中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

中国科学院大学学位论文 LaTeX 模板 $\pi\pi\pi$

作者姓名：莫晃锐

指导教师：刘青泉 研究员 中国科学院力学研究所

学位类别：理学硕士

学科专业：流体力学

培养单位：中国科学院力学研究所

2014 年 6 月

LaTeX Thesis Template

of

The University of Chinese Academy of Sciences $^{\pi}\pi^{\pi}$

A thesis submitted to the

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Natural Science

in Fluid Mechanics

By

Mo Huangrui

Supervisor: Professor Liu Qingquan

Institute of Mechanics, Chinese Academy of Sciences

June, 2014

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。本人完全意识到本声明的法律结果由本人承担。

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大学有关保存和使用学位论文的规定，即中国科学院大学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

本文是中国科学院大学学位论文模板 ucasthesis 的使用说明文档。主要内容为介绍 LaTeX 文档类 ucasthesis 的用法，以及如何使用 LaTeX 快速高效地撰写学位论文。

关键词：中国科学院大学，学位论文，LaTeX 模板

Abstract

This paper is a help documentation for the LaTeX class ucasthesis, which is a thesis template for the University of Chinese Academy of Sciences. The main content is about how to use the ucasthesis, as well as how to write thesis efficiently by using LaTeX.

Keywords: University of Chinese Academy of Sciences (UCAS), Thesis, LaTeX Template

目 录

图形列表

表格列表

符号列表

字符

Symbol	Description	Unit
R	the gas constant	$m^2 \cdot s^{-2} \cdot K^{-1}$
C_v	specific heat capacity at constant volume	$m^2 \cdot s^{-2} \cdot K^{-1}$
C_p	specific heat capacity at constant pressure	$m^2 \cdot s^{-2} \cdot K^{-1}$
E	specific total energy	$m^2 \cdot s^{-2}$
e	specific internal energy	$m^2 \cdot s^{-2}$
h_T	specific total enthalpy	$m^2 \cdot s^{-2}$
h	specific enthalpy	$m^2 \cdot s^{-2}$
k	thermal conductivity	$kg \cdot m \cdot s^{-3} \cdot K^{-1}$
S_{ij}	deviatoric stress tensor	$kg \cdot m^{-1} \cdot s^{-2}$
τ_{ij}	viscous stress tensor	$kg \cdot m^{-1} \cdot s^{-2}$
δ_{ij}	Kronecker tensor	1
I_{ij}	identity tensor	1

算子

Symbol	Description
Δ	difference
∇	gradient operator
$\delta^{\pm}$	upwind-biased interpolation scheme

缩写

CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
EOS	Equation of State

JWL	Jones-Wilkins-Lee
WENO	Weighted Essentially Non-oscillatory
ZND	Zel'dovich-von Neumann-Doering

第 1 章　引言

1.1　研究背景

进入 21 世纪以来，网络与信息技术的发展日新月异，快速改变着人们的工作和生活方式。越来越多的政府、企业建立了依赖于网络的业务信息系统，比如电子商务、网上银行、电子政务等，对社会的各行各业造成了深远的影响。与此同时，网络安全的重要性也在不断提升。无论是即时通信技术中的隐私保护，还是网上银行和电子商务中的财产安全，乃至信息时代中的国家安全，都离不开网络安全技术的保障。根据 Gartner 统计数据，2018 年全球网络安全产业规模达到 1119.88 亿美元，预计 2019 年增长至 1216.68 亿美元。从增速上看，2018 年全球网络安全产业增速为 11.3%，创下自 2016 年以来的新高。2014-2019 年全球网络安全产业规模及增速如图1.1所示。

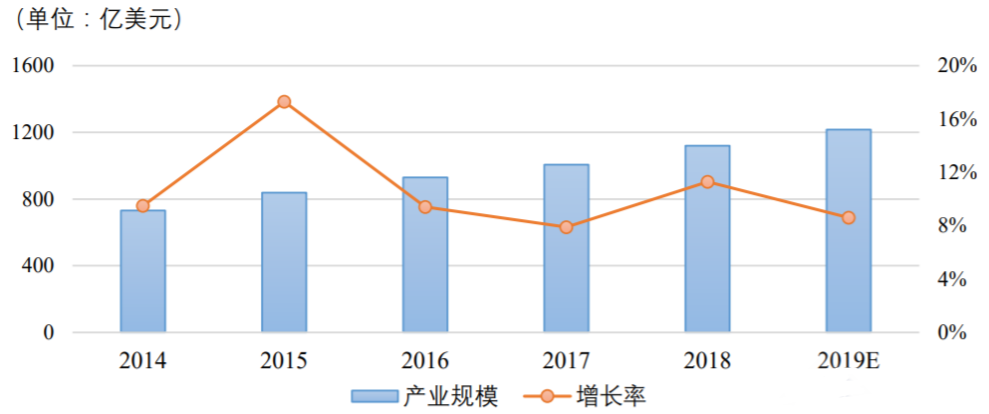


图 1.1 2014-2019 年全球网络安全产业规模及增速

Figure 1.1 Global cyber security industry scale and growth rate, 2014-2019

随着网络攻击行为日趋复杂，政府、企业以及个人所面临的安全威胁正在飞速增长，如蠕虫病毒、木马后门、僵尸网络、DDOS 攻击等，给企业的信息网络造成严重破坏。其中，多数网络攻击的发起都是基于不同主机属性如各类操作系统和浏览器的软件漏洞。根据中国互联网网络安全报告显示，国家信息安全漏洞共享平台（CNVD）在 2018 年共收录通用软硬件漏洞 14201 个。根据影响对象的类型，漏洞可分为：应用程序漏洞，Web 应用漏洞，操作系统漏洞，网络设备漏洞，安全产品漏洞和数据库漏洞。如图1.2所示，在 2018 年 CNVD 收录的漏洞信息中，操作系统漏洞占 10.6%，Web 应用漏洞占 18.7%。

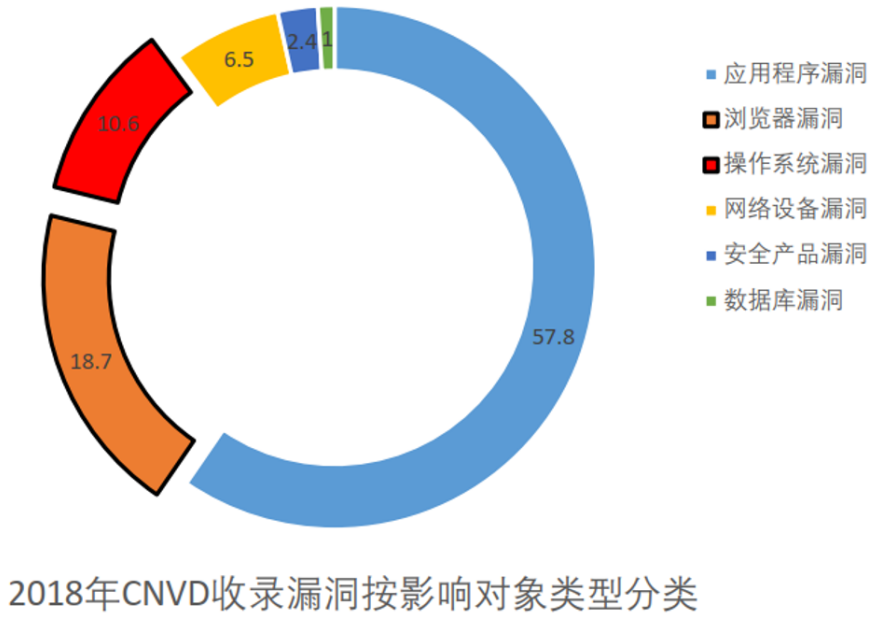


图 1.2 2018 年 CNVD 收录的漏洞按影响对象类型分类统计

Figure 1.2 The vulnerabilities collected by CNVD in 2018 are classified according to the types of affected objects.

北京时间 2017 年 05 月 12 日，多家安全机构监测到黑客利用 NSA 黑客武器库泄漏的“永恒之蓝”工具发起的网络攻击事件：大量服务器和个人 PC 感染病毒后被远程控制，成为不法分子的比特币挖矿机（挖矿会耗费大量计算资源，导致机器性能降低），甚至被安装勒索软件，磁盘文件会被病毒加密为.onion 或者.WNCRY 后缀，用户只有支付高额赎金后才能解密恢复文件，对个人及企业重要文件数据造成严重损失。“永恒之蓝”工具利用的是微软 Windows 操作系统的 SMBv1 协议中的安全漏洞。未经身份验证的攻击者可以向目标机器发送特制报文触发缓冲区溢出，导致在目标机器上远程执行任意代码。“永恒之蓝”工具会扫描开放 445 文件共享端口的 Windows 机器，只要用户开机上网，黑客就可能在电脑和服务器中植入勒索软件。这次攻击袭击了上百个国家不计其数的电子设备，对全球经济发展造成了难以估量的损失。

综上所述，网络信息安全问题在我国以及全球显著的增加，严重的威胁到人民和国家的利益，而对网络中机器设备的主机属性识别技术则是网络攻防任务中的关键所在。

1.2　研究目的和意义

保障网络安全最大的挑战之一就是及时发现漏洞，而绝大部分安全漏洞和隐患都与主机属性息息相关。此外，识别网络中主机的多维属性还可以帮助网络运营者有效地进行网络管理、网络资源分配、网络服务质量优化。

1.3　本文的研究内容与主要贡献

本文拟研究复杂网络中的细粒度主机属性发现技术，基于加密流原始载荷的主机属性发现技术以及大规模网络中的海量主机属性发现技术。本文的主要贡献包括以下三个方面：

1. **复杂网络中的细粒度主机属性发现。**首先以流为单位，提取目标主机发起的 TLS 会话中的 13 维协议头部字段特征和 3 类流统计特征，包括 IP 协议的跳数、包长、分片标识等字段，TCP 协议的传输窗口大小、窗口缩放因子、最大报文长度等字段，TLS 协议的版本、扩展长度、密钥算法套件序列等字段以及流的包长序列统计特征、时间序列统计特征、速率统计特征等。然后结合以 LightGBM 模型为代表的机器学习算法，识别目标主机的操作系统类型及版本、浏览器类型及版本。

2. **基于加密流原始载荷的主机属性发现。**随着理论的成熟和机器性能的提升，深度学习模型在流量分类领域里的表现越来越出色。通过提取 TLS 会话中 TCP 握手包和 Client-Hello 包的原始流信息，并结合以 CNN 模型和 LSTM 模型为代表的表示学习算法，便可在不需要先验知识的前提下，进一步提高复杂网路中主机属性的识别精度。

3. **大规模网络中的海量主机属性发现。**本文基于 Stacking 技术，结合以人工特征为基础的 LightGBM 模型和以原始流信息为基础的深度学习模型，构建了一个用于海量主机属性发现的原型系统。该系统主要包含四个模块。协议识别模块用于在高速网络中检测、解析并识别 TLS 流量。特征提取模块用于从原始 TLS 流量中提取所需的人工特征和原始特征。属性识别模块基于 stacking 技术，综合各学习模型的的检测结果，得到最终的识别信息。数据存储和可视化模块用于识别结果的存储和可视化展示。

1.4 论文组织结构

本文共分为 X 个章节，组织结构如图 1.3 所示。

第一章为引言部分，介绍了本文的研究背景、研究内容以及主要贡献。

第 2 章　复杂网络中的细粒度主机属性发现

本章提出了一种基于协议字段特征和流统计特征的主机属性识别技术，用于在复杂的动态网络中精准地识别主机的多维属性。本章首先概述该工作的背景意义、技术原理和思路流程，然后分别介绍识别技术中的关键环节，最终展示识别技术的实验效果。

2.1　引言

保障网络安全最大的挑战之一就是及时发现漏洞，而绝大部分安全漏洞和隐患都与细粒度的主机属性息息相关。此外，识别网络中主机的多维细粒度属性还可以帮助网络运营者有效地进行网络管理、网络资源分配、网络服务质量优化。探测目标主机的相关属性一般分为主动和被动两种方式，而由于入侵检测技术的成熟，主动探测技术的局限性越来越明显，本章将介绍一种被动探测技术，可以精准、快速的识别加密网络中的细粒度主机属性。

尽管 RFC 文档已经对 TCP/IP 协议栈的规范进行了统一，但是不同厂商的软件开发者在实际开发和更新操作系统、浏览器等软件的过程中，由于缺乏协商，对网络协议栈的初始化参数进行了不同的设置。这一现象造成了不同属性的主机在网络会话过程中，存在明显的协议字段参数差异和流统计数据差异，称为 TCP/IP 协议栈指纹。本章方法以 TCP/IP 协议栈指纹为基础，结合以 LightGBM 为代表的机器学习模型，对占比 95% 市场份额的主流操作系统和浏览器进行识别。

本章方法首先从科技网中采集标注后的 TLS 会话，以流为单位，提取每条 TLS 流的 TCP/IP 协议字段特征、TLS 协议字段特征以及流统计特征，在经过数据处理后作为机器学习模型的输入，最终得到多维主机属性。

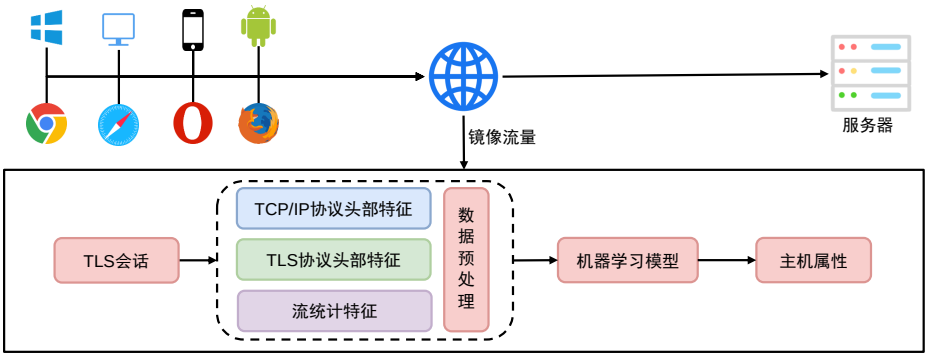


图 2.1 复杂网络中的细粒度主机属性发现

Figure 2.1 Fine-grained host attribute discovery in complex networks

2.2 特征工程

本节介绍以双向流为单位的协议字段特征、统计特征提取方法和数据预处理过程。在流量识别场景中，单向网络流通常指在一段时间内，具有相同五元组 < 源地址，目的地址，源端口，目的端口，传输协议类型 > 的所有数据包形成的序列。双向网络流由同一时间段内源目地址可互换的单向流组成。

2.2.1 协议字段特征

TCP/IP 协议栈采用五层网络模型结构，自下向上分别为：物理层，数据链路层、网络层、传输层和应用层。在一次完整的 TLS 会话中，网络层的主要协议为 IP 协议，传输层的主要协议为 TCP 协议。TCP 协议是面向连接的协议，一般通过交互三个数据包完成连接的建立。本章从由客户端发往服务端的第一个数据包即 TCP SYN 包中提取 IP 协议头部字段和 TCP 协议头部字段，从 TLS Clien Hello 包中提取 TLS 协议头部字段。

IP 报文通常由头部和载荷组成，头部长度可变，通常为 20 字节，可以根据不同的需要加入各种可选项部分。IP 头部的一般格式如图 3.2 所示，其中，本章方法所提取的 IP 协议头部参数主要为生存时间，总长度以及分片标志。

- 生存时间（TTL）：IP 报文所允许通过的路由器的最大数量。每经过一个路由器，TTL 减 1，当为 0 时，路由器将该数据报丢弃。TTL 字段是由发送端初始设置一个 8bit 字段，记录了报文在网络中的存活时间，各类操作系统的 TTL 初始值不同。
- 总长度：IP 报文的总长度，是报头的长度和载荷的长度之和。

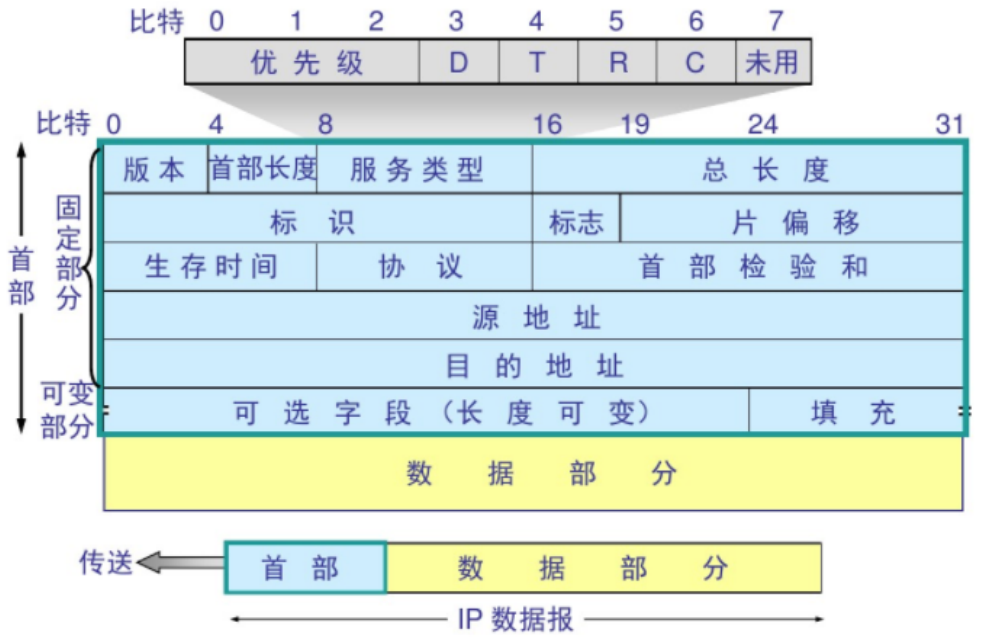


图 2.2 IP 报文格式

Figure 2.2 Fine-grained host attribute discovery in complex networks

- 分片标志：1bit 字段，1 表示报文不分片，0 表示分片。

TCP 报文同样由头部和载荷组成，头部长度一般为 20 字节，可以扩展不同的可选项部分。TCP 头部的一般格式如图 3.3 所示。其中，本章方法所提取的 TCP 协议头部参数主要为窗口大小，窗口缩放因子，最大报文长度以及可选项类型序列。

- 窗口大小：用来告知对端本地的缓存大小，以此控制对端发送数据的速率，从而达到流量控制的目的。窗口大小为一个 16bit 字段，因而最大为 65535。
- 窗口缩放因子：用于提高传输窗口的上限。
- 最大报文长度：每一个 TCP 报文段中数据字段的最大长度。其值太大或太小都不合适，太小会使得数据传输效率很低，太大会增加网络丢包的可能性，增大网络开销。
- 可选项类型序列：按照字节顺序解析 TCP SYN 包的可选项字段，从中提取出每一个可选项子字段的类型，并按照一定的编码方式将其转换为类型码，便可得到有序的类型码序列。在本章方法的编码方式中，将 NOP 类型记作 0，最大报文长度类型记作 1，窗口缩放因子类型记作 2，选择确认项类型记作 3，时间戳类型记作 4，其他类型记作 5。最终得到的类型序列一般形式为 opt_seq=o1,o2,o3,...,on,oi∈[0,7]

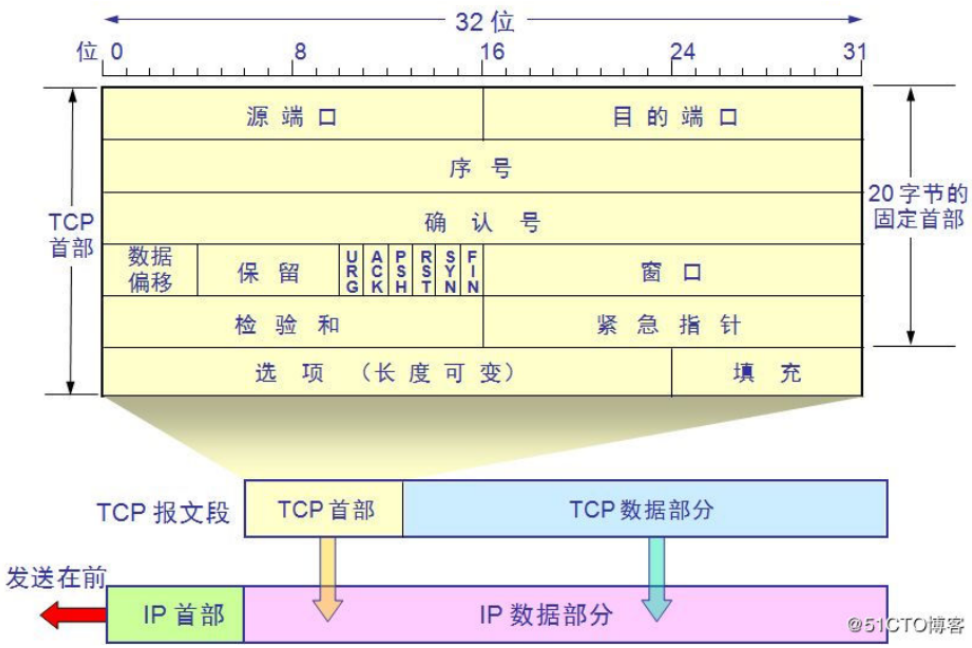


图 2.3 TCP 报文格式

Figure 2.3 Fine-grained host attribute discovery in complex networks

TLS Clien Hello 报文是 TLS 建立会话的第一个数据报文，本章方法提取的报文头部字段主要有协议版本，密钥算法套件序列，扩展长度，扩展类型序列，支持加密组件序列以及应用层协议协商状态码序列。其报文格式如图 3.4 所示。

- 协议版本：表示客户端支持的最佳 TLS 协议版本。
- 密钥算法套件序列：由客户端支持的所有密钥算法套件组成的状态码序列。加密套件列出了客户端能够支持的加密方式、算法等信息。不同的加密套件性能存在差异，同时会产生不同的 TLS 交互报文。
- 扩展长度：TLS 协议的扩展部分长度。
- 扩展类型序列：TLS 协议的扩展部分由任意数量的不同扩展组成。每一类扩展都有一个独特的类型码，按字节顺序解析扩展部分的所有扩展，即可得到扩展类型序列。
- 支持加密组件序列：按字节顺序解析，得到 Supported Groups 扩展字段中的类型码序列。
- 应用层协议协商状态码序列：按字节顺序解析，得到 Application Layer Protocol Negotiation 扩展字段中的类型码序列。

```
Secure Sockets Layer
  TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 512
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 508
      Version: TLS 1.2 (0x0303)
      Random: 00f88ee95c9a4b31adaddb7917714847dc0457e919a85e14...
      Session ID Length: 32
      Session ID: eac88fd00642f450dd84e974d916a9dceafb0fb89ce3cc76...
      Cipher Suites Length: 34
      Cipher Suites (17 suites)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 401
      Extension: Reserved (GREASE) (len=0)
      Extension: server_name (len=17)
      Extension: extended_master_secret (len=0)
      Extension: renegotiation_info (len=1)
      Extension: supported_groups (len=10)
      Extension: ec_point_formats (len=2)
      Extension: SessionTicket TLS (len=176)
      Extension: application_layer_protocol_negotiation (len=14)
```

图 2.4 TLS 报文格式

Figure 2.4 Fine-grained host attribute discovery in complex networks

2.2.2　流统计特征

流统计特征可以轮廓性地描述不同属性的主机在网络通信过程中的差异。以每条流前 50 个包的长度序列和时间间隔序列为基础，分别提取网络流在空间、时间以及速率等三个维度的统计特征。空间特征主要包括总包数、总字节数、包长均值、包长均方差、包长直方图分布和包长马尔可夫状态转移矩阵等。时间特征主要包括间隔均值、间隔均方差、间隔直方图分布和间隔马尔可夫状态转移矩阵等。速率特征主要包括包数吞吐量均值、字节数吞吐量均值以及传输峰值分布等。

包长直方图分布。假设 IP 协议的最大传输单元为 1500 字节，设置组数为 10，组距为 150 字节。根据包长序列统计前 50 个数据包的包长在每个分组中的频数，可得到长度为 10 的包长直方图分布。

间隔直方图分布。类似包长直方图分布，以毫秒为单位，将包到达时间间隔分为 10 组，组距为 100 毫秒，其中最后一组的组距为 [900,Inf)。根据包到达时间间隔序列统计其在每个分组中的频数，可得到长度为 10 的间隔直方图分布。

包长马尔科夫状态转移矩阵。以包长直方图分布为基础，对每个分组按从小

到大分配状态码 1-10，根据包长在直方图中的分布便可得到每个包对应的包长状态码。初始化一个 10*10 的全 0 矩阵，行号表示前一个数据包的包长状态，列号表示后一个数据包的包长状态，依次统计前 50 个包的包长状态转移概率，可得到包长序列的马尔科夫状态转移矩阵。

间隔马尔科夫状态转移矩阵。同样，以间隔直方图分布为基础，对每个分组按从小到大分配状态码 1-10，根据相邻数据包的时间间隔在直方图中的分布便可得到对应的间隔状态码。初始化一个 10*10 的全 0 矩阵，行号表示间隔序列中前一个间隔状态，列号表示后一个间隔状态，依次统计前 50 个包的时间间隔状态转移概率，可得到间隔马尔科夫状态转移矩阵。

2.3 机器学习模型构建

无论在主动或者被动的主机属性发现技术中，传统方法一般都是获取 TCP/IP 协议字段初始值以生成指纹，然后将获得的指纹与预先生成的已知主机属性指纹库中的指纹进行对比，当待识别指纹与已知指纹完全匹配时，便可得到具体的主机属性。传统方法属于静态方法，优点是识别速度快，缺点是无法识别不在指纹库中的未知指纹。为了克服以上缺陷，本章将构建五类经典的机器学习模型用于识别所有指纹的主机属性，提高识别方法的灵活性、准确性和鲁棒性。这五类机器学习模型分别为逻辑回归 (LR) 模型、最近邻居 (KNN) 模型、随机森林 (RF) 模型、XGBoost 模型以及 LightGBM 模型。

2.3.1 逻辑回归模型

线性回归算法是统计学和机器学习中最经典和最常用的算法之一。该算法以可解释性为代价，主要关注最小化模型误差或者尽可能作出最准确的预测。逻辑回归算法是对线性回归的改进，通常是解决二分类问题的首选方法，可将其推广到多项逻辑回归算法，以用于多分类任务。与线性回归相似，算法目标都是找到每个输入变量的权重，即系数值。然后对输出的预测使用被称为 logistic 函数的非线性函数进行变换，使得最终的预测输出值介于 0 到 1 之间。优点是计算代价小，易于理解和实现。缺点是容易出现欠拟合问题，分类精度可能不高。

表 2.1 逻辑回归模型参数选择

Table 2.1 This is a sample table.

参数	含义	取值
penalty	正则项选择	l1
solver	损失函数的优化方法	liblinear
max_iter	算法收敛的最大迭代次数	4000
multi_class	多分类方法	multinomial
random_state	随机种子	6

2.3.2　最近邻居模型

最近邻居法（KNN 算法，又译 K-近邻算法）是一种用于分类和回归的非参数统计方法。算法采用向量空间模型来分类，概念为相同类别的样本，彼此的相似度高，而可以借由计算与已知类别样本的相似度，来评估未知类别样本可能的分类，是一种基于实例的学习算法。因此，参数 K 的取值、样本之间的距离度量算法以及分类决策规则是 KNN 算法的三要素。优点是精度高，对异常值不敏感，无数据输入假定。缺点是计算复杂度和空间复杂度高。

表 2.2 最近邻居模型参数选择

Table 2.2 This is a sample table.

参数	含义	取值
n_neighbors	KNN 模型中的 k 值	7
weights	每个样本的近邻样本的权重	distance
algorithm	限定半径最近邻法使用的算法	auto
metric	距离度量算法	minkowski

2.3.3　随机森林模型

决策树是机器学习中一个基本的预测模型，表示样本类别与样本属性值之间的一种映射关系。树中每个节点表示某类样本，每个分叉路径代表某个可能的属性值，而每个叶节点则对应从根节点到该叶节点经历的路径所表示的属性值集合。随机森林作为基于 bagging 思想的集成学习算法，是一个包含多个决策树的分类模型，其输出类别由所有树输出的类别众数而定。优点是抗过拟合能力

强。

表 2.3 随机森林模型参数选择

Table 2.3 This is a sample table.

参数	含义	取值
n_estimators	决策树的数量	1500
max_features	基学习器进行切分时随机挑选的特征子集中的特征数目	auto
max_depth	树的最大深度	None
min_samples_split	节点最小分割的样本数	3
criterion	树的切分策略	gini
bootstrap	是否有放回的采样	true
random_state	随机种子	5

2.3.4 XGBoost 模型

XGBoost 模型是一种提升树集成模型，属于梯度提升树 (GBDT) 模型的范畴，基学习器一般选择 CART 回归树模型，但也可以选择其它类型的模型如逻辑回归等。GBDT 算法的基本思想是让新的基学习器去拟合前面学习器的偏差，从而不断将加法模型的偏差降低。相对比 GBDT 算法，XGBoost 将目标函数进行了二阶泰勒展开，同时加入更多的正则项以及其他改进，显著提升了模型效果和性能。优点是分类精度非常高，缺点是计算代价较大。

表 2.4 XGBoost 模型参数选择

Table 2.4 This is a sample table.

参数	含义	取值
early_stopping_rounds	早期停止迭代次数	1500
eta	学习率	0.15
max_depth	每棵树的最大深度	9
subsample	每棵树的随机样本采样比例	0.4
colsample_bytree	每棵树随机采样的特征比例	0.7
alpha	L1 正则化项的权重	0.5
seed	随机种子	5

2.3.5　LightGBM 模型

与 XGBoost 模型相同，LightGBM 模型也是一种提升树集成模型，属于梯度提升树模型的范畴。但比 XGBoost 模型更强大、速度更快。主要原因是 LightGBM 模型用 histogram 算法替换了传统的预排序方法，用带有深度限制的按叶子生长算法代替了传统的决策树生长策略，同时进行内存优化、多线程优化等处理。

表 2.5 LightGBM 模型参数选择

Table 2.5 This is a sample table.

参数	含义	取值
learning_rate	学习率	0.05
boosting_type	模型所用算法	gbdt
application	模型用途	multiclass
max_depth	每棵树的最大深度	10
num_leaves	叶子节点数	512
max_bin	bin 的最大数量	500
min_data_in_leaf	叶子节点中最小样本树	20
num_boost_round	迭代次数	500

2.4　实验设计与结果分析

2.4.1　数据集介绍

本章于 2019 年 5 月份持续五天在中国科学技术网（CSTNET）中采集了上千万条网络流数据，并从中过滤出目的端口为 80（HTTP 流）或 443（TLS 流）的所有流量。由于主机属性识别任务属于监督学习任务，需要对 TLS 流样本数据进行属性标注。首先通过 HTTP 流请求报文中的 User-Agent 字段识别客户端的主机属性信息，包括操作系统信息或浏览器信息，如图 3.5 所示。并以键值对 <client IP：OS type, OS major version> 的形式保存在数据库中。然而，由于受到 NAT 和 DHCP 等网络技术的影响，同一 client IP 可能会被标识上多个主机属性标签。因此需要将多标签的 client IP 重新标记为未知。最后，对携带主机属性标签的 client IP 发起的所有 HTTP 会话或 TLS 会话提取实验所需特征，构成有标签的样本数据集。

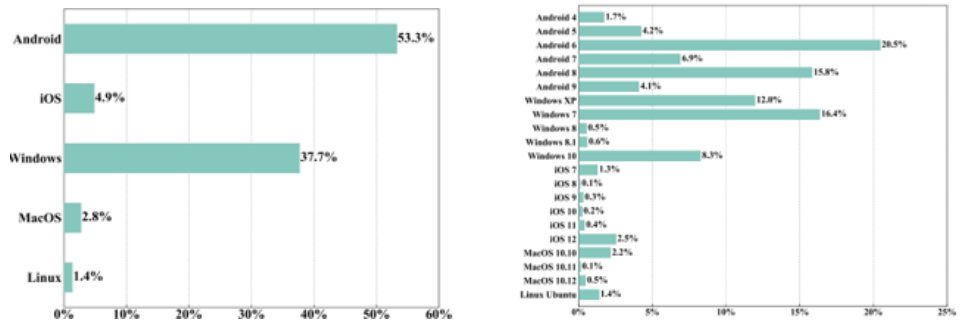
接下来统计样本数据集中的各类主机属性分布，如图 3.6 所示。在操作系统

```
GET /resources/stylelib/images/icon/icon-server-discipline.png HTTP/1.1
Host: sep.ucas.ac.cn
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.87 Safari/
537.36
DNT: 1
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://sep.ucas.ac.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: sepuser="bWlkPTQxNTU5OTQxLTYzZTUtNDM3NC1iZGI3LTV1ZTFhYWU3NTc4Yg==  "; JSESSIONID=I8T
+KiDoen2gIef3shzvxdZU.undefined
```

图 2.5 HTTP 协议请求报文中的 User-Agent 字

Figure 2.5 Fine-grained host attribute discovery in complex networks

属性方面，一共收有五类操作系统被收录，包括各类操作系统对应的二十一类主版本。其中，Android 设备的数量最多，其次是 Windows 设备和 iOS 设备。在各类操作系统主版本的数量上，安卓 6.X，7.X，8.X 和 Windows 7 占比最多。此外，由于从 HTTP 的 User-Agent 字段无法识别 Linux 发行版的主要版本，本章仅研究最流行的桌面 Linux 发行版即 Ubuntu 系统的网络特征。在浏览器属性方面，分别收集了 Firefox、Safari、IE、Chrome 以及 Opera 等五种浏览器类型，如图 3.7 所示，谷歌公司的 Chrome 浏览器和 apple 公司的 safari 浏览器在数据集中占比最多，这些基本符合国际市场操作系统和浏览器份额占比状况。



(a) (b)

图 2.6 数据集中操作系统分布 (a) 操作系统类型分布，(b) 操作系统版本分布

Figure 2.6 OASPL.(a) This is the explanation of subfig, (b) This is the explanation of subfig

2.4.2 模型对比

本小节分别对比逻辑回归模型、最近邻居模型、随机森林模型、XGBoost 模型以及 LightGBM 模型等五类机器学习模型在操作系统类型识别、操作系统版本识别和浏览器类型识别任务中的效果和性能。评估指标主要为准确率、精度、召回率、PR 曲线和模型预测速率。

如图 3.8 所示，在准确率、精度以及召回率方面，LightGBM 模型在所有识

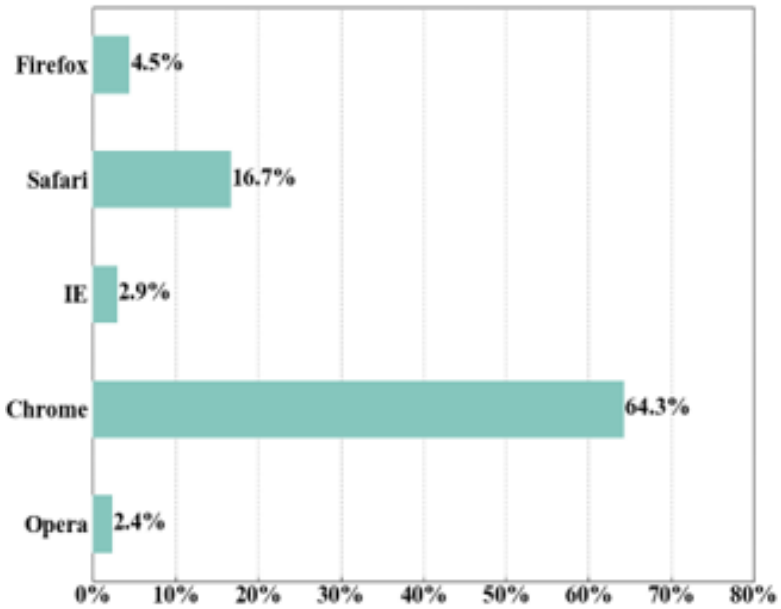


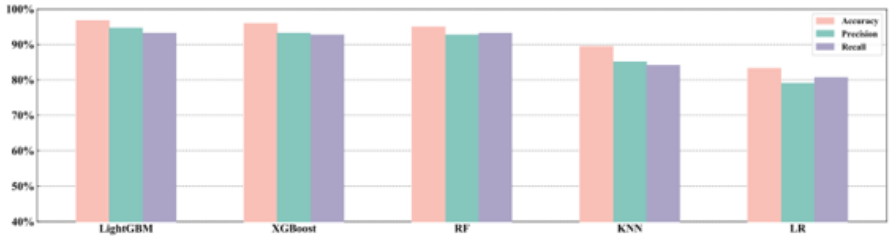
图 2.7 数据集中浏览器类型分布

Figure 2.7 Fine-grained host attribute discovery in complex networks

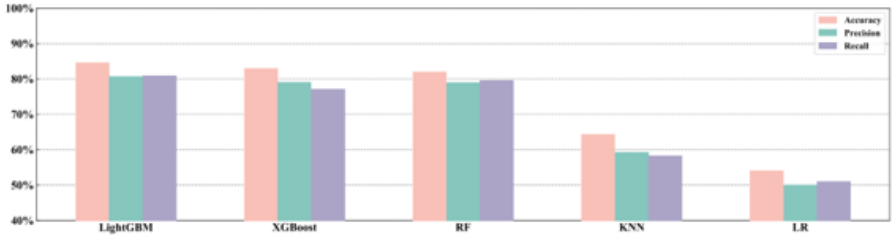
别任务中的表现均是最佳。在操作系统和浏览器的类型识别任务中，RF 模型、XGBoost 模型以及 LightGBM 模型的各项指标都在 90% 以上，充分显示了集成树模型的优越性。在主机属性识别任务中，大部分输入特征均为类别特征，而树模型的分类策略对这种离散特征的适应性非常强，再经过集成模型对基学习器的效果改善，使得集成树模型较为适用于该识别任务。KNN 模型和 LR 模型虽然三项指标都达到了 80% 左右, 但由于算法结构较为简单，更适合处理线性分类任务，在决策空间较为复杂的主机属性识别任务中效果较差。

PR 曲线是以召回率为横轴，以精度为纵轴绘制出的曲线，可以直观衡量一个模型的预测能力。在操作系统版本识别结果中，如图 3.9 所示，RF 模型、XGBoost 模型以及 LightGBM 模型的 PR 曲线均将 KNN 模型和 LR 模型的 PR 曲线完全包住，说明前三者的性能优于后者。而 RF 模型、XGBoost 模型、LightGBM 模型的 PR 曲线由于存在交叉点，不能根据以上标准分析优劣，但可以借助平衡点（BEP）即召回率和精度相等的点比较性能。可以看到，LightGBM 模型在平衡点处的值最大，说明该模型的性能更佳。

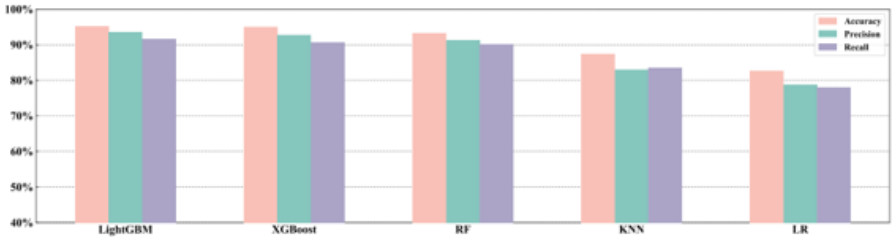
本章通过比较在操作系统版本识别任务中的吞吐量分析各模型的预测速率差异。如图 3.10 所示，LR 模型由于简单的分类策略，识别效率最高，比位于第二名的 LightGBM 模型快几十倍。在三类集成树模型中，LightGBM 模型的预测



(a)



(b)



(c)

图 2.8 识别任务准确率、精度以及召回率 (a) 操作系统类型，(b) 操作系统版本，(c) 浏览器类型

Figure 2.8 OASPL.(a) This is the explanation of subfig, (b) This is the explanation of subfig

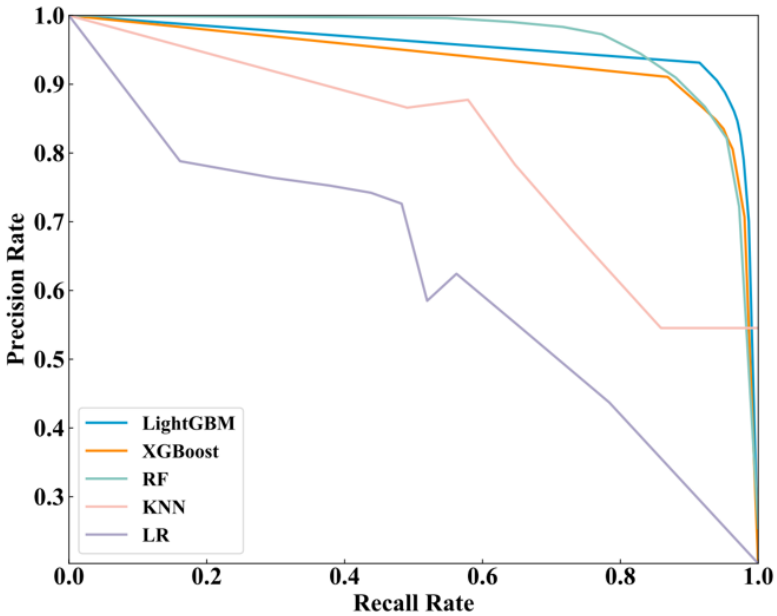


图 2.9 操作系统版本识别任务 PR 曲线

Figure 2.9 Fine-grained host attribute discovery in complex networks

速率最快，这主要是因为 LightGBM 模型采用直方图算法优化了树的生长速率，并对多线程任务进行了特殊优化，更适合多分类任务。对于 KNN 模型，由于独特的识别算法，其预测速率依赖于训练样本空间的规模，因此不适合在单一任务中与其他模型进行对比。

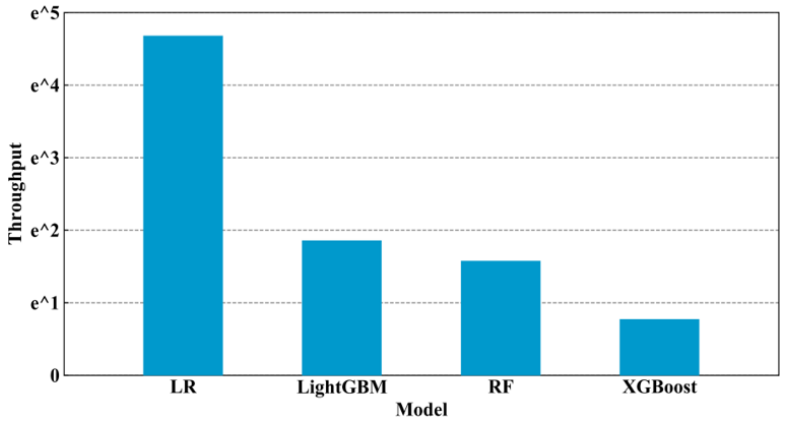


图 2.10 操作系统版本识别任务 PR 曲线

Figure 2.10 Fine-grained host attribute discovery in complex networks

通过以上实验结果对比可以发现，LightGBM 模型由于在 GBDT 算法的基础上进行了优化改进，在准确率、精度、召回率、PR 曲线以及模型预测速率等方面的表现都较为优越。因此，本章采用 LightGBM 模型作为主机属性发现任务中的最终模型。

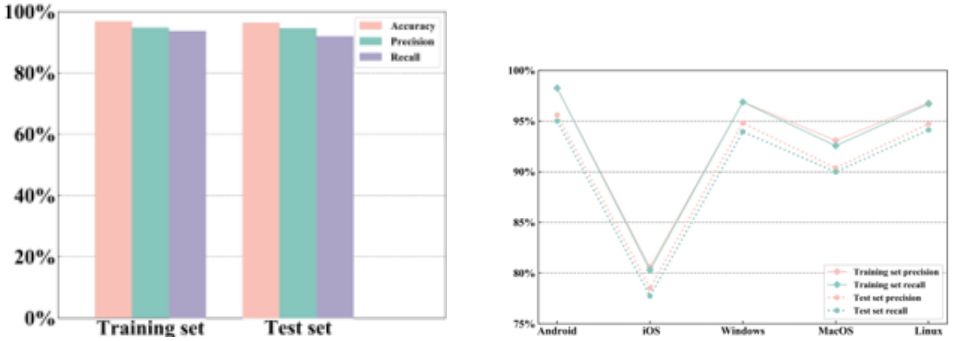
2.4.3　泛化性能评估

为了验证模型的泛化性能，本节将数据集按照日期进行切分，以前四天的数据作为训练集，最后一天的数据作为测试集，比较 LightGBM 模型在两个数据集上的性能差异。

如图 3.11 所示，在操作系统类型识别任务中，LightGBM 模型在训练集和测试集中的识别结果差异性非常小，准确率、精度以及召回率的最大差异不超过 1%，说明该模型具有很强的泛化能力。

2.4.4　主机属性识别结果

以 LightGBM 模型作为学习器，以 13 维协议字段特征和上百维流统计特征作为输入，其中流统计特征采集于每条流的前 50 个包，得到分类器在操作系统类型与版本、浏览器类型识别任务中的结果，见表 2.6-表 2.8。



(a) (b)

图 2.11 操作系统类型识别 (a) 准确率、精度和召回率柱状图，(b 精度和召回率折线图

Figure 2.11 OASPL.(a) This is the explanation of subfig, (b) This is the explanation of subfig

表 2.6 操作系统类型识别结果

Table 2.6 This is a sample table.

ID	OS type	Training set			Test set		
		Precision	Recall	F1	Precision	Recall	F1
1	Android	98.27%	95.60%	96.92%	98.25%	95.01%	96.60%
2	iOS	80.57%	78.53%	79.54%	80.30%	77.75%	79.00%
3	Windows	96.89%	94.80%	95.83%	96.87%	93.95%	95.39%
4	MacOS	93.08%	90.37%	91.70%	92.57%	89.98%	91.26%
5	Linux	96.80%	94.73%	95.75%	96.69%	94.13%	95.39%
AVE		93.12%	90.81%	91.95%	92.94%	90.16%	91.53%
Accuracy		96.89%			96.29%		

表 2.7 操作系统版本识别结果

Table 2.7 This is a sample table.

ID	OS Version	Test set		ID	OS Version	Test set	
		Precision	Recall			Precision	Recall
1	Android 4	81.23%	66.27%	12	iOS 7	99.44%	99.36%
2	Android 5	71.44%	60.67%	13	iOS 8	70.58%	86.04%
3	Android 6	94.64%	87.23%	14	iOS 9	72.18%	80.84%
4	Android 7	89.27%	59.85%	15	iOS 10	51.45%	69.11%
5	Android 8	67.07%	88.97%	16	iOS 11	49.80%	69.10%
6	Android 9	63.72%	42.41%	17	iOS 12	57.36%	85.79%
7	Windows XP	98.30%	94.11%	18	MacOS 10	99.26%	94.89%
8	Windows 7	91.08%	95.07%	19	MacOS 11	88.67%	66.50%
9	Windows 8	78.87%	43.36%	20	MacOS 12	81.45%	64.82%
10	Windows 8.1	90.99%	57.74%	21	Ubuntu	94.44%	95.38%
11	Windows 10	89.19%	88.22%	**AVE**		80.02%	75.99%

表 2.8 浏览器类型识别结果

Table 2.8 This is a sample table.

ID	BS type	Training set			Test set		
		Precision	Recall	F1	Precision	Recall	F1
1	Firefox	96.22%	93.31%	94.74%	94.09%	93.03%	93.56%
2	Safari	90.25%	90.15%	90.20%	89.63%	89.17%	89.40%
3	IE	94.45%	92.77%	93.60%	94.23%	91.47%	92.83%
4	Chrome	90.37%	87.57%	88.95%	89.76%	87.60%	88.67%
5	Opera	94.31%	92.55%	93.42%	94.44%	91.73%	93.07%
AVE		93.12%	91.27%	92.18%	92.43%	90.60%	91.50%
Accuracy		94.89%			94.02%		

2.4.5 本章小结

本章介绍了一种可在复杂网络中识别细粒度主机属性的技术方法。该方法基于加密流量中的 TLS 会话数据，以双向网络流为单位，提取 TCP/IP 协议栈指纹，包括 IP 层的跳数、包长、分片标识等字段，TCP 层的传输窗口大小、窗口缩放因子、最大报文长度等字段以及 TLS 层的版本、扩展长度、密钥算法套件序列等字段，再结合对应流的包长序列、时间序列和速率等统计特征，并将训练后的 LightGBM 模型作为分类器，可以从加密网络中被动识别主机的操作系统类型、版本和浏览器类型等主机属性。

本章首先对特征工程部分做了详细描述，分析了 TCP/IP 协议指纹的组成和作用。实验部分，巴拉巴拉。

第 3 章　基于加密流原始载荷的主机属性发现

上一章是基于 TCP/IP 协议栈指纹特征识别细粒度主机属性，然而这种特征的提取与构造特别依赖于丰富的网络流量分析经验，人工成本和时间成本较高。本章提出了一种基于深度学习模型的主机属性识别技术，利用表示学习的思想，不需要任何先验知识，只需将网络流的原始流数据作为分类器输入，便可完成细粒度的主机属性发现，并拥有更佳的识别效果。本章首先从该方法的背景意义和技术路线开始介绍，然后从可视化角度分析原始流信息的特征规律，接着通过分组实验对比不同深度学习模型的适用性，最后分析该方法的实验结果。

3.1　引言

由于近些年来计算性能和数据规模的发展，深度学习模型再度兴起，在信息时代的多个领域都有着重要应用。深度学习是机器学习中一种基于对数据进行表征学习的算法，网络流量本身也是一种数据，在通信技术发达的今天，网络世界中的流量数据规模已经十分庞大，可以充分训练成熟的深度人工神经网络模型。

在计算机视觉领域，最广为应用的模型是卷积神经网络 (CNN) 模型，该模型可以自动挖掘输入数据中的序列信息和局部特征，拥有非常优越的性能。而在加密会话中，多数协议头部并未加密，将协议头部以字节序进行解析，可以发现其存在较为明显的局部特征，因此，本章引入深度学习模型从表征学习的角度自动生成加密流指纹，并用于识别细粒度的主机属性。

如图 4.1 所示，本方法首先从 TLS 加密会话中提取 TCP SYN 包和 TLS Clinet Hello 包的原始流信息，在经过归一化处理和平整对齐后，传入训练后的深度人工神经网络模型，即可得到客户端主机的多维主机属性。

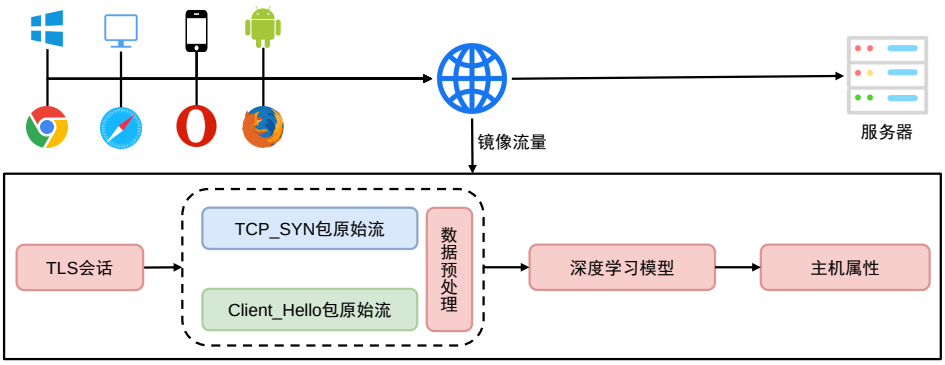


图 3.1 基于加密流原始载荷的主机属性发现

Figure 3.1 Fine-grained host attribute discovery in complex networks

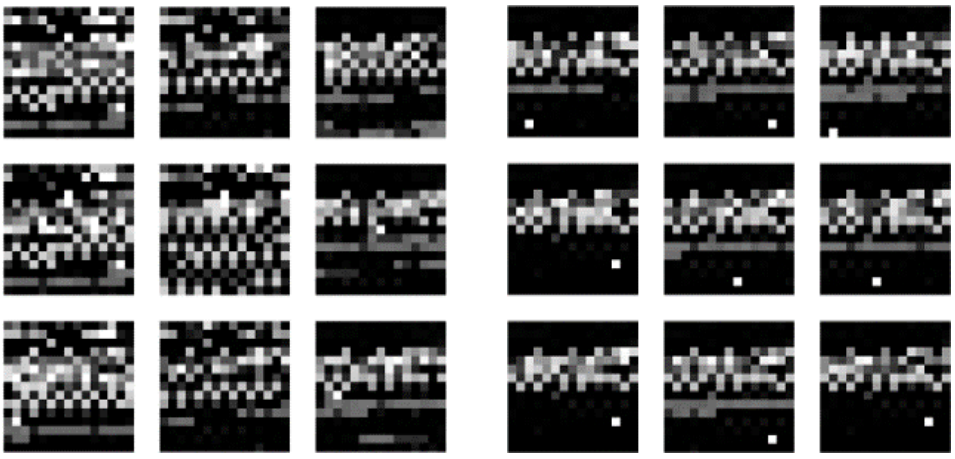
3.2 深度模型适应性分析

3.2.1 原始流量可视化分析

类似于第三章中介绍的 TCP/IP 协议栈指纹，可用于识别客户端主机属性的特征信息隐藏于 TCP 协议三次握手过程中的 SYN 包和 TLS 协议建立连接过程中的 Client Hello 包中。在一般的 TLS 会话过程中，被加密的部分通常是应用层载荷数据，而对于 TCP SYN 包和 TLS Client Hello 包这种控制报文，由于考虑到通信的效率和稳定性，往往都是非加密的。

在控制报文这种明文数据中，可以表达语义信息的最小单位一般是字节 (byte)，对应的数值为 0-255。将 TCP SYN 包和 TLS Client Hello 包中的数据拼接后，并绘制 256 级灰度图，如图 4.2 所示，分别是由 iOS 系统和 Linux 系统产生的九条原始流对应的灰度图。在图 a 中，可以看到 iOS 系统发送的多数 Client Hello 握手报文中都存在高频的大小值交错序列片段，这一现象在其他操作系统的原始流灰度图中是看不到的。在图 b 中，由 Linux 系统产生的流量灰度图同样存在可被观察到的一些特征。例如，Client Hello 握手报文中的有语义信息主要存在于报文的中上部，并且所有包都在尾部拥有一个独立亮点，其对应数值 0xff01，在 RFC 文档中，该亮点表示 TLS 协议的重协商信息扩展。

通过对不同操作系统的原始流量可视化分析，可以发现不同属性的主机产生的流量存在显著的局部特征。又因网络流量数据本身所具备的顺序特性，使得主机属性发现技术中的 TCP/IP 协议栈指纹特别适合用表征学习的思想进行自动挖掘。



(a)　　(b)

图 3.2 可视化分析 (a)iOS 原始流量灰度图 (b)Linux 原始流量灰度图

Figure 3.2 OASPL.(a) This is the explanation of subfig, (b) This is the explanation of subfig

3.2.2　基于卷积神经网络的原始载荷特征挖掘模型

上世纪 60 年代，Hubel 等人通过对猫视觉皮层细胞的研究，提出了感受野这个概念，到 80 年代，Fukushima 在感受野概念的基础之上提出了神经认知机的概念，可以看作是卷积神经网络的第一个实现网络，神经认知机将一个视觉模式分解成许多子模式（特征），然后进入分层递阶式相连的特征平面进行处理，它试图将视觉系统模型化，使其能够在即使物体有位移或轻微变形的时候，也能完成识别。卷积神经网络是多层感知机（MLP）的变种，由生物学家休博尔和维瑟尔在早期关于猫视觉皮层的研究发展而来，视觉皮层的细胞存在一个复杂的构造，这些细胞对视觉输入空间的子区域非常敏感，称之为感受野。

卷积神经网络是一种带有卷积结构的深度神经网络，卷积结构可以减少深层网络占用的内存量，包括三个关键的操作，其一是局部感受野，其二是权值共享，其三是汇聚层，可以有效降低网络参数规模，缓解了模型的过拟合问题。如图 4.3 所示，一般的卷积神经网络结构包括：卷积层，降采样层，全链接层。每一层有多个特征图，每个特征图通过一种卷积滤波器提取输入的一种特征，每个特征图有多个神经元。

卷积神经网络的核心思想就是局部感受野、是权值共享和 pooling 层，以此来达到简化网络参数并使得网络具有一定程度的位移、尺度、缩放、非线性形变稳定性。

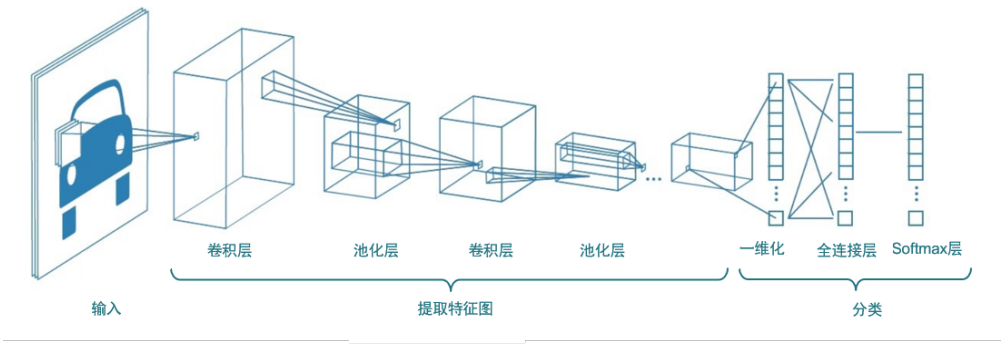


图 3.3 卷积神经网络的一般结构

Figure 3.3 Fine-grained host attribute discovery in complex networks

附录 A　中国科学院大学学位论文撰写要求

学位论文是研究生科研工作成果的集中体现，是评判学位申请者学术水平、授予其学位的主要依据，是科研领域重要的文献资料。根据《科学技术报告、学位论文和学术论文的编写格式》（GB/T 7713-1987）、《学位论文编写规则》（GB/T 7713.1-2006）和《文后参考文献著录规则》（GB7714—87）等国家有关标准，结合中国科学院大学（以下简称“国科大”）的实际情况，特制订本规定。

A.1　论文无附录者无需附录部分

A.2　测试公式编号 $\Lambda, \lambda, \theta, \bar{\Lambda}, \sqrt{S_{NN}}$

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \\ \frac{\partial (\rho \boldsymbol{V})}{\partial t} + \nabla \cdot (\rho \boldsymbol{V} \boldsymbol{V}) = \nabla \cdot \boldsymbol{\sigma} \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \boldsymbol{V}) = \nabla \cdot (k \nabla T) + \nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{V}) \end{cases} \quad \ldots (A.1)$$

$$\frac{\partial}{\partial t} \int_{\Omega} u \, \mathrm{d}\Omega + \int_{S} \mathbf{n} \cdot (u \boldsymbol{V}) \, \mathrm{d}S = \dot{\phi} \quad \ldots (A.2)$$

$$\mathcal{L}\{f\}(s) = \int_{0^-}^{\infty} f(t) e^{-st} \, \mathrm{d}t, \ \mathscr{L}\{f\}(s) = \int_{0^-}^{\infty} f(t) e^{-st} \, \mathrm{d}t$$

$$\mathcal{F}\big(f(x + x_0)\big) = \mathcal{F}\big(f(x)\big) e^{2\pi i \xi x_0}, \ \mathscr{F}\big(f(x + x_0)\big) = \mathscr{F}\big(f(x)\big) e^{2\pi i \xi x_0}$$

mathtext: $A, F, L, 2, 3, 5, \sigma$, mathnormal: $A, F, L, 2, 3, 5, \sigma$, mathrm: $\mathrm{A, F, L, 2, 3, 5, \sigma}$.

mathbf: $\mathbf{A, F, L, 2, 3, 5, \sigma}$, mathit: $\mathit{A, F, L, 2, 3, 5, \sigma}$, mathsf: $\mathsf{A, F, L, 2, 3, 5, \sigma}$.

mathtt: $\mathtt{A, F, L, 2, 3, 5, \sigma}$, mathfrak: $\mathfrak{A, F, L, 2, 3, 5, \sigma}$, mathbb: $\mathbb{A, F, L, 2, 3, 5, \sigma}$.

mathcal: $\mathcal{A, F, L, 2, 3, 5, \sigma}$, mathscr: $\mathscr{A, F, L, 2, 3, 5, \sigma}$, boldsymbol: $\boldsymbol{A, F, L, 2, 3, 5, \sigma}$.

vector: $\boldsymbol{\sigma}, \boldsymbol{T}, \boldsymbol{a}, \boldsymbol{F}, \boldsymbol{n}$, unitvector: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

matrix: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$, unitmatrix: $\boldsymbol{\sigma}, \mathbf{T}, \mathbf{a}, \mathbf{F}, \mathbf{n}$

tensor: $\boldsymbol{\sigma}, \mathsf{T}, \mathsf{a}, \mathsf{F}, \mathsf{n}$, unittensor: $\boldsymbol{\sigma}, \mathsf{T}, \mathsf{a}, \mathsf{F}, \mathsf{n}$

A.3 测试生僻字

霜蟾盥薇曜灵霜颸妙鬘虚霏淩澌菀枯菡萏沉寥窅冥毰毸濩落雲霅便嬛岧峣瀺灂姽婳愔嫕飒纚棽俪緸冤莩甲摛藻卮言倥侗椒觞期颐夜阑彬蔚倥偬澄廓簪缨陟遐迤逦缥缃鹣鲽憯懔闺闼璀错媕婀噌吰澒洞阛阓覼缕玓瓑逡巡諓諓琭琭瀌瀌踽踽叆叇氤氲瓠犀流眄蹀躞赟嬛茕頔璎珞螓首蘅皋惏悷缱绻昶皴皱颟顸愀然菡萏卑陬纯懿犇麤舜喤塷墍塮墼墐墒墒墓墔墕墖墘墖墚墛坠墝增墠墡墢墣墤墥墦墧墨墩墪樽墬墭堕墯墰墱墲坟墴墵垯墷墸墹墺墙墼墽垦墿壀壁壂壃壄壅壆坛壈壉壊垱壌壍壎壏壐壑壒压壔壕壖壗垒圹垆壛壜壝垄壠壡坜壣壤壥壦壧壨坝壪圭嫶嫷嫸嫹嫺娴嫼嫽嫾婳妫嬁嬂嬃嬄嬅嬆嬇娆嬉嬊娇嬍嬎嬏嬐嬑嬒嬓嬔嬕嬖嬗嬘嫱嬚嬛嬜嬞嬟嬠媛嬢嬣嬥嬦嬧嬨嬩嫔嬫嬬奶嬬嬮嬯婴嬱嬲嬳嬴嬵嬶嬷婶嬹嬺嬻嬼嬽嬾嬿孀孁孂娘孄孅孆孇孆孈孉孊娈孋孊孍孎孏嫫婿媚嵭嵮嵯嵰嵱嵲嵳嵴嵵嵶嵷嵸嵹嵺嵻嵼嵽嵾嵿嶀嶁嶂嶃崭嶅嶆岖嶈嶉嶊嶋嶌嶍嶎嶏嶐嶑嶒嶓嵚嶕嶖嶘嶙嶚嶛嶜嶝嶞嶟峤嶡峣嶣嶤嶥嶦峄峃嶩嶪嶫嶬嶭崄嶯嶰嶱嶲嶳岙嶵嶶嶷嶸嶹岭嶻屿岳帋巀巁巂巃巄巅巆巇巈巉巊巋巌巍巎巏巐巑峦巓巅巕岩巗巘巙巚帠帡帢帣帤帨帩帪帬帯帰帱帲帴帵帷帹帺帻帼帽帾帿幁幂帏幄幅幆幇幈幉幊幋幌幍幎幏幐幑幒幓幖幙幚幛幜幝幞帜幠幡幢币幤幥幦幧幨幩幪幭幮幯幰庁庂広庄庅庈庉庋庌庍庎庑庖庘庛庝庠庡庢庣庤庥庨庩庪库庬庮庯庰庱庲庳庴庵庹庺庻庼庽庿廀厕廃厩廅廆廇廈廋廌廍庼廏廐廑廒廔廕廖廗廘廙廛廜廞庑廤廥廦廧廨廭廮廯廰痈廲巡廸廹回廼迴廿弁弅弆弇弉弓弖弙弚弜弝弡弢弣弤弨弩弪弫弬弭弮弰弲弪弴弶弸弻弼弽弿彖彗彘彚彛彜彝彞彟彴彵彶彷彸役彺彻彽彾佛徂徃徆徇徉后徍徎徏径徒従徔徕徖徙徚徛徜徝从徟徕御徢徣徤徥徦徧徨复循徫旁徭微徯徰徱徲徳徴徵徶德徸彻徺徻徼徽徾徿忀忁忂忄忇忈忉忊忋忎忓忔忕忚忛応忝忞忟忪挣挦挧挨挩挪挫挬挭挮挰挱挲挳挴挵挶挷挸挹挺挻挼挿捀捁捂捃捄捅捆捇捈捊捋捌捍捎捏捐捑捒捓捔捕捖捗捘捙捚捛捜捝捞损捠捡换捣捤捥捦捧捨捩捪捫捬捭据捯捰捱捲捳捴捵捶捷捸捹捺捻捼捽捾捿掀掁掂扫抡掅掆掇授掉掊掋掍掏掐掑排掓掔掕挜掚挂掜掝掞掟掠采探掣掤掦措掫掬掭掮掯掰掱掲掳掴掵掶掸掹掺掻掼掽掾掿拣揁揂揃揅揄揆揇揈揉揊揋揌揍揎揑揓揔揕揖揗揘揙揤揥揦揧揨揫捂揯揰揱揲揳援揵揶揷揸揻揼揾揿搀搁搂搃搄搅搇搈搉搊搋搌搎搏搐搑搒摓摔摕摖摗摙摚摛掼摝摞摠摡斫斩斮斱斲斳斴斵斶斸旪旫旮旯晒晓晔晕晖晗晘晙晛晜晞晟晠晡晰晣晤晥晦晧晪晫晬晭晰晱晲晳晴晵晷晸晹晻晼晽晾晿暀暁暂暃暄暅暆暇晕晖暊暋暌暍暎暏暐暑暒暓暔暕暖暗旸暙暚暛暜暝暞暟暠暡暣暤暥暦暧暨暩暪暬暭暮暯暰昵暲暳暴暵

参考文献

作者简历及攻读学位期间发表的学术论文与研究成果

本科生无需此部分。

作者简历

casthesis 作者

吴凌云，福建省屏南县人，中国科学院数学与系统科学研究院博士研究生。

ucasthesis 作者

莫晃锐，湖南省湘潭县人，中国科学院力学研究所硕士研究生。

已发表 (或正式接受) 的学术论文:

1. ucasthesis: A LaTeX Thesis Template for the University of Chinese Academy of Sciences, 2014.

申请或已获得的专利:

(无专利时此项不必列出)

参加的研究项目及获奖情况:

可以随意添加新的条目或是结构。

致　谢

感激 casthesis 作者吴凌云学长，gbt7714-bibtex-style 开发者 zepinglee，和 ctex 众多开发者们。若没有他们的辛勤付出和非凡工作，LaTeX 菜鸟的我是无法完成此国科大学位论文 LaTeX 模板 ucasthesis 的。在 LaTeX 中的一点一滴的成长源于开源社区的众多优秀资料和教程，在此对所有 LaTeX 社区的贡献者表示感谢！

ucasthesis 国科大学位论文 LaTeX 模板的最终成型离不开以霍明虹老师和丁云云老师为代表的国科大学位办公室老师们制定的官方指导文件和众多 ucasthesis 用户的热心测试和耐心反馈，在此对他们的认真付出表示感谢。特别对国科大的赵永明同学的众多有效反馈意见和建议表示感谢，对国科大本科部的陆晴老师和本科部学位办的丁云云老师的细致审核和建议表示感谢。谢谢大家的共同努力和支持，让 ucasthesis 为国科大学子使用 LaTeX 撰写学位论文提供便利和高效这一目标成为可能。