均值小于 0.6，则该设备认为该设备是指纹库中已有的设备，系统返回相应的设备 ID 根据分类器的识别结果生成表格；若 F1 均值大于 0.6，则认为该设备为未知设备。

图 5-15 展示了使用随机森林对某个设备识别的结果。表格的 2 至 4 列表示使用的特征指纹，2 至 4 行分别为基于该特征指纹得到的准确率、召回率和 F1 值，表格的最后一列为各评估指标的平均值。由表格的最后一列可以看到，三种评估指标的均值分别为 0.5113、0.5464、0.5086，所以判定该设备为指纹库中的已有设备。

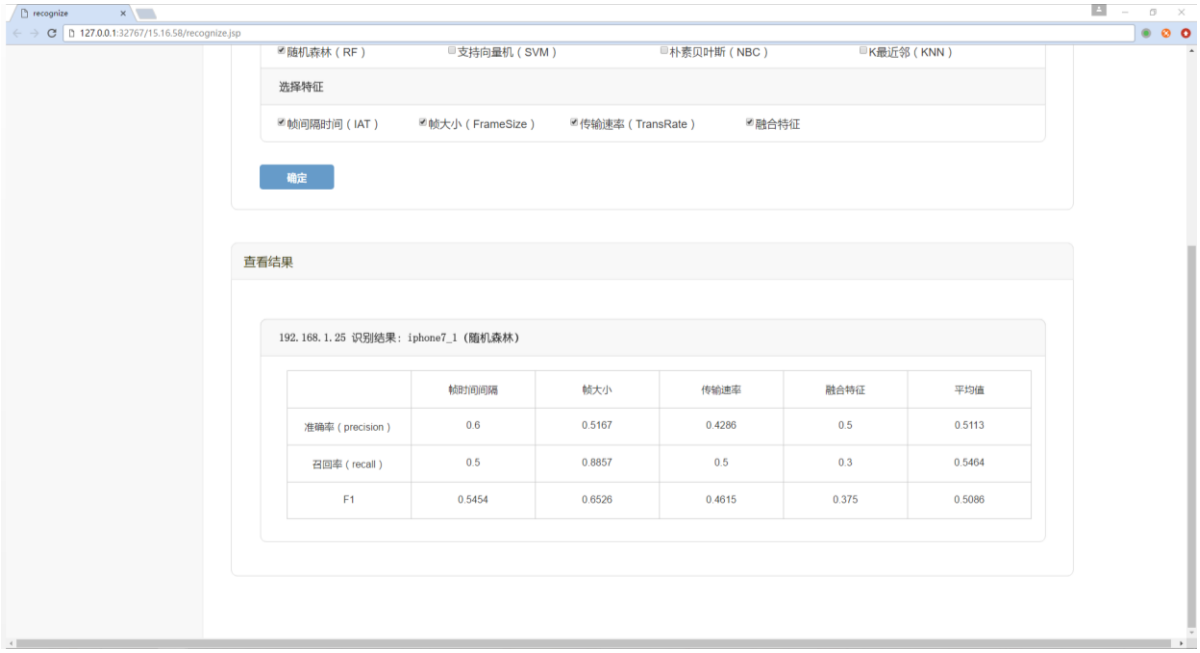

图 5-15　设备识别页面（2）

5.4.4 指纹库更新模块功能测试

点击左侧菜单栏中“指纹库更新”按钮，跳转到指纹库更新页面，如图 5-16 所示。该页面中根据设备识别结果显示每个设备的状态：未知设备或已知设备。若为未知设备，用户需自己手动为设备添加名称，系统会自动在设备名称后添加一个数字，构成设备 ID，对于已有设备则无需做任何更改。

用户点击最下方的“一键更新”按钮，系统根据 5.3.5 中设计的数据库为每个设备生成数据表，完成指纹库的更新。

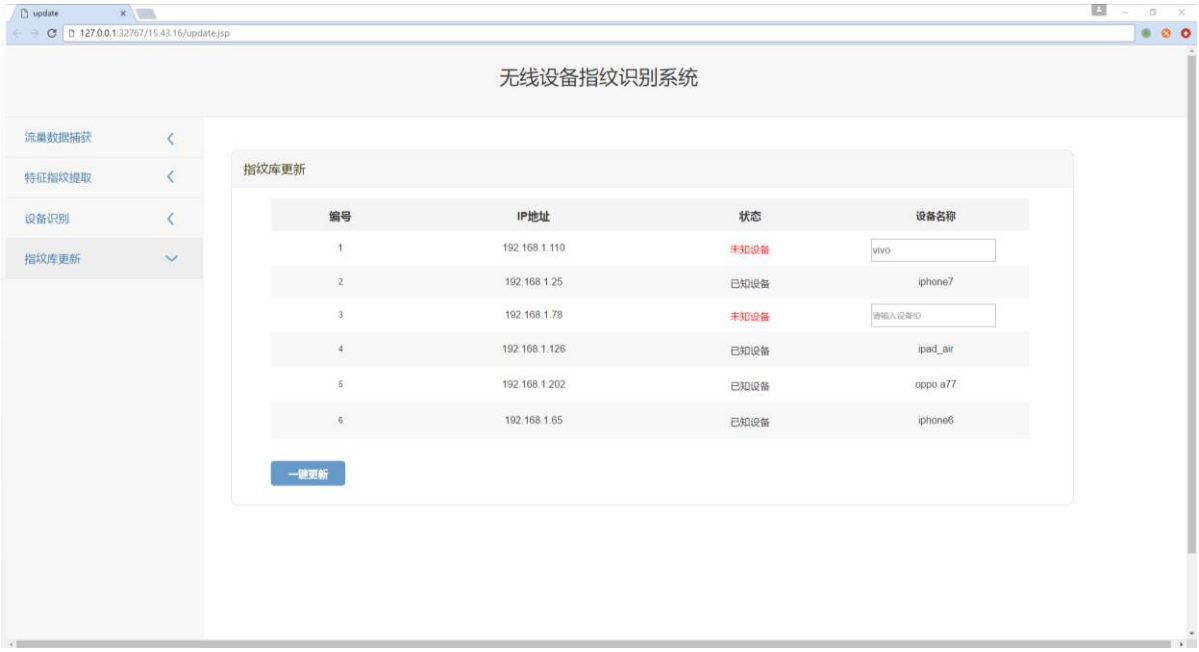

图 5-16　指纹库更新页面

5.5 本章小结

本章首先分析了无线设备指纹识别原型系统的功能需求。紧接着设计系统的框架结构，将系统分为四个功能模块：流量数据捕获模块、特征指纹形成模块、无线设备识别模块和指纹库更新模块。然后阐述了原型系统的实现细节，包括每个模块的工作步骤和前后端交互流程；Web 服务器端程序基于 MTV 架构实现，并分别从这三方面进行介绍。最后对整个系统进行功能测试，验证系统实现的正确性。

6 结论与展望

6.1 论文工作总结

本文提出了一种基于流量认知的无线设备指纹识别技术。在传统的设备身份认证信息容易被篡改和伪装的前提下，本文基于对无线网络中移动设备发出流量数据帧的统计分析，从无线数据帧与设备个体的相关性角度对设备的身份进行认证识别。这个思路不仅适用于无线网络移动设备的识别，也可以被用于有线网络中设备、路由器、交换机等网络个体的识别。对于网络安全防御和网络用户隐私信息的安全保护起到很大的积极作用。本文所用的方法克服了主动式识别需要被识别设备参与、回复消息或者安装第三方软件的弊端，在无需被识别设备的参与下就能完成对设备的识别工作。

本文的工作总结如下：

1）创建了设备指纹识别的数据集，该数据集中包含个人 PC、智能手机、平板电脑在内的 23 台无线设备的流量数据，每台设备的数据帧数目都达到了 50 万条以上。此数据集不仅是本文的研究基础，也将作为一个公开的数据集，供国内外的同行研究使用。

2）在分析无线设备的网络流量数据与设备硬件之间关系的基础上，从 TCP 流量数据帧中提取出三种参数：帧时间间隔（IAT）、帧大小（FrameSize）和传输速率（TransRate），用于特征指纹的构建。需要指出的是，这三种参数的数据并不能直接用于构建设备指纹，还需要经过降噪和归一化处理。

3）本文提出两种特征指纹生成的方法：基于概率密度的特征指纹和基于多特征融合的特征指纹。基于概率密度的特征指纹是统计每种参数的概率密度作为特征，三个参数分别形成三类特征指纹。多特征融合的特征指纹是将三个独立的特征指纹组合在一起，以形成能够更加完整表征设备身份信息的特征指纹。

4）针对无线设备识别问题，本文使用随机森林（RF）、支持向量机（SVM）、K 最近邻（KNN）、朴素贝叶斯（NBC）四种分类器进行设备身份模型的构建与评估，并采用准确率（precision）、召回率（recall）和 F1 值来评估识别的效果。实验的结果令人鼓舞：当使用随机森林分类器时，precision、recall 和 F1 值分别为 0.9930、0.976 和 0.9783。

5）本文还探讨了特征空间的变化对识别效果的影响，即更改特征指纹生成过程中的部分参数，比较各分类器的效果和两种特征指纹的性能。实验结果表明，随机森林的分类效果比较稳定，始终能保持较好的识别结果；相比于基于概率密度的特征指纹，融合特征指纹的鲁棒性更好，当特征指纹生成过程中的参数变化时，融合特征仍能较好地标识设备的身份。

6）本文设计并实现了基于网络流量认知的无线设备识别原型系统，该原型系统集成了无线设备指纹识别方案中的各个模块，可在线捕获移动设备的网络流量并对其进行身份识别。原型系统的实现使移动设备的身份识别从学术研究的角度向工程实用的

角度逐步过渡。

6.2 不足与展望

本文提出的基于网络流量的无线设备指纹识别技术存在一些局限和不足，许多问题仍有待进一步的研究和完善。

限于实验环境和经济实力，我们实验中的数据规模较小，仅包含 23 台设备的数据，这对于无线设备的识别是远远不够的，因此创建更大规模的指纹库来测试算法的性能是完全必要的。此外，我们对每台设备的数据采集是一次性完成的，由于设备存在器件老化问题，长时间的积累会引起器件参数的退化老化效应[2]，设备指纹是否能够保持较长时间内的稳定性仍是未知之数。更大规模的指纹库不仅应当包含更多设备的指纹，还应当包含一台设备在较长时间间隔内的指纹。

当前基于流量的无线设备指纹相关研究缺乏一个标准的数据库，我们考虑将采集到的原始的设备流量数据作为公开的数据集，以便更多的研究者研究无线设备指纹识别问题，同时也为我们将来提取更丰富的特征提供基础。

本文从网络流量中提取帧时间间隔、帧大小和传输速率三个参数，网络流量中也许存在其它能够反映设备身份特征的信息。我们拟在后期的研究中从流量中提取更加丰富的特征，并采用类内离散度、类间离散度和 PCA 主成分分析等方法对各项特征展开分析，以期达到更好的识别效果。

本文所使用的分类器均为有监督学习的分类器，即输入的样本标签均已知。但在实际情况中新获取的数据样本标签通常是不可知的，这种情况下可以考虑结合非监督分类方法，先利用多个模型对未标记样本进行预测，再根据标记样本进行数据帧指纹识别模型的更新。一般来说，非监督学习方法的分类准确率会低于监督学习，为了克服这个缺点，我们拟采用多分类器联合决策的方案，通过运用机器学习和信息融合的方法，发挥各种分类器的优势，提高分类准确率。

致　谢

“读书不觉已春深，一寸光阴一寸金”。在近七载寸金光阴中，我在交大完成了本科学业，并且即将结束我的硕士科研生涯。在我硕士求学道路中，得到了老师、同学以及家人的诸多帮助和鼓励，使我能够锲而不舍地在研究领域学习和进步，最终取得一定成果。在此，我要向他们表示由衷谢意：

首先衷心感谢我的导师沈超老师，沈老师在我读研期间不仅关心我们科研学习，还在生活上给与无微不至的关照。在我毕业论文的选题过程中，沈老师给出了指导性的推荐和意见，论文的撰写和修改也是在他的悉心指导下完成的。沈老师对科研巨大的热情使我深受鼓舞，他对待学术严谨的态度更是让我钦佩不已。细算起来我承蒙恩师教诲已近四年，四年来沈老师对我的教诲不仅止于科研，沈老师对我失意时的鼓励和懈怠时的劝勉我将永感于心。

感谢系统工程研究所的各位老师和同学。蔡忠敏老师和刘烃老师在工作和生活方面的讨论使我获益良多。感谢实验室的孙鸿师兄、季建廷师兄，他们在我科研和生活中的无私帮助一直温暖着我。还要感谢实验室的陈宇飞师弟、刘鹏飞师弟、刘莹师妹、贾占培师弟、王诏师弟、陶静师妹，因为有了他们的存在，给枯燥的科研注入一股清冽的源泉，科研道路上因此趣味丛生。

感谢我的舍友朱思敏、彭星宇和寇倩，我们从大一持续到研三的友情对我而无比珍贵。与你们的相遇是我大学生活最美好的回忆，很开心能与你们一起生活七年，七年里我们有太多难忘的时光。虽然即将各奔东西，但是海内存知己，天涯若比邻，我们的友情经过时光的发酵也会变得更加醇厚。

我的父母和男友罗登是我在困境中最大的精神支柱，感谢你们在我多年求学道路上的关怀和支持，你们永远是我不懈奋斗的源动力。是你们无私的奉献和付出帮我度过人生中的风风雨雨，让我对未来充满信心和希望。

感谢西安交通大学，七年交大求学路，终生凭风砥砺行。良好的学习生活环境让我的大学生活充实而又有活力，严谨的治学理念和良好的学习氛围让我没有辜负人生中最该奋斗的时光。

最后还要感谢为我提供奖学金的胡保生教授和广东建北逻辑高科技有限公司，感谢他们在我求学生涯中提供的帮助。特别是胡保生教授，他已年逾古稀，本人生活朴素却仍然坚持每年为学生提供奖学金，他的无私奉献精神值得我们永远学习。

参考文献

[1] 第41次《中国互联网络发展状况统计报告》发布[J]. 中国广播, 2018(3).

[2] 俞佳宝, 胡爱群, 朱长明,等. 无线通信设备的射频指纹提取与识别方法[J]. 密码学报, 2016, 3(5).

[3] 袁红林. 基于射频指纹的无线网络物理层认证关键技术研究[D]. 东南大学, 2011.

[4] Chaabouni R. Break WEP Faster with Statistical Analysis[J]. Epfl, 2006.

[5] Mavridis I P, Androulakis A I E, Halkias A B, et al. Real-Life Paradigms of Wireless Network Security Attacks[C]// Informatics. IEEE, 2011:112-116.

[6] Ross A, Jain A. Information fusion in biometrics[J]. Pattern Recognition Letters, 2003, 24(13):2115-2125.

[7] Tuyls P, Goseling J. Capacity and Examples of Template-Protecting Biometric Authentication Systems[C]// Biometric Authentication, ECCV 2004 International Workshop, BioAW 2004, Prague, Czech Republic, May 15, 2004, Proceedings. DBLP, 2004:158-170.

[8] Talbot K I, Duley P R, Hyatt M H. Specific emitter identification and verification[J]. Technology Review, 2003.

[9] Riezenman M J. Cellular security: better, but foes still lurk[J]. Spectrum IEEE, 2000, 37(6):39-42.

[10] Langley L E. Specific emitter identification (SEI) and classical parameter fusion technology[C]// Wescon/'93. Conference Record. IEEE, 2002:377-381.

[11] 张旭. 基于信号分析的无线设备“指纹”特征提取[D]. 北京邮电大学, 2015.

[12] Guo F, Chiueh T. Sequence Number-Based MAC Address Spoof Detection[J]. Lecture Notes in Computer Science, 2005, 3858:309-329.

[13] 顾杨. 基于无线设备特征指纹的无线钓鱼接入点检测技术研究[D]. 南京邮电大学, 2014.

[14] Desmond L C C, Yuan C C, Tan C P, et al. Identifying unique devices through wireless fingerprinting[C]// ACM Conference on Wireless Network Security, WISEC 2008, Alexandria, Va, Usa, March 31 - April. DBLP, 2008:46-55.

[15] Pang J, Greenstein B, Gummadi R, et al. 802.11 user fingerprinting[C]// International Conference on Mobile Computing and Networking, MOBICOM 2007, Montréal, Québec, Canada, September. DBLP, 2007:99-110.

[16] Sieka B. Active fingerprinting of 802.11 devices by timing analysis[C]// Ccnc 2006. 2006, IEEE Consumer Communications and NETWORKING Conference. IEEE, 2006:15-19.

[17] Gao K, Corbett C, Beyah R. A passive approach to wireless device fingerprinting[C]// Ieee/ifip International Conference on Dependable Systems and Networks. IEEE, 2010:383-392. Yen T F, Xie Y, Yu F, et al. Host Fingerprinting and Tracking on the Web:Privacy and Security Implications[J]. 2012, 11(1):111 - 124.

[18] Lyon G. Nmap: a free network mapping and security scanning tool[EB/OL]. [11-5]. https://nmap.org/.

[19] Yarochkin F, Kydyraliev M, Arkin O. Xprobe project[EB/OL]. http://ofirarkin.wordpress.com/xprobe/.

[20] Eckersley P. How Unique Is Your Web Browser? Privacy Enhancing Technologies, International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings, 2010[C].

[21] Yen T F, Xie Y, Yu F, et al. Host Fingerprinting and Tracking on the Web:Privacy and Security Implications[J]. 2012, 11(1):111 - 124.

[22] Mowery K, Bogenreif D, Yilek S, et al. Fingerprinting Information in JavaScript Implementations[C]// Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE

International Conference on. IEEE, 2011:9-12.
[23] Acar G, Juarez M, Nikiforakis N, et al. FPDetective:dusting the web for fingerprinters[C]// ACM Sigsac Conference on Computer & Communications Security. ACM, 2013:1129-1140.
[24] Łukasz Olejnik, Castelluccia C, Janc A. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns[J]. Hot Topics in Privacy Enhancing Technologies, 2012, 69(1-2):63-74.
[25] Bratus S, Cornelius C, Kotz D, et al. Active behavioral fingerprinting of wireless devices[C]// ACM Conference on Wireless Network Security. ACM, 2008:56-61.
[26] Radhakrishnan S V, Uluagac A S, Beyah R. GTID: A Technique for Physical Device and Device Type Fingerprinting[J]. IEEE Transactions on Dependable & Secure Computing, 2015, 12(5):519-532.
[27] Khlifi H, Gregoire J C. Estimation and removal of clock skew from delay measures[C]// IEEE International Conference on Local Computer Networks. IEEE, 2004:144-151.
[28] Kohno T, Broido A, Claffy K C. Remote Physical Device Fingerprinting[C]// Security and Privacy, 2005 IEEE Symposium on. IEEE, 2005:211-225.
[29] Cristea M, Groza B. Fingerprinting Smartphones Remotely via ICMP Timestamps[J]. IEEE Communications Letters, 2013, 17(6):1081-1083.
[30] Langley L E. Specific emitter identification (SEI) and classical parameter fusion technology[C]// Wescon/'93. Conference Record. IEEE, 2002:377-381.
[31] Lanze F, Panchenko A, Braatz B, et al. Clock skew based remote device fingerprinting demystified[C]// Global Communications Conference. IEEE, 2012:813-819.
[32] Neumann C, Heen O, Onno S. An Empirical Study of Passive 802.11 Device Fingerprinting[J]. 2014:593-602.
[33] Franklin J, Mccoy D, Tabriz P, et al. Passive data link layer 802.11 wireless device driver fingerprinting[C]// Conference on Usenix Security Symposium. USENIX Association, 2006:167--178.
[34] Gerdes R M, Daniels T, Mina M, et al. Device Identification via Analog Signal Fingerprinting: A Matched Filter Approach.[C]// 144 Proceedings of the Network and Distributed System Security Symposium. 2004:78.
[35] Brik V, Banerjee S, Gruteser M, et al. Wireless device identification with radiometric signatures[C]// ACM International Conference on Mobile Computing and NETWORKING. ACM, 2008:116-127.
[36] Nguyen N T, Zheng G, Han Z, et al. Device fingerprinting to enhance wireless security using nonparametric Bayesian method[C]// INFOCOM, 2011 Proceedings IEEE. IEEE, 2011:1404-1412.
[37] Li Z, Xu W, Miller R, et al. Securing wireless systems via lower layer enforcements[C]// ACM Workshop on Wireless Security. ACM, 2006:33-42.
[38] Shi Y, Jensen M A. Improved Radiometric Identification of Wireless Devices Using MIMO Transmission[J]. IEEE Transactions on Information Forensics & Security, 2011, 6(4):1346-1354.
[39] Das A, Borisov N, Caesar M. Do You Hear What I Hear?: Fingerprinting Smart Devices Through Embedded Acoustic Components[C]// ACM Sigsac Conference on Computer and Communications Security. ACM, 2014:441-452.
[40] Dey S, Roy N, Xu W, et al. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable[M]. 2014.
[41] Breiman L I, Friedman J H, Olshen R A, et al. Classification and Regression Trees (CART)[J]. Encyclopedia of Ecology, 1984, 40(3):582-588.
[42] 陈峰. 基于 CART 算法的空气质量指数回归预测模型的学习[J]. 上饶师范学院学报, 2016, 36(6):16-21.
[43] Breiman L. Random Forests[J]. Machine Learning, 2001, 45(1):5-32.

[44] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20(3):273-297.
[45] Hsu C W. A practical guide to support vector classification[J]. 2003, 67(5).
[46] 周志华. 《机器学习》[J]. 中国民商, 2016(3).
[47] Cover T M, Hart P E. Nearest neighbor pattern classification[J]. IEEE Trans.inf.theory, 1967, 13(1):21-27.
[48] J H D, Yu K. Idiot's Bayes—Not So Stupid After All?[J]. International Statistical Review, 2001, 69(3):385-398.
[49] 赵敏. 聚类分析与朴素贝叶斯分类在客户价值预测中的应用研究[D]. 合肥工业大学, 2010.
[50] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]// International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1995:1137-1143.

攻读学位期间取得的研究成果

期刊论文：

[1] Shen C, Yu T, Xu H, et al. User practice in password security: an empirical study of real-life passwords in the wild[J]. Computers & Security, 2016, 61:130-141.

[2] Shen C, Yu T, Yuan S, et al. Performance Analysis of Motion-Sensor Behavior for User Authentication on Smartphones[J]. Sensors, 2016, 16(3):345.

会议论文：

[1] Shen C, Li Y, Yu T, et al. Motion-Senor Behavior Analysis for Continuous Authentication on Smartphones[J]. Proceedings of the World Congress on Intelligent Control & Automation.

[2] Shen C, Zhang Y, Cai Z, et al. Touch-interaction behavior for continuous user authentication on smartphones[C]// International Conference on Biometrics. IEEE, 2015:157-162.

[3] Shen C, Pei S, Yu T, et al. On motion sensors as source for user input inference in smartphones[C]// IEEE International Conference on Identity, Security and Behavior Analysis. IEEE, 2015:1-6.

学位论文独创性声明（1）

本人声明：所呈交的学位论文系在导师指导下本人独立完成的研究成果。文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，也不包含本人已用于其他学位申请的论文或成果。

本人如违反上述声明，愿意承担以下责任和后果：

1．交回学校授予的学位证书；

2．学校可在相关媒体上对作者本人的行为进行通报；

3．本人按照学校规定的方式，对因不当取得学位给学校造成的名誉损害，进行公开道歉。

4．本人负责因论文成果不实产生的法律纠纷。

论文作者（签名）：　　日期：　　年　　月　　日

学位论文独创性声明（2）

本人声明：研究生_____ 所提交的本篇学位论文已经本人审阅，确系在本人指导下由该生独立完成的研究成果。

本人如违反上述声明，愿意承担以下责任和后果：

1．学校可在相关媒体上对本人的失察行为进行通报；

2．本人按照学校规定的方式，对因失察给学校造成的名誉损害，进行公开道歉。

3．本人接受学校按照有关规定做出的任何处理。

指导教师（签名）：　　日期：　　年　　月　　日

学位论文知识产权权属声明

我们声明，我们提交的学位论文及相关的职务作品，知识产权归属学校。学校享有以任何方式发表、复制、公开阅览、借阅以及申请专利等权利。学位论文作者离校后，或学位论文导师因故离校后，发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为西安交通大学。

论文作者（签名）：　　日期：　　年　　月　　日

指导教师（签名）：　　日期：　　年　　月　　日