心求至善

SOUTHEAST UNIVERSITY

基于设备指纹和行为可信的物联网访问控制系统

黄强

东南大学

学校代码: 10286
分类号: 000
密级: 公开
U D C: 000
学号: 160922

東南大學
工程硕士学位论文

基于设备指纹和行为可信的物联网访问控制系统

研究生姓名: 黄强
导师姓名: 宋宇波 副教授

申请学位类别 工学硕士　学位授予单位 东南大学
一级学科名称 信息与通信工程　论文答辩日期 2019年4月17日
二级学科名称 信息安全　学位授予日期 2019年4月17日
答辩委员会主席　评阅人

2019年4月17日

学校代码: 10286
分 类 号: 000
密 级: 公开
U D C: 000
学 号: 160922

东南大学

SOUTHEAST UNIVERSITY

工程硕士学位论文

基于设备指纹和行为可信的物联网访问控制系统

研究生姓名: 黄强

导 师 姓 名: 宋宇波 副教授

申请学位类别 工学硕士 学位授予单位 东南大学

一级学科名称 信息与通信工程 论文答辩日期 2019 年 4 月 17 日

二级学科名称 信息安全 学位授予日期 2019 年 4 月 17 日

答辩委员会主席 评 阅 人

2019 年 4 月 17 日

东南大学

硕士学位论文

基于设备指纹和行为可信的物联网访问控制系统

专 业 名 称: 信息与通信工程

研究生姓名: 黄 强

导 师 姓 名: 宋宇波 副教授

AN ACCESS CONTROL SYSTEM BASED ON DEVICE FINGERPRINT AND BEHAVIOR TRUST FOR IOT

A Thesis submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

Huang qiang

Supervised by:

Associate Prof. Song Yubo

and

School of Information Science and Engineering

Southeast University

2019/4/17

东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：_____ 日 期：_____

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名：_____ 导师签名：_____ 日 期：_____

摘　要

随着物联网设备大规模接入网络，网络访问控制管理变得愈加重要。这些问题本可以通过访问控制技术得以缓解，但是传统的基于复杂加密协议和认证客户端的认证技术不再适用于计算和存储资源受限制的物联网设备。基于此，本文设计并实现了基于设备指纹和行为可信的物联网访问控制系统。该系统首先提取设备启动阶段的网络流量特征，并利用机器学习算法识别设备的类型。设备类型信息将作为权限初始分配的依据。然后系统基于设备当前网络行为和历史网络行为的偏离程度计算其行为可信度。行为可信度将作为设备与权限分配关系动态调整的依据，以实现对同类型设备的访问控制策略的差异化管理。本文的主要工作和创新点如下：

1. 针对传统的基于复杂加密协议和认证客户端的访问控制技术不再适用于计算和存储资源受限制的物联网设备的问题，本文设计并实现了基于设备指纹和行为可信的物联网访问控制系统。该系统首先基于设备启动阶段的网络流量特征识别设备的类型，实现权限的初始分配。然后基于设备的当前行为和历史行为的偏离程度计算其行为可信度，实现设备与权限分配关系的动态调整。
2. 针对现有设备指纹识别算法对相似型号设备类型的识别准确率低的问题，提出了一种二阶段指纹识别算法——TSMC-SVM。该算法将修正余弦相似度引入多分类机器学习模型中，提升了对相似型号设备类型的识别准确率。实验表明，TSMC-SVM 的平均识别准确率达到 93.2%。同时为降低相似度匹配的时间复杂度，提出了一种样本预处理方法。该方法通过计算样本预处理因子，将相似度匹配的时间复杂度从 $\mathcal{O}(nm)$ 降低到了 $\mathcal{O}(n)$。
3. 为实现对设备个体访问控制策略的定制化管理，本文提出了一种基于设备行为可信度的访问控制模型。该模型将设备网络行为可信度和权限可信度阈值引入基于角色访问控制模型中。对于设备行为可信度，模型从多个维度观察设备当前网络行为和历史网络行为的偏离程度生成评价因子，然后通过模糊综合评价方法计算最终的可信度。权限可信度阈值可以根据资源的环境上下文动态设置。
4. 为实现通过交换机完成对设备资源访问的管理，本文提出了一种基于 VLAN 的权限管理策略。该权限管理策略首先将资源划分到不同 VLAN 中，然后依据设备指纹和设备行为可信度知识调整设备与 VLAN 的隶属关系，以实现设备对资源访问的管理。
5. 完成了基于设备指纹和行为可信的物联网访问控制系统的设计与实现。并从逻辑视图、过程视图、实现视图、物理视图和场景视图完整介绍了系统的设计与实现过程。

关键词： 设备指纹，行为信任，访问控制，机器学习，物联网

Abstract

As mass devices access to network in the era of the Internet of Things (IoT), the network access control becomes more important. These problems could have been alleviated by access control techniques, but traditional authentication mechanisms based on complex encryption protocol and installing authentication agents are no longer suitable for IoT devices with limited computing and storage resources. Therefore, an IoT access control system based on device fingerprint and behavioral trust is designed and implemented in this thesis. Firstly, the system extracts network traffic features as devices starting and uses these features to identify device types by machine learning algorithms. The information of device types will be used for assigning privileges initially. Then, for adjusting dynamically the assignment relationship between devices and privileges, the system introduces the device behavioral truth which describes the deviation degree of the current behavior and historical behavior of a device. It is used to achieve differentiated management of access control policies for devices within same type. The main work and innovations of this thesis are as follows:

1. The traditional access control technologies based on complex encryption protocol and authentication mechanism are no longer suitable for IoT devices with limited computing and storage resources. Therefore, a novel access control system based on device fingerprint and behavioral truth for IoT is proposed in this thesis. Firstly, the system identifies device types based on network traffic features as devices starting, which for assigning privileges initially. Then, for adjusting dynamically the assignment relationship between devices and privileges, the system introduces the device behavioral truth which describes the deviation degree of the current behavior and historical behavior of a device.

2. The TSMC-SVM, a two-stage classification algorithm, is proposed for solving the problem that existing device fingerprint identification algorithm can not effectively identify devices' type from the same manufacturer which equips these devices with similar hardwares, firmwares and sofwares. The algorithm introduces the adjustment cosine similarity into the multi-classification model, which improves the recognition accuracy of these similar device types. Experiments illustrate that the average identification accuracy of the TSMC-SVM reaches 93.2%. Furthermore, samples are processed in advanced, which reduces the time complexity of similarity matching from $\mathcal{O}(nm)$ to $\mathcal{O}(n)$.

3. An access control model based on the device behavioral trust is proposed for achieving differentiated management of access control policies for devices within same type. The

model introduces device behavioral truth and truth threshold into the role-based access control model. For the device behavioral truth, the model extracts evaluation factors by observing the deviation degree between the current network behavior and historical network behavior of devices from multiple dimensions. And then calculating the behavioral trust through the fuzzy comprehensive method. The truth threshold can be set dynamically according to the environmental context of the resource.

4. A VLAN-based privileges management strategy is proposed in this thesis for management of device access resources through switches. The resource management policy first divides the resources into different VLANs. And then adjusts the membership relationship between the device and the VLAN according to the device fingerprint and the device behavior trust.
5. This these completed the design and implementation of an IoT access control system based on device fingerprint and behavioral trust. And the process of design and implementation for the system are fully described with logical view, process view, implementation view, physical view and scene view.

Keywords: Device fingerprint, Behavioral trust, Access control, Machine learning, IoT

目录

插图目录

表格目录

第一章 绪论

1.1 研究背景及意义

以物联网为核心基石的第四次技术革命正在引领人类社会迈向万物感知、万物互联、万物智能的新时代。物联网通过提供一种信息共享和协调的方式，使人与物、物与物实现互联。然而在物联网驱动全行业数字化的同时，也带来了新的网络安全风险。纵观近年来物联网安全的发展态势，不难发现以下趋势：

(1) 物联网市场规模快速增长，安全支出持续增加。一方面，根据 GSMA 预测显示，2025 年全球物联网设备（包括蜂窝和非蜂窝）联网数量将达到 252 亿，远高于 2017 年的 63 亿[1]。此外，工业物联网设备联网数量将在 2025 年达到 138 亿，相比于 2016 年，增长 500%。同时根据华为 GIV 预测显示，到 2025 年个人智能终端数量将达到 400 亿，这些联接推动各行业的数字化转型，创造总数字经济高达 23 亿美金[2]。另一方面，物联网安全事件频发，全球物联网安全支出不断增加。根据 Gartner 预测显示，到 2020 年，企业安全威胁中物联网占比 25%。同时预计到 2021 年物联网终端安全、网关安全和专业服务分别将达到 6.13 亿美元、4.15 亿美元、20.71 亿美元。总支出相比于 2017 年 11.74 亿美元增长 166%[3]。

(2) 用户隐私威胁日益加剧。随着物联网终端逐步迈入家庭，不安全的设备终端将用户隐私暴露在开放的互联网环境中。例如玩具和婴儿监视器泄露了家人的日常隐私[4;5]。同时，随着物联网与公有云的结合，通过对个人数据的访问和处理，引入了更多的隐私窃取和泄漏的风险。而且随着黑色产业链的成熟，产生了更多的为了经济利益的物联网攻击，比如对加密货币挖矿、医疗设备以及工业设施的勒索攻击。

(3) 攻击网络基础设施门槛日益降低。由于物联网设备种类繁多，且计算和存储资源有限，难以实施统一的安全保护机制。而且由于行业安全标准的滞后，导致其极易成为网络不法分子的攻击对象。同时物联网设备基数大、分布广，一旦出现漏洞将形成大量被控设备组成的僵尸网络，导致攻击者向网络基础设施发起分布式拒绝服务攻击的门槛降低[6]。

上述安全问题本可以通过网络访问控制技术得到缓解。传统的网络访问控制技术严重依赖于复杂加密协议和认证机制来保障其安全性和可靠性。但是这些资源消耗型的方案却无法适用于计算资源和存储资源有限的物联网设备。因此本文的目的是希望寻找一种设备身份标识。该标识能够为认证提供必要的身份信息，而且生成该标识无需消耗设备的计算和存储资源。同时应该注意，这些身份标识不应该是软标识，例如设备的 MAC 地址和 IP 地址，因为它们极易容易伪造。例如，在 Linux 环境下，攻击者可以利

用 raw socket 编程就能自定义报文的 MAC 地址和 IP 地址，实际上攻击者能够自定义报文中的任何字段。

设备的网络流量是设备固有属性和行为的一种表现形式。本文通过提取设备的网络流量特征以生成认证所需的身份信息。这些特征就像生物指纹一样具有唯一性，因此称其为设备指纹。设备指纹的唯一性和可行性是两个值得关注的问题。不幸的是，实验结果显示本文的设备指纹不具有严格意义上的唯一性。它只能识别设备的类型，而无法识别设备个体。但是这已经为本文的访问控制系统提供了重要的身份信息了。因为设备的类型信息已经能够明确其职责，并赋予特定的权限。进一步的工作只需要根据设备的当前行为与历史行为偏离程度调整权限的分配关系。对于可行性问题，理论上系统可以提取在任何时间的设备网络流量特征以生成设备指纹。但是在网络访问控制系统中这并不可行。如果将设备的运行时间大致分为启动阶段和工作阶段，那么可以发现启动阶段的设备网络流量要比工作阶段稳定许多。对于物联网设备，由于其功能的单一性，设备启动所需的配置信息相对固定，有些甚至通过硬编码被固化在硬件或者软件中。而在工作阶段，由于时间的变迁以及网络环境的变化，设备会进行相应的调整以适应新的上下文。或许可以发现流量的某种具有周期性的固定模式，就像 Ke Gao[7] 等人利用小波分析从多组网络序列提取出一种相异但是可重复的模式。但是重现该模式所需的时间窗口大小却无法保证，Ke Gao 等人发现的模型需要 10^5 个网络报文。从实时性角度出发，身份认证不应该经历这么长的时间窗口。基于以上原因本文选择设备启动阶段的网络流量提取生成设备指纹的信息。

本文利用机器学习从设备网路流量中挖掘设备身份信息。机器学习作为一门涉及统计学、概率论、计算法杂性理论的交叉学科，通过在大量数据中分析挖掘规律，并利用这些规律预测未知数据。机器学习已经被成功应用于计算机视觉、自然语言处理、生物特征识别、医学分析等领域。同样，网络异常检测[8;9;10;11] 及网络流量分类[12;13;14;15;16] 也开始运用机器学习来分析庞大且复杂的网络流量。基于此，机器学习同样适用于挖掘终端设备的身份信息。通过提取设备所属的网络流量特征并加以分析，为网络访问控制系统提供必要的身份凭证。该设备身份提取方案无需在终端安装身份信息采集客户端，适用于物理和逻辑资源匮乏的物联网终端设备。同时，由于基于机器学习的设备网络指纹技术分析的是网络报文头部信息和流量的统计规律，同样适用于加密流量场景。由于不深入分析报文载荷，符合当今对用户隐私保护日益关注的需求。

访问控制的目的是实现对可信任用户的资源访问权限的分配。但由于本文的设备指纹不具有严格意义的唯一性，即只能提供设备类型信息。而基于设备类型信息无法实现对设备个体的个性化权限管理。因此，本文提出了一种基于行为可信度的授权机制（Bebavior Trust-Based Access Control，BTBAC）。该机制采用基于角色和可信度的访问控制模型，通过评估设备行为可信度和权限可信度阈值来动态调整授权状态。设备行为可信度是通过度量设备历史行为和当前行为的偏离程度得到的。而权限可信度阈值可以根据资源的当前上下文动态设定。

1.2 国内外研究现状

设备指纹分为主动式设备指纹和被动式设备指纹，这取决于所采集的流量是通过主动探测得到还是被动监控得到。主动式设备指纹通过发送精心构造的探寻帧，进而分析响应报文以生成指纹。而被动式设备指纹则是在网络节点处监听并捕获设备的网络流量，为进一步的指纹生成提供必要的数据。

Nmap[17] 是一款用于网络发现和安全审计的主动式指纹识别工具，其通过主动探寻响应分析目标主机的操作系统的类型和版本信息。与使用 TCP/IP 协议栈特征来识别操作系统指纹的 Nmap 不同，Xprobe[18] 通过分析 ICMP 协议信息来区分操作系统，其的核心技术是模糊统计分析[19]。Bratus[20] 等人通过观察无线设备对一系列的非标准或者畸形的 802.11 帧的响应来发现设备之间的差异（芯片组、固件和驱动程序的差异）。实验证明，这样的响应存在很大的差异，足以区分许多常见的无线设备。Sieka[21] 提出了一种基于时间统计分析和机器学习的无线设备指纹识别技术。该技术研究了无线设备执行各种通信所需时间量的变化，特别是与 802.11 的 MAC 层通信中的认证过程相关的时间变换。这个时间度量方法侧重于来自设备的第一次确认（ACK）和第一次身份验证响应（AUTH）之间的时间差。该过程经过多次重复试验之后得到足够的数据，然后使机器学习构建分类器。该方法在一组无线设备上的测试中取得了 86% 的预测准确率。然而试验的设备非常有限（准确的说是 5 个）。因此，随着更多的设备不断扩充，预测准确率可能会进一步降低。Corbett[22] 等人提出使用频谱分析来识别无线网卡的类型。该技术可用于支持检测未授权系统的无线网卡。他们发现主动扫描引起的无线流量表现出了一种周期的和稳定的传输模式，而且该模式对于不同的无线网卡存在差异。实现结果表明该指纹识别技术的误报率为零。

主动式设备指纹需要精心构造探寻帧，以期待能够识别设备的响应报文。因此合适的探寻帧构造成为了其技术瓶颈。同时针对不同的设备以及协议需要不同的探寻帧，这对于可扩展性提出了挑战。此外，主动式设备指纹的另一个问题是可能会触发网络异常检测系统报警。主动式设备指纹获取响应报文的过程类似于网络扫描，这些行为会被网络管理系统检测到，并标识为异常行为。网络带宽的消耗本不应该是主动式设备指纹的技术瓶颈，但是在某些特殊场合下，确实会影响到系统的正常运行，例如工业控制系统[23]。

相比于主动式设备指纹，被动式设备指纹技术更受到研究人员的关注。文献[7;24;25;26;27;28]研究的是无线设备的指纹。文献[7] 基于不同厂商的 AP 在体系结构异构性（例如：芯片、固件、驱动）的事实，将小波分析应用于网络分组序列，提取出一种不同但可重复的模式，从而实现对不同类型 AP 的分类。尽管此方法在实验环境下的识别准确率为 100%，但识别所需的 10^5 的报文数量严重影响了识别的实时性。文献[28] 使线性规划和最小二乘拟合技术分析 802.11 的时间同步函数（TSF）检测非法 AP。实验结果发现同类型 AP 的时钟偏差随着时间的推移保持一致，而在不同类型 AP 之间存在显著差异。Franklin[26] 等人研究了基于不同类型的 802.11 无线驱动的设备指纹技术。通过设备驱动程序中的

主动通道扫描策略之间统计关系的差异实现对设备类型的区分。此技术关注的焦点在设备驱动程序而非设备类型本身，因此无法识别驱动程序相同的不同类型设备。

基于物理层的设备指纹识别已受到相当多的关注[29;23;30]。Vladimir[29] 等人开发了一种基于模拟缺陷的辐射测量方法用于识别无线网卡。该方法严重依赖于物理帧的可用性，而在物联网环境中设备可能通过交换设备后者其他网络节点相连，导致物理帧的不可用。Formby[23] 等人研究了工业控制系统设备的物理特征，并提出了两种物理指纹识别方法。Radhakrishnan[30] 等人提出了一种基于设备时钟偏差的指纹识别技术，并提供了一个达 300GB 的数据集用于设备指纹研究。

近年来，被动式设备指纹有又得到了高度关注[31;32;33;34;35;36;37]。Yang[31] 等人通过分析物联网设备在不同网络层中协议的差异，设计并开发了一个大规模设备发现系统。通过该系统研究人员在网络中发现了 1530 万个网络设备并分析了这些设备在网络空间中的分布特征。而 Aneja[32] 等人则聚焦于将深度学习应用于设备指纹识别，实验结果显示其准确率达到 86.7%。Miettinen[33] 等人提出并实现了一个基于设备类型指纹的物联网安全框架，但是对于形式型号的设备类型的识别准确率较低。

综上所述，虽然设备指纹技术在国内外已经取得了很多重要的技术突破，但是仍然存在不足。相比于被动式设备指纹技术，主动式设备指纹技术较为成熟，而且已经出现得到业内肯定的工具，例如 Nmap、Xprobe 等。但是由于需要针对不同类型的设备构造相应的探寻帧，成为了其技术瓶颈。而且扫描式的响应数据获取方式，容易被网络安全监测系统误判为非法设备。而现阶段，被动式设备指纹技术主要针对的是无线设备的识别，如无线接入点和无线网卡等。文献[31;33] 对普适性的设备指纹技术作出了开创新研究，但是对于相似型号的设备的识别准确率低下。

1.3 本文的主要工作与章节安排

1.3.1 本文的主要工作

本文研究了基于设备指纹和行为可信的物联网访问控制系统。首先分析了现有的设备指纹技术的研究成果和不足之处，并基于此提出了二阶段设备指纹识别方法；其次为实现对设备个体的权限管理，提出了基于设备行为可信度的访问控制模型；在此基础之上设计并实现了一个基于设备指纹和行为可信的物联网访问控制系统。本文完成的主要工作有：

1. 分析了基于设备指纹和行为可信的物联网访问控制系统的研究背景，概述了国内外的研究进展以及存在的问题。
2. 针对物联网设备计算资源和存储资源受限制而无法支持复杂的加密协议和安全认证机制的问题，提出了一种被动式设备指纹识别技术。该指纹识别技术通过在网络节点处被动监听设备的网络流量，并提取流量的特征信息，通过算法建立指纹识别模型。该技术能够有效识别设备的类型。

3. 针对设备指纹的稳定性问题，提出了一种设备启动指纹技术，该技术通过提取设备启动阶段的网络流量特征生成设备指纹。这些特征包括报文特征和流量统计特征。其中前者刻画的是流量的局部细节信息，而后者刻画的是的流量的全局统计信息。

4. 针对现有设备指纹识别算法对相似型号设备的识别准确率低的问题，提出了一种二阶段指纹识别算法。而算法首先通过多分类模型完成对设备类型的初步分类，然后对于无法分类的设备实施相似度匹配，以实现进一步的细分。同时为降低相似度匹配算法的时间复杂度，提出了一种样本预处理方法。该方法将相似度匹配算法的时间复杂度从 $\mathcal{O}(nm)$ 降低到了 $\mathcal{O}(n)$。

5. 针对本文的设备指纹技术只能完成对设备类型的识别，而无法实现对设备个体识别的问题，提出了一种基于设备行为可信度的访问控制模型。该模型基于设备当前网络行为和历史行为的偏离程度评估其信任程度。该评估过程由模糊综合评价技术完成。

6. 针对交换机访问控制策略设计薄弱的问题，提出了一种基于 VLAN 的权限管理策略。该管理策略设计结合设备指纹和设备行为可信度技术能够有效实现对资源的访问控制。

7. 完成了基于设备指纹和行为可信的物联网访问控制系统的设计与实现。本文根据“4+1”视图模型介绍了系统的逻辑视图、过程视图、实现视图、物理视图和场景视图。

1.3.2 本文的章节安排

基于上述研究内容，本文共分为六章，每章的具体安排如下：

第一章为绪论。首先介绍了本文的研究背景，分析了物联网设备在网络访问控制方面存在的问题；接着总结了现阶段设备指纹技术在国内外的研究进展。最后提出了本文的研究内容和章节安排。

第二章为技术背景。首先介绍了粗糙集理论的基础知识，包括基本概念和和基于粗糙集的决策知识发现原理。其次介绍了网络准入控制相关的理论知识，包括基本概念和相对成熟的网络准入控制方案，以及基于可信度的访问控制的相关知识。最后介绍了用于设备指纹识别的机器学习理论，包括基本概念、机器学习流程和各种分类算法。

第三章为基于 TSMC-SVM 的设备指纹识别算法。首先介绍了如何对设备启动阶段的网络流量报文进行特征提取和编码。然后提出了 TSMC-SVM，一种二阶段多分类算法。最后通过实验分析该设备指纹识别算法的有效性，并与其它分类算法进行比较。

第四章基于设备行为可信度的访问控制模型。首先介绍了基于设备行为可信度的授权机制，然后介绍了基于模糊综合评价的行为可信度评价算法。

第五章为基于设备指纹和行为可信的物联网访问控制系统设计与实现。根据“4+1”视图模型介绍了系统的逻辑视图、过程视图、实现视图、物理视图和场景视图。

第六章为总结和展望，归纳和总结了全文的内容，分析了基于设备指纹和行为可信的物联网访问控制系统需要进一步完成的工作。

第二章 技术背景

2.1 引言

基于设备指纹和行为可信的物联网访问控制系统的核心思想是根据设备的网络流量和网络行为判断设备是否有相应资源的访问权限。系统首先进行设备指纹识别，以获得相应的角色。接下来，在设备运行阶段内，系统会周期性分析和判别该设备网络行为的可信程度，以便实现动态授权。在设备指纹识别阶段，系统通过机器学习算法分析设备的网络流量特征，以实现对设备的识别。而在设备网络行为可信度判别阶段，系统从多个维度观察设备的网络行为并提取评价因子。然后通过模糊综合评价方法计算其行为可行度。基于上述面熟，本章主要从访问控制、机器学习和模糊集理论三个方面介绍基于设备指纹和行为可信的物联网访问控制系统的背景技术知识。

2.2 访问控制

2.2.1 基本概念

访问控制是一种限制用户对系统的物理或逻辑资源访问的技术。其实施过程主要分为两个阶段：身份认证和权限分配。身份认证阶段中，系统接收用户访问资源时提交的身份凭证并验证合法性。这些凭证可以有多种形式，包括密码，个人识别码（PIN），安全令牌，生物识别或其他验证元素。然后系统根据身份认证结果实施权限分配，以最大限度地降低未经授权访问资源的风险。访问控制主要有主体、客体和策略规则三个元素构成：

(1) 主体：主体是访问控制模型中信息流的启动者，由其主动发起对客体的访问请求。在此过程中，主体必须以主动或者被动的方式提交身份的凭证，以便系统对其合法性的判别。主体通常指用户，也可以是运行中的进程或者系统连接的设备；

(2) 客体：客体是受控信息的载体或者从其他主体和客体接受信息的实体，通常也被称作资源。主体有时也会成为访问或受控的对象，如一个主体可以向另一个主体授权，一个进程可以控制多个子进程等。这时受控的主体也被认为是一种客体。

(3) 策略规则：策略规则是访问控制模型中主体对客体的操作行为和约束条件的集合，它反映了一种授权模式。一般来讲，系统需要根据资源的类型设计不同的策略规则，或者同时运用多种策略规则来保护资源的机密性和完整性。

相比于主体和客体的概念，策略规则是区别于不同的访问控制模型的关键。根据不同的策略规则，常见的访问控制模型有自主访问控制（DAC）、强制访问控制（MAC）、

基于角色的访问控制（RBAC）和基于属性的访问控制（ABAC）。本节接下来对上述访问控制模型进行简单介绍。

2.2.2　DAC

DAC 模型的核心思想是由主体管理客体的访问权限规则。在 DAC 模型中，每个客体都隶属于一个主体，该主体是客体的创建者或者是由系统分配。对客体的访问策略由其所属主体制定，即主体决定是否将其拥有的客体的访问权限授予其他主体。这种模型使得主体具备了对其拥有的客体的绝对控制权。Unix 文件系统是一个典型的 DAC 模型，每一个文件都属于一个用户。同时文件拥有者可以定义其他用户对该文件资源的读、写和执行权限规则。DAC 模型能够有效满足资源管理的安全要求，但是由于对资源的访问策略是由其拥有者制定，因此系统对权限的管理比较松散。

2.2.3　MAC

MAC 一种基于资源敏感度和用户许可安全级的访问控制技术，其策略规则由系统制定并强制执行。该模型实现方式是根据敏感程度为每一个客体分配一个安全等级标签，而每一个主体也有一个对应的安全等级标签。主体能够访问客体取决于两者的安全标签的等级关系。一般地，MAC 模型以上读、下读、上写和下写四种访问模式来制定访问策略规则。图2.1 是 Bell-Lapadula 模型，该模型利用禁止上读和禁止下写来实现资源的保密性。而图2.2 是 Biba 模型，该模型则是利用禁止下读和禁止上写来实现数据的完整性。虽然 MAC 是最安全的访问控制策略，但需要仔细规划和持续监控，以使资源对象和用户的分类保持最新。

图 2.1: Bell-Lapadula 模型

2.2.4　RBAC

RBAC 是一种被广泛使用的访问控制机制。其核心思想是对客体的访问权限不直接面向主体，而是在主体与客体之间建立了一个角色层。角色层中的每一个实体会绑定一

图 2.2: Biba 模型

组权限实体。角色与客体之间的关系是多对多的，即一个角色能够拥有多个客体的权限，一个客体的权限能够分配给多个角色。当角色被分配给某一主体时，该主体就拥有了该角色绑定的权限集合。主体和客体之间的的关系也是多对多的，即一个主体能够拥有多个角色，一个角色能够分配给多个主体。图2.3 是 RBAC 模型中主体、角色和客体的关系示意图。RBAC 模型的目的是为了解耦主体与客体的关系，其依据是角色-客体之间的变化比主体-角色要慢得多。角色-客体之间的关系定义是责任单元划分的过程，即完成某一任务所需的权限的集合，该关系是相对稳定的。而主体-角色之间关系的定义是系统对主体授权的过程。因此 RBAC 模型降低了权限管理的复杂度并提高了伸缩性。

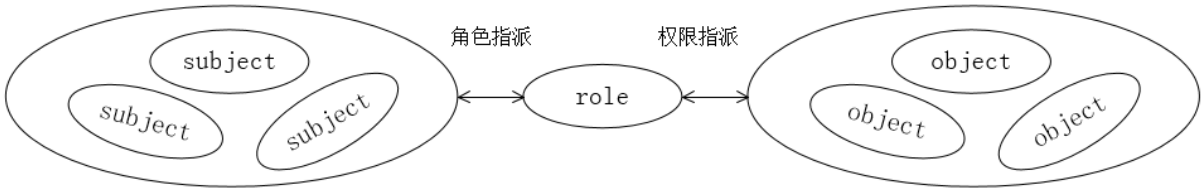


图 2.3: RBAC 模型

2.2.5 ABAC

ABAC 模型的核心思想是根据实体的属性作为授权依据的访问控制机制。对属性的定义较为宽泛，包括主体的属性、客体的属性、环境条件和访问上下文等。图2.4 是 ABAC 模型示意图。当主体发起对客体的访问请求时，决策引擎首先收集主体、环境和客体的相关属性。然后根据一定的策略规则决定是否授权。ABAC 模型支持布尔逻辑，如 IF,THEN 语句。每个规则（布尔逻辑）通过比较基于运算符的两个值（“是”或“不是”）和条件（“等于”，“小于”等）来产生真或假的结果。可见，ABAC 模型对于大型的权限管理系统能够提供更加灵活的和可扩展的方式。即便如此，本文并未采用 ABAC 模型来实现物联网访问控制系统。因为对于一个非大型的系统，维护大量的和复杂的属性策略规则的代价远高于其在灵活性和扩展性等方面带来的优势。

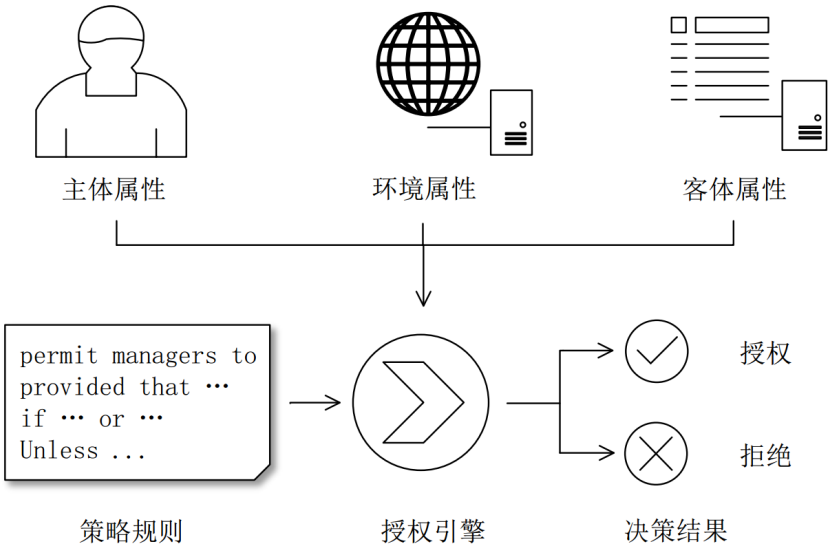


图 2.4: ABAC 模型

2.3　机器学习

2.3.1　基本概念

机器学习是人工智能的一个子集，其核心思想是利用经验和数据来到达既定的目的，而无需程序开发人员为待预测结果的每一个潜在条件进行编程。机器学习的目的是完成数据分析，通过分析数据发现某种自然模式，并从中获取有价值的信息。在数据分析领域，主要存在以下几种分析手段：

(1) 描述性分析：描述性分析是用于确定所发生的事情，即对某种现象的描述性报告。例如，用这个月的销售额与去年同期进行比较而产生结果。

(2) 特征性分析：特征性分析是用于解释现象发生的原因。这通常涉及使用带有 OLAP 技术的控制台用于分析和研究数据，根据数据挖掘技术来发现数据之间的相关性。

(3) 预测性分析：预测性分析用于评估可能发生的事情的概率。例如，预测性分析系统根据工作性质、个人兴趣爱好和年龄等因素刻画人物肖像，以便挖掘潜在的用户对象。

机器学习能够利用已存在的数据来对新数据产生原因的多种可能结果进行预测，因此机器学习非常适合于数据分析中的预测性分析。该预测过程实际上是循环和连续的，因为为了优化模型，通常会有更多的数据被添加进来。如图 2.6 所示是机器学习的工作流程示意图，总共分为目标定义、数据收集、数据准备、模型训练、模型评估和部署改进六大步骤：

(1) 目标定义：机器学习从一个明确的问题和清晰的目标开始，该目标能够为后续的步骤提供方向指导；

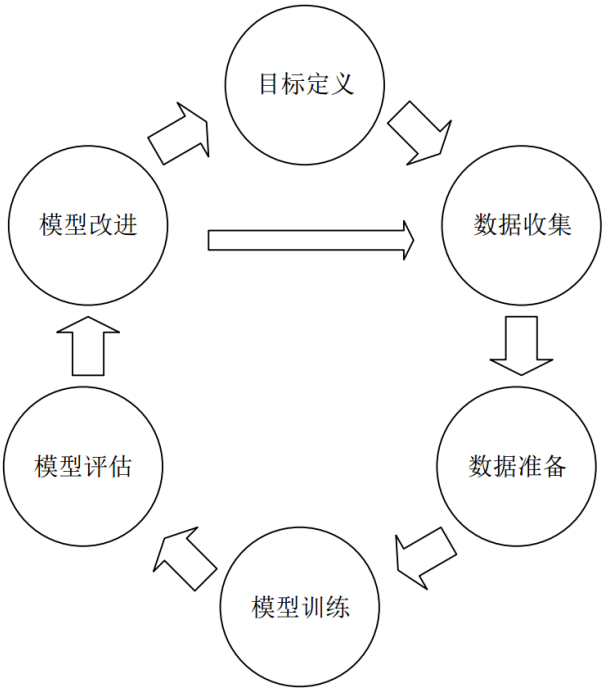


图 2.5: 机器学习工作流程

(2) 数据收集：在既定目标的指导下收集相关的数据。合适数据的体量和类型越多，最终训练得到的模型越接近现实。机器学习对数据的来源并无特别的要求，这些数据可以是来自电子表格、文本文件和数据库。

(3) 数据准备：数据准备包括数据的清理和解析，例如删除和纠正异常值（失控的错误值）。该工作会对最终模型的准确率产生显著的影响，同时这可能会占用机器学习总时间和工作量的 60% 以上。然后将数据集分成训练集和测试集两部分;

(4) 模型训练：模型训练涉及模型选择和数据训练两个过程。在模型选择过程中，应该根据既定目标和收集的数据情况选择适当的模型算法。例如，既定的目标属于分类问题还是回归问题；收集的数据是否带有标签；收集的数据体量情况。数据训练的目的是发现数据中的自然模式和相关性，或者用于预测；

(5) 模型评估：模型评估通过比相应的误差目标函数对训练的模型性能进行评估。常用的评估方法有留出法、交叉验证法和自助法等。而常见的性能度量指标有准确率、精度和召回率等。

(6) 模型改进：模型改进的方法通常有两种，一种是可以尝试完全不同的算法模型，另一种是通过收集更多种类和体量的数据来优化现有的模型;

到目前为止，在机器学习领域的研究者们已经提出了大量的算法模型。图2.6 表明这些算法模型大致可以分为监督学习、非监督学习和强化学习。其定义如下：

(1) 监督学习：有监督学习是一种从有标记的训练数据中推导出预测函数的学习模型。有标记的训练数据是指每一个数据实例都包括输入和期望的输出；

(2) 无监督学习：和有监督学习相反，无监督学习是从无标记的训练数据中推导出预测函数。最典型的无监督学习就是聚类分析，它可以在探索性数据分析阶段用于发现隐藏的模式或者对数据进行分组；

(3) 强化学习：强化学习的核心思想来源于行为主义理论，即物体或机器如何在环境的刺激和反馈下做出适当的回应。当回应满足预期时，则说明该物体或机器已经学习到了某种知识;

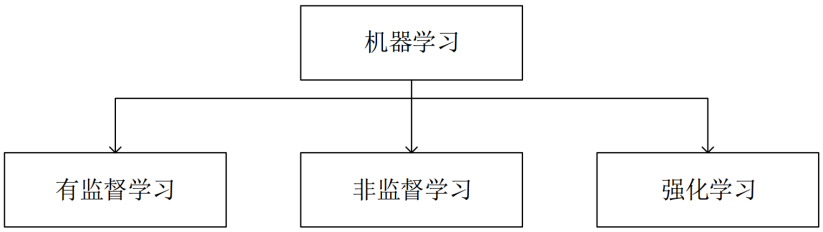


图 2.6: 机器学习分类

虽然本文主要利用改进的支持向量机来训练设备指纹识别模型，但是还选择了其他机器学习算法进行了相同的工作，以对比算法性能。因此本节接下来将简单介绍上述涉及的机器学习算法。

2.3.2　支持向量机

支持向量机的基本思想是求解能够正确划分样本数据集，并且使得几何间隔最大化的超平面。如图2.7 所示，$w \cdot x + b = 0$ 即为该超平面。同时可以发现，对于一个线性可分的数据集，这样的超平面存在无穷多个（即感知机）。但是几何间隔最大的超平面却只存在一个，这也是支持向量机区别于感知机的关键。

在 n 维实数空间 $\mathbb{R}^n$ 中，超平面的数据定义如式2.1 所示：

$$w^T x + b = 0 \tag{2.1}$$

其中 $w = (w_1, w_2, w_3, ..., w_n)$。根据点到平面的距离公式，空间中任意一点 x 到超平面的距离为：

$$\gamma_i = y_i(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|}) \tag{2.2}$$

因此支持向量机的求解过程就是当超平面满足样本集正确划分的约束条件下，使得 r_i 最小化：

$$\gamma = \min_{i=1,2,..,N} \gamma_i \tag{2.3}$$

上述的约束条件为：

$$TA = \begin{cases} w^T x + b > 0, & y_i = +1 \\ w^T x + b < 0, & y_i = -1 \end{cases} \tag{2.4}$$

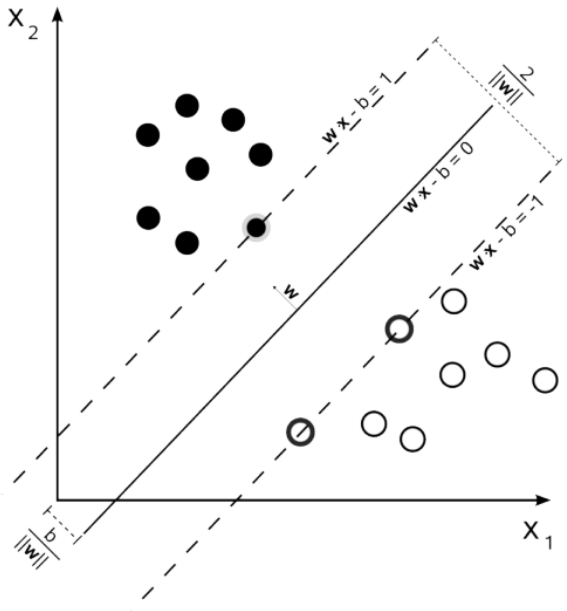


图 2.7: 支持向量机原理图

2.3.3 K 邻近算法

K 邻近算法（KNN）是一种基本的分类和回归方法。在分类问题中，KNN 算法假设给定的训练集的实例类别已经确定。K 值的选择、距离度量方式已经分类决策规则是 KNN 的三个基本要素。

KNN 中的距离度量方式通常采用欧式距离，但也可能是更为一般的 L_p 距离。设特征空间 X 是 n 维数据向量空间 $\mathbb{R}^n$，$x_i, x_j \in X$，$x_i = (x_i{}^{(1)}, x_i{}^{(2)}, ..., x_i{}^{(n)})^T$，$x_j = (x_j{}^{(1)}, x_j{}^{(2)}, ..., x_j{}^{(n)})^T$，那么 x_i，x_j 的 L_p 距离定义为：

$$L_p(x_i, x_j) = (\sum_{i=1}^{n} |x_i{}^{(i)} - x_j{}^{(j)}|^p)^{\frac{1}{p}} \tag{2.5}$$

当 $p = 2$ 时，即为欧式距离：

$$L_2(x_i, x_j) = (\sum_{i=1}^{n} |x_i{}^{(i)} - x_j{}^{(j)}|^2)^{\frac{1}{2}} \tag{2.6}$$

当 $p = 1$ 时，即为曼哈顿距离，即：

$$L_1(x_i, x_j) = (\sum_{i=1}^{n} |x_i{}^{(i)} - x_j{}^{(j)}|) \tag{2.7}$$

当 $p = \infty$ 时，即各个坐标距离的最大值：

$$L_{\infty}(x_i, x_j) = \max_i |x_i^{(i)} - x_j^{(j)}| \tag{2.8}$$

k 值的选择会对算法最终结果产生严重的影响。当取较小的 k 值时，会降低训近似误差，即只有与待预测数据邻近的样本会对结果产生作用。但也容易增大估计误差，因为预测结果会对带预测数据邻近的样本非常敏感，造成过拟合。相反，如果取较大的 k 值时，会增大近似误差，而降低估计误差。这使得模型变得简单，容易导致欠拟合。

KNN 的分类决策规则比较简单，往往采用多数表决方法。即在预测阶段，根据待预测数据的 k 个最近邻样本的类别，并通过投票的方式对结果进行预测。获得票数最多的类别即为最终的结果。

2.3.4　Adaboost

Adaboost 是一种集成学习模型。机器学习的目标是学习出一个稳健的，且在各方面表现良好的模型。但是在实际情况下往往只能得到多个有偏好的模型，即在某些方面表现良好，称其为弱分类模型。集成学习的目的就通过组合多个弱分类模型以期得到一个更好的更全面的强分类模型。其核心思想是通过部分弱分类模型来纠正剩余的弱分类模型。图 2.9 是集成学习的一般结构。

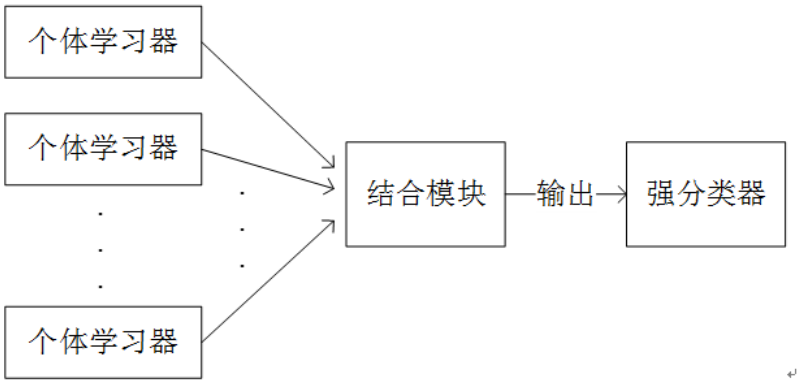


图 2.8: 集成学习的一般结构

Adaboost 是属于 Boosting 类型的集成学习模型。Boosting 的核心思想是在学习过程中调整样本数据的权值分布来得到一系列弱分类器。其权值调整原则是提高哪些被前一轮弱分类错误分类的样本的权值，从而降低哪些被正确分类样本的权值；加大分类错误率小的弱分类其的权值，使其在表决中起较大的作用；减小分类误差率大的弱分类器的权值，使其在表决中起较小的作用。图 2.10 所示是 Boosting 框架图。

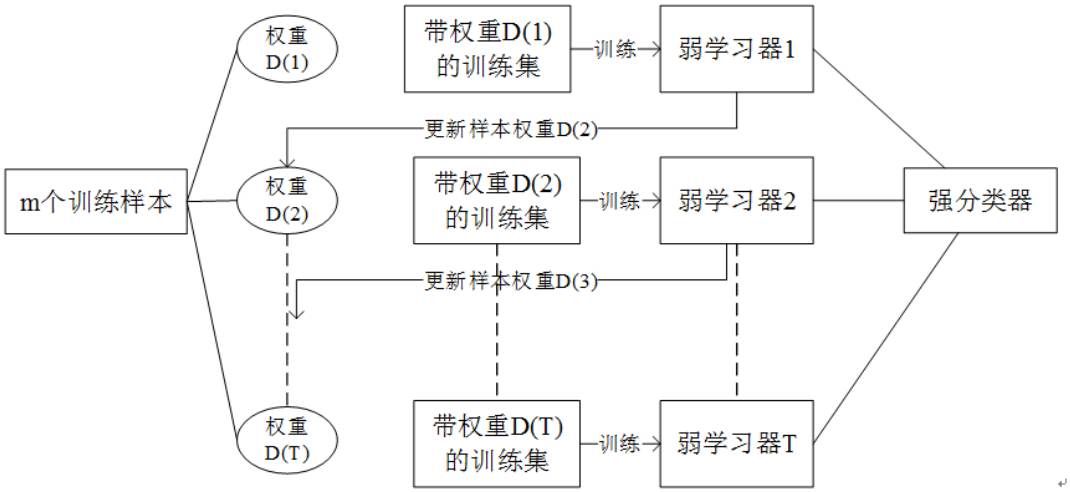


图 2.9: Boosting 框架

2.3.5 随机森林

随机森林也是一种集成学习模型，它属于 Bagging 类型。Bagging 也叫自举汇聚法（Bootstrap Aggregating），是一种在原始数据集上通过有放回抽样重新选出 k 个新数据集来训练分类器的集成技术。它使用训练出来的分类器的集合来对新样本进行分类，然后用多数投票或者对输出求均值的方法统计所有分类器的分类结果，结果最高的类别即为最终标签。其算法流程如图 2.10 所示。随机森林将决策树作为弱分类器。对于随机森林中的每棵树，其选取特征的方式是随机的，这对降低模型的方差很有作用。因此随机深林通常不需要进行额外的剪枝，即可以取得较好的泛化能力和抗过拟合能力。

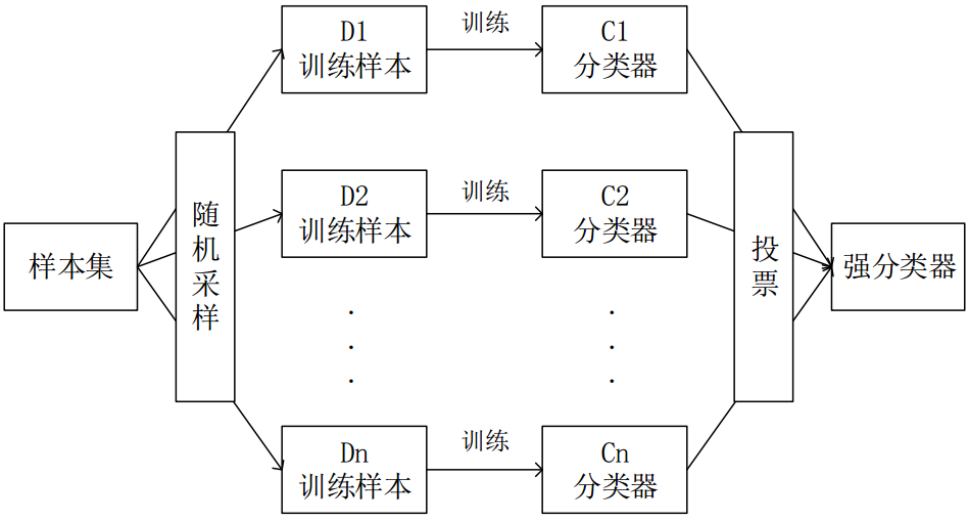


图 2.10: Bagging 框架

2.4 模糊集

2.4.1 基本概念

在经典集合论中，对于集合 A 和元素 a，两者的关系定义明确，不外乎 $a \in A$ 或者 $a \notin A$，不允许存在模棱两可的情况。这限定了经典集合论的应用范围，即无法研究具有模糊性质的事物。模糊性客观存在于现实中，例如，“大与小”，“多与少”，“好与坏”等，这些概念没有明确的界限。于此同时，模糊性也长期存在于控制、决策、智能等工程领域，而传统的以经典集合论为基础的数学方法无法胜任这些问题。基于上述问题，Zadeh 提出了模糊数学理论。模糊数学理论利用隶属函数量化模糊性问题，将其转化到精确数值研究领域。例如在经典集合论中，元素 x 和集合 A 的隶属关系如下：

$$\mu(x) = \begin{cases} 1, & a \in A \\ 0, & a \notin A \end{cases} \tag{2.9}$$

其中 $\mu(x)$ 为隶属函数。模糊数学将二值逻辑 {0,1} 扩展到 [0,1] 区间内任意的无穷多个值的连续逻辑。模糊集的相关定义如下：

定义 **3.1** 模糊集：论域 U 到区间 [0,1] 的映射：

$$\mu_A : U \to [0,1], u \mapsto \mu_A(u) \in [0,1] \tag{2.10}$$

定义 3.2 并集：模糊集 A 和模糊集 B 并集 $\mu_{A \cup B}$：

$$\mu_{A \cup B} = max\{\mu_A(u), \mu_B(u)\} \tag{2.11}$$

定义 3.2 交集：模糊集 A 和模糊集 B 交集 $\mu_{A \cap B}$：

$$\mu_{A \cap B} = min\{\mu_A(u), \mu_B(u)\} \tag{2.12}$$

定义 3.4 补集：模糊集 A 的补集 $\mu_{A^-}(u)$

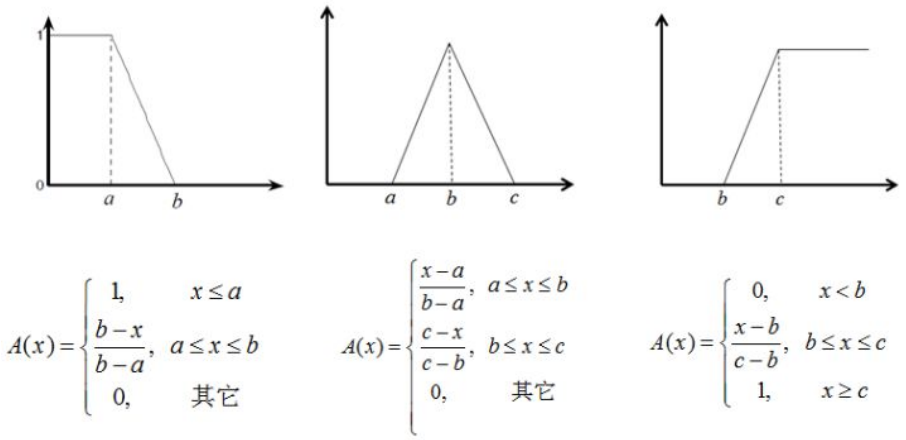
$$\mu_{A^-}(u) = 1 - \mu_A(u) \tag{2.13}$$

定义 3.5 模糊算法：模糊集 A 和模糊集 B 通过模糊算子合成多种运算：

$$\begin{aligned} A \vee *B &= \mu_A(u) \vee *\mu_B(u) \\ A \wedge *B &= \mu_A(u) \wedge *\mu_B(u) \end{aligned} \tag{2.14}$$

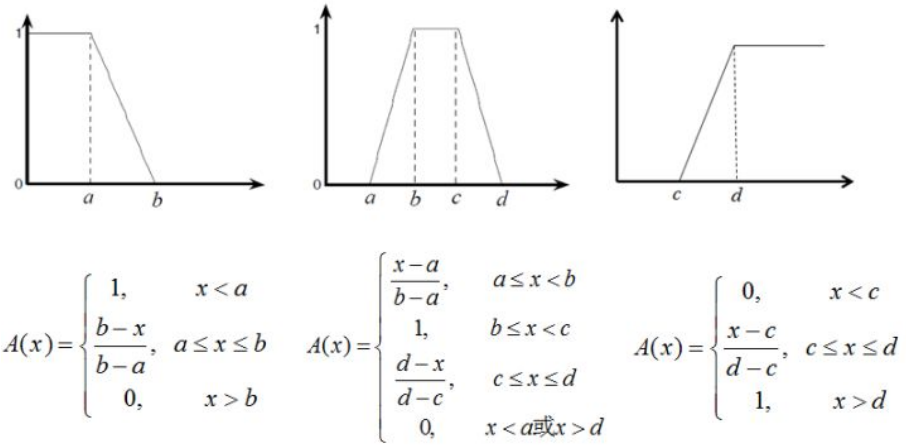
2.4.2 隶属函数

隶属函数是将模糊问题转化为精确数值研究的关键，通常可以凭借直觉或者通过二元对比排序、模糊统计实验和最小模糊度等方法得到隶属函数。常见的隶属函数如图2.11所示。



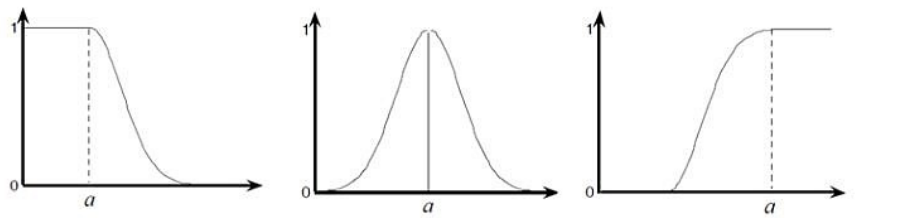
$$A(x)=\begin{cases}1, & x\le a\\ \dfrac{b-x}{b-a}, & a\le x\le b\\ 0, & \text{其它}\end{cases}\qquad A(x)=\begin{cases}\dfrac{x-a}{b-a}, & a\le x\le b\\ \dfrac{c-x}{c-b}, & b\le x\le c\\ 0, & \text{其它}\end{cases}\qquad A(x)=\begin{cases}0, & x<b\\ \dfrac{x-b}{c-b}, & b\le x\le c\\ 1, & x\ge c\end{cases}$$

(a)



$$A(x)=\begin{cases}1, & x<a\\ \dfrac{b-x}{b-a}, & a\le x\le b\\ 0, & x>b\end{cases}\qquad A(x)=\begin{cases}\dfrac{x-a}{b-a}, & a\le x<b\\ 1, & b\le x<c\\ \dfrac{d-x}{d-c}, & c\le x\le d\\ 0, & x<a\text{或}x>d\end{cases}\qquad A(x)=\begin{cases}0, & x<c\\ \dfrac{x-c}{d-c}, & c\le x\le d\\ 1, & x>d\end{cases}$$

(b)



$$A(x)=\begin{cases}1, & x\le a\\ e^{-(\frac{x-a}{\sigma})^2}, & x>a\end{cases}\qquad A(x)=e^{-(\frac{x-a}{\sigma})^2},\quad -\infty<x<\infty\qquad A(x)=\begin{cases}e^{-(\frac{x-a}{\sigma})^2}, & x\le a\\ 1, & x>a\end{cases}$$

(c)

图 2.11: 常见隶属函数

2.5 本章小结

本章主要对基于设备指纹和行为可信的物联网访问控制系统的技术背景进行阐述。首先研究了访问控制技术，并对常见了访问控制模型进行介绍，包括 DAC、MAC、RBAC 和 ABAC 模型。其次研究了机器学习算法。在本文的访问控制系统中，机器学习算法被应用于设备指纹识别。最后研究了模糊集相关理论，因为基于模糊集的模糊综合评价法被应用于本文系统中的设备行为可信度计算。

第三章　基于 TSMC-SVM 的设备指纹识别算法

3.1 引言

随着物联网的快速发展，智能终端开始大规模接入网络，这对网络安全访问控制提出了严峻的挑战。虽然这些设备提出了更加新颖的概念，提供了更加便捷的服务，但是经常忽视安全问题。这其中的部分原因是很多物联网设备生产商是从传统的电器制造商转型过来的，因此缺乏对计算机安全技术的积累。有的生产商甚至不提供设备的固件，软件升级和漏洞补丁。这些安全问题本可以通过现有的安全认证机制缓解，例如使用复杂加密协议和安装认证代理软件。但是上述复杂的安全认证机制无法适用于计算资源和存储资源匮乏的物联网设备。基于此，本文研究了设备的网络流量特征，提出了一种设备指纹技术用于设备的安全认证。

设备指纹技术是通过分析设备的网络流量特征来发现设备在硬件、固件和软件上的差异。本文用于分析的流量是通过被动监听得到，因此对设备是非侵入的，而且对设备的计算和存储能力没有任何要求。理论上，设备指纹就像生物指纹一样，具有唯一性，能够作为设备身份的标识。遗憾的是，本文的设备指纹并不具有正真意义上的唯一性，即本文的设备指纹只能识别设备的类型。造成该问题的原因是同型号设备在网络上表现出极其相似的行为。但是设备类型指纹已经为本文的物联网访问控制系统提供了重要的设备身份信息。而对于如何在设备类型信息的基础上实现对设备个体的访问控制管理，将在第四章中介绍。第四章将会提出一个基于行为可信度的访问控制模型，该模型能够根据设备个体的网络行为实现对权限分配关系的动态调整。

机器学习作为一门涉及概率论、统计学、算法复杂度理论的多领域交叉学科，适合在大规模数据中发现隐含的规律和模式。其中 SVM 是一种成熟并被广泛使用的分类算法。其核心思想是通过非线性映射把原数据集中的向量点转化到更高维度空间中，并在这个高纬度空间中寻找一个线性的超平面。同时，通过组合多个基于 SVM 的二分类器可以方便地实现多分类模型（OvA-SVM）。同时该模型具有很强的可扩展性。但是由于分类重叠问题的存在，即模型在分类过程中有多个二分类器的输出结果为正值。这限制了其可扩展性。基于此，本文提出了一种基于 OvA-SVM 的二阶段多分类模型（Two Stage Mulit-classification，TSMC-SVM）。该模型通过引入修正余弦相似度理论，有效解决了 OvA-SVM 模型的分类重叠问题。

3.2 特征提取

设备指纹技术可以分为主动式设备指纹技术和被动式设备指纹技术，其区别在于获取网络流量的方式。主动式设备指纹技术通过发送精心构造的探针，进而分析其响应报

文来提取设备的信息。例如 Namp 和 Xprobe 是两款业界著名的主动式设备指纹分析工具。而被动式设备指纹技术是通过在合适的网络节点被动监听和捕获设备的网络流量，进而分析出设备的指纹信息。很明显，被动式设备指纹具有更低的侵入性，这也是本文选择该指纹技术作为访问控制模型中的认证技术的原因。本节接下来会从时间窗口选择和特征提取和编码两个视角来详细介绍特征提取的过程。其中时间窗口选择的目的是希望能够找到一个时间窗口，在该窗口内，设备的具有稳定的网络行为。

3.2.1　时间窗口选择

对于设备指纹，有两个问题值得重视，即稳定性和可行性。稳定性描述的是设备指纹不应受外界因素的干扰。而可行性描述的是设备指纹的生成应该切实可行，主要关注的是计算的复杂性。

随着时间的变迁和任务的调整，设备会表现出完全不同的网络行为。因此设备的网络行为具有很大的不确定性。这将影响指纹的稳定性和可行性。但是从另一个视角，设备的整个工作过程可以被分为两个阶段：启动阶段和服务阶段。启动阶段是设备从上电到完成各项硬件和软件配置的过程。而服务阶段是设备在完成启动之后进入正常的工作服务。在服务阶段，设备的网络行为容易受到环境上下文变化的影响，例如网络管理员更改了网络环境的配置，对端通信实体因为某些因素发生了行为的改变。或许这些看似随机的行为在一定的时间周期内能够表现出某种重复但是具有唯一性的行为特征。但是这个时间周期的长度得不到保证。例如 []，通过分析 10^5 个报文才分析出了 AP 存在一种重复而独特的行为模式。这不符合设备指纹的可行性要求。此外，访问控制为保证授权的实时性，不应该经历很长的时间窗口。

物联网设备由于功能单一，资源受限，很多的配置项会通过简单的方式固化在硬件或者软件中。这些糟糕的设计可能从某种角度加大了开发者维护和迭代固件和软件的难度，但是却为设备指纹提供了一个稳定的流量时间窗口，即启动阶段。启动的工作是将设备从断电的状态推送到正常服务的状态，主要工作就是加载软件和相关配置项。该过程是具有严格的加载顺序的，因为加载项之前存在依赖关系。就像复杂的 Linux 操作系统启动一样，先是基本输入输出系统（BIOS）检查硬件状态，然后加载文件系统中的主引导记录（MBR），然后再有 MBR 加载操作系统的自举程序，最后由自举程序完成对操作系统的加载。此外，启动阶段的稳定性还表现在所经历的时间窗口的长度。设备启动所经历的时间是相对固定的，而且由于物联网设备的系统是轻量级系统，其开机时间是很短的。基于上述原因，本文选择提取设备启动阶段的流量特征用于生成指纹。

3.2.2　特征提取与编码

本节将详细介绍设备网络流量的特征提取和编码。为更加全面地刻画设备的网络行为，本文分别提取了流量的报文特征和统计特征。报文特征主要是报文的相关字段，反映的是流量的局部细节信息，而统计量用于展示流量的全局信息。

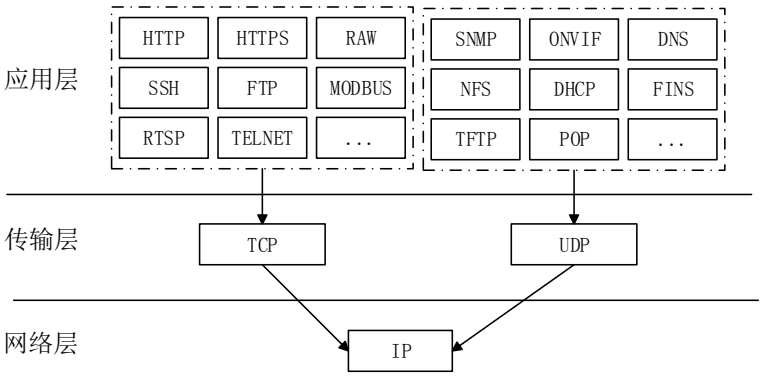


图 3.1: 网络协议框架

表 3.1: 不同物联网设备之间报文字段的差异

特征量	TOS	TTL	DF	MSS	OPT
PLC SIEMENS	0	30	0	14660	M
PLC Schneider	0	254	1	1452	M
Printer Lexmar	192	65	0	-	-
Camera Linksys	0	64	0	1380	M
DVR Dahua	28	64	1	-	-
Router Cisco	0	255	0	536	M

1) 报文特征：图 3.1 描述的是物联网设备的网络协议栈架构，主要包括网络层、传输层和应用层（本文的特征提取并不涉及物理层和链路层）。在网络层，本文研究的是不同类型设备在 IP 报文头部字段之间的差异。在网络层，不同类型的设备在 IP 报文头部字段会存在明显的差异，例如 Version 字段在报文头部中占 4bit，描述的是 IP 协议版本。0100B 为 IPv4，0110 为 IPv6。占 1bit 的 DF 字段描述的是报文是否分片，1 表示不分片，0 表示分片。其它还有 TTL，PORT，OPTION 都会应该设备类型的不同而存在差异。其中文献 [46] 在做过大量的研究工作。表 3.1 是研究人员在分析大量物联网设备的基础上发现不同类型设备在报文字段上的差异。这些设备涵盖了物联网的很多领域：PLC SIEMENS 和 PLC Schneider 属于工业物联网设备，Printer Lexmar，Camera Linksys 和 DVR Dahua 属于普通消费类物联网设备，而 Router Cisco 则属于网络传输类设备。可以发现物联网设备在这些中有明显不同的值。基于上述描述，本文在文献 [46] 工作的基础上，增加了其它报文头部字段，并对这些字段特征进行了编码。表 3.1 是本文选取的字段特征及其编码表。接下来对字段特征的编码规则展开详细的介绍。 其中 $VERSION$ 字段用于区分 IP 协议版本，$IPv4$ 编码为 0，$IPv6$ 编码为 1。$NET\_PROTOCOL$ 表示的传输层协议，考虑到 TCP 和 UDP 的使用广泛的传输层协议，因此本文将协议分为三类，TCP 编码为 0，UDP 编码为 1，其它协议编码为 2。TTL 指定是 IP 报文的生存时间，即 IP 报文被路由器丢弃之前允许通过的最大网络跃点数。其最常见数值的 64，因此本文的

表 3.2: 报文字段特征与编码

特征量	编码
version	0/1
ttl	0/1/2
df	0/1
net_proto	0/1/2
net_has_opt	0/1
trans_has_opt	0/1
port	0/1/2

表 3.3: 物联网协议

类型	应用层协议
TCP-based	http,https,ssh,ftp rtsp,telnet,raw
UDP-based	snmp,onvif,dns,nfs dhcp,tftp,pop

编码规则是当 TTL 小于 64 编码为 0，等于 64 编码为 1，大于 64 编码为 2。在 IP 报文中，OPTION 字段是可选项，是预留给开发者的自定义字段。本文提取的 NET_HAS_OPT 字段特征用于表征 IP 报文中是否存在 $OPTION$ 选项。提取方式是检查 IP 报文头部长度是否大于 20，若是，则编码为 1，否则，编码为 0。同理 $TRANS\_HAS\_OPT$ 分别用于表征 TCP/UDP 报文是否存在 $OPTION$ 选项. 最后，对于 $PORT$ 字段，依据 $IANA$（英特网已分配数值权威机构）对端口的划分，周知端口 $(0 \sim 1023)$，注册端口 $(1024 \sim 49151)$ 和动态端口 $(49152 \sim 65535)$ 分别编码为 0，1 和 2。

表 3.2 中的网络层和传输层特征提取自单个独立的报文，因此对于一段报文序列，可以生成特征矩阵 F1：

$$M = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{7,1} & f_{7,2} & f_{7,3} & \cdots & f_{7,n} \end{bmatrix} \tag{3.1}$$

其中 n 代表报文数量。将其转化向量形式，可以得到 $7n$ 维特征向量 $F2$:

$$F1 = (f_{1,1}, f_{1,2}, ..., f_{1,n}, ..., f_{7,1}, f_{7,2}, ..., f_{7,n}) \tag{3.2}$$

在应用层，设备生产商会依据设备提供的服务类型，使用不同的协议来承载相应的服务。表 3.2 是本文的指纹涉及的 15 种应用层协议，包括基于消费者物联网协议和跨行业物联网协议。考虑到特征向量的维度过长会影响模型的训练和识别，因此应用层特征的提取方式不同于网络层和传输层。通过分析报文序列中是否存在表 3.2 中的协议来生成特征向量，向量元素的编码值为 0 和 1。因此对于一段报文序列，可以得到一个 14 维

表 3.4: 网络流量统计特征

特征量	计算方法
最大值	Max
最小值	Min
均值	$\frac{1}{n}\sum_{i=1}^{n} x_i$
方差	$\frac{1}{n}\sum_{k=1}^{n}(x_i-\bar{x})^2$
标准差	$\sqrt{\frac{1}{n}\sum_{k=1}^{n}(x_i-\bar{x})^2}$

的特征向量 F3:

$$F1 = (f_1, f_2, f_3, ..., f_{14}) \tag{3.3}$$

例如，网络摄像头通常会运行 ONVIF 协议，该协议是一个网络视频的全球性开放接口标准，致力于不同厂商生产的网络视频产品的互通性。此外，为了方便管理，产商还会为其部署 SNMP，SSH,TELNET 等协议。因此，对于该产品生成的应用层特征向量 F2:

$$F1 = (0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0) \tag{3.4}$$

2) 流量统计特征：流量的统计量被广泛应用于网络管理和异常检测。文献 [] 通过分析什么样的数据集给出了一系列的统计特征用于研究网络流量。随后大量的研究人员 [] 借助于这些统计特征实现了对网络流量类型的识别。基于此，本文研究了设备网络流量的统计特征，使用的统计量如表 3.4 所示。

本文将表 3.5 中的统计量应用于 IP 报文长度和 IAT。其中 IAT（ineral-arrival time，Δt）表示的连续报文的到达时间间隔。因此，可以得到流量统计特征向量 F_3：

$$F1 = (f_{len_1}, f_{len_2}, ..., f_{len_5}, f_{iat_1}, f_{iat_2}, ..., f_{iat_5}) \tag{3.5}$$

基于上述的特征提取过程，可以得到最终的特征向量 F:

$$F = (F_1, F_2, F_3, Y) \tag{3.6}$$

其中 Y 代表设备类型标签。

3.3 二阶段多分类模型

本节将利用具体的二分类算法来介绍 TSMC 模型。虽然 TSMC 模型并不局限于某种特定的二分类算法，但考虑到 SVM 算法原理简单且被广泛使用，本节将基于 SVM 算法来介绍 TSMC 模型，因此称该模型为 TSMC-SVM。选择 SVM 算法的另一个原因是在实验仿真的结果显示基于 TSMC-SVM 算法的性能明显优于基于其它机器学习算法构造的 TSMC 模型。

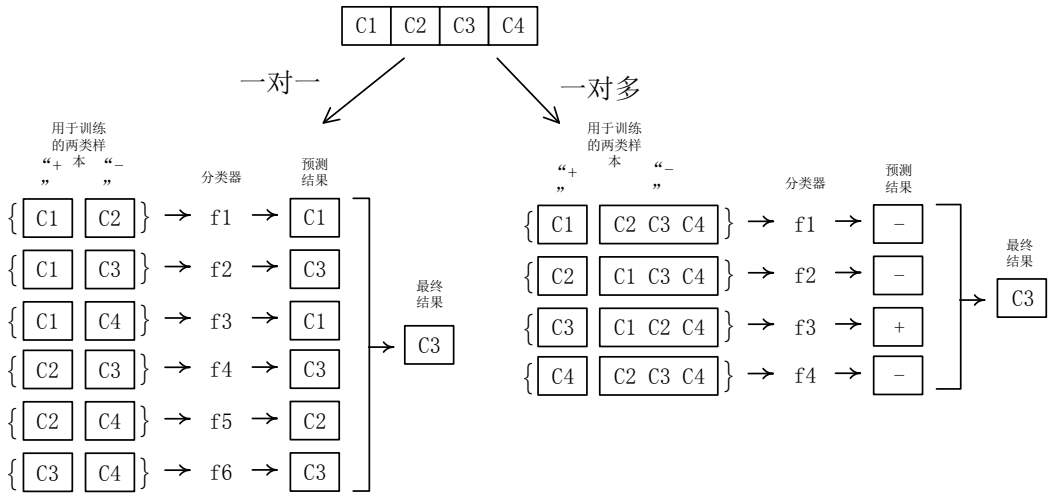


图 3.2: “一对一”和“一对多”模型示意图

3.3.1　SVM 多分类模型

SVM 算法最初是为二分类问题设计的，目前基于 SVM 构造多分类器主要有两种方式。第一种方法是直接构造目标函数，将求解多个分类面的参数合并到单个最优化问题中。第二种方法通过组合多个二分类器以实现多分类器的构造。前者的计算复杂度高，实现困难；而后者基于分治策略将多分类问题分解成多个二分类的问题，通过策略组合二分类器实现最终结果的分类。其中最为常见的策略是“一对多”,“一对一”和“多对多”。

1) 一对多：一对多模型是最早被广泛使用的多分类策略。其核心思想是在训练阶段依次将某一类别的样本归为正样本集，而其他剩余样本自动归为负样本集。这样对于拥有 k 种类型的样本集可以构造出 k 个二分类器。在测试阶段，每一个二分类器都会根据训练样本计算出判别结果。当有一个二分类器的判别结果为正值，则最终的类型为该二分类器所属的类型。该模型的优点是分类速度快，对于分类 K 种类型只需要构造 K 个二分类器。同时具备高可扩展性。当需要增加新的类型时，只需要为新增类型训练新的二分类器，无需重新训练整个分类算法。但是该模型也存在其缺点，即当测试结果中存在多个二分类器的判别结果为正值时，无法确定最终的判别结果。该现象被称为分类重叠现象。

2) 一对一：一对一模型又被称为成对组合模型。其核心思想是在训练阶段将样本集中的任意两个不同类别的样本作为正样本集和负样本集，并为其构建二分类器。因此对于拥有 K 种类型的样本集需要构造 $(k-1)k/2$ 个二分类器。可见该模型的缺点是随着样本集类别的增加，所需的二分类器数量将 n^2 增长。

一对一模型的最终类型判别方法一般采用权重投票策略。根据一对一模型的（k-1）

k/2 个二分类器的输出结果，可以构造一个矩阵 R：

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,k} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ r_{k,1} & r_{k,2} & \cdots & r_{k,k} \end{bmatrix} \tag{3.7}$$

其中 $r_{i,j} \in [0,1]$ 表示对于类型 i,j 的二分类器的识别结果。可以发现 $r_{i,j} = 1 - r_{j,i}$。当构建完矩阵 R 时候，就可以有多种策略来计算最终的识别结果，例如最常见的就是权重投票策略。策略的公式定义如下：

$$class = \arg\max_{i=1,..,k} \sum_{1 \le i \ne j \le k} r_{i,j} \tag{3.8}$$

3) 多对多：多对多模型依次将多个类作为正样本集，多个类作为负样本集。可见，“一对多”和“一对一”模型是“多对多”模型的特例。“多对多”模型的正例和负例的构造是需要遵守相应的设计原理的，例如最常用的是技术是“纠错输出码”。由于本文并不涉及“多对多”模型，因此不展开介绍。

从上述的介绍可以发现，“一对多”模型能够通过将二分类器组合快速设计出多分类器。此外，基于模型设计的多分类器具有高可扩展性的特点。当任务中需要增加分类的分支时，只需要为该分支单独训练一个二分类器，然后组合到原多分类器中即可。但是，该多分类模型也存在相应的问题，其中最值得关注的是分类重叠现象，即有多个二分类器的输出结果为正值，模型无法在这些类型之间进一步的识别。基于此，本文提出了 TSMC-SVM 模型，该模型能够有效解决“一对多”模型的分类重叠问题。

3.3.2 TSMC-SVM 模型

TSMC-SVM 模型的核心思想是将余弦相似度理论引入到上述的“一对多”模型中。图 3.4 描述了 TSMC-SVM 模型的二阶段结构。在第一阶段中，从流量中提取的特征向量 F 被输入到上文介绍“一对多”模型中，其中每一个二分类器用于识别一种设备类型。因此对于有 K 个分支的多分类任务只需要训练 K 个二分类器。当只有一个二分类器 Type(i) 的输出结果为正值时，那么就判定 F 属于类型 i。如果有多个二分类器的输出结果为正值时，模型会进入第二阶段，触发余弦相似度匹配模块。该模块计算的是特征向量 F 与上阶段中输出结果为正值类型所属的样本特征向量之间的余弦相似度，以实现进一步的类型识别。将其定义为特征向量 F 与样本特征向量集中元素的平均余弦相似度。例如设备类型 i 的样本特征向量集为 $(s_{i,1}, s_{i,2}, ..., s_{i,m})$。那么平均余弦相似度的公式定义如下：

$$similarity = \frac{1}{m} \sum_{j=1}^{m} \frac{F \cdot S_{i,j}}{\|F\| \cdot \|S_{i,j}\|} \tag{3.9}$$

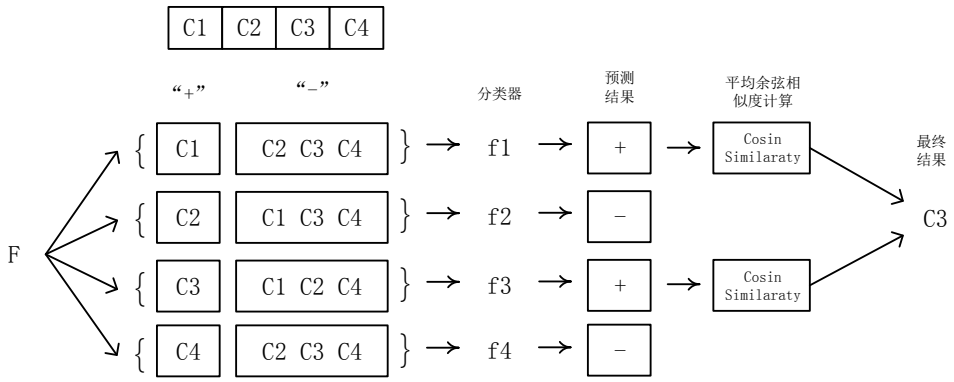


图 3.3: TSMC-SVM 模型

其中 $s_{i,j}$ 表示设备类型 i 样本集中的第 j 个元素，m 表示样本个数。最后选择平均余弦相似度最大的设备类型作为最终的识别结果。可以，TSMC-SVM 模型的原理不难理解。但是对于上述过程还有几个问题亟待解决。1) 不可分现象：上文介绍了 TSMC-SVM 第一阶段输出的两种情况，分别是只有一个二分类器的输出结果为正值和有多个二分类器的输出结果为正值。但是其实还存在另一种特殊情况，所有的二分类器的输出结果为负值，即类型不可分。幸运的是，这种情况在本文的访问控制系统中得到了很好的解释。当类型不可分时，表明该设备对于当前模型是未知设备类型。对于一台未知的陌生设备，系统应该拒绝其任何访问资源的请求。因此本文将其直接作为异常设备处理。

2) 余弦相似度的量纲偏差：余弦相似度是向量的夹角的余弦值来度量其个体之间的差异。其公式定义如下：但是余弦相似度更多关注的是向量方向上相似度，而对向量的绝对数值不敏感，这会导致度量误差。假设设备 d1 和 d2 的特征向量 (min_ip_payload，max_ip_payload) 分别为 d1=（1,2）和 d2=（4,5）。根据公式（3）计算得到的 d1,d2 的余弦相似度为 0.98，表明两者极为相似。但从特征上可以明显发现设备 d2 的 IP 报文载荷明显比设备 d2 的长。产生这种误差的根本原因是余弦相似度仅考虑向量维度方向上的相似而忽略了各个维度的量纲误差。

虽然本文在特征提取的时候对原始特征进行了最大最小值归一化处理，将有量纲数据转变成了无量纲数据。但是这种去量纲方式并不能弥补余弦相似度的上述误差。假设 d1,d2 经过归一化后得到 dn1=(0,1)，dn2=(0，1)。很明显 dn1，dn2 的余弦相似度变成了 1。可见，最大最小值归一化并没有对降低上述误差产生任何贡献。最大最小值归一化主要关注的是在向量方向的伸缩变换，使得数据统一映射到区间 [0,1]。这对于向量之间的夹角的改变影响甚微。而余弦相似度度量的是向量方向上的相似度，即向量之间的夹角。不同于最大最小值归一化，修正余弦相似度从去中心化的角度来实现对量纲偏差的矫正。其核心思想是在计算相似度之前对每个维度减去它们的均值。例如对于上述的向量 d1 和 d2，它们的 min_ip_payload 和 max_ip_payload 的均值都为 3，修正后 da1=(-2,-1)，da2=（1,2），那么 d1,d1 的余弦相似度为-0.8。这更加符合了设备 D1 和 D2 的 IP 报文载

荷差异的事实。3) 时间复杂度优化：余弦相似度匹配模块中的平均余弦相似度是对待识别特征向量 F 和样本特征向量集的每一个元素计算余弦相似度后取平均。对于一个 n 维的特征向量，假设某一类型的样本集大小为 n 个样本，那么其计算平均余弦相似度的时间复杂度为（n*m）。对此，本文提出了一种对样本预处理方法，使得计算平均余弦相似度的时间复杂度降为（n）。为描述该样本预处理方法以及其降低时间复杂度的原理，本节将重新给出余弦相似度的完整定义。首先定义 S 为某一类设备的样本集，其中 Sj 表示一个样本特征向量。而 F 表示待识别设备的特征向量：

$$\begin{aligned} S &= \{S_1, S_2, S_3, ..., S_j, ..., S_m\}, \\ S_j &= (s_{j_1}, s_{j_2}, s_{j_3}, ..., s_{j_i}, ..., s_{j_n}), \\ V &= (v_1, v_2, v_3, ...v_i, ..., v_n) \end{aligned} \tag{3.10}$$

其中 n 表示提取的特征数量，m 表示样本集中样本的数量。因此平均余弦相似度定义如下：

$$similarity = \frac{1}{m}\sum_{j=1}^{m}\frac{V \cdot S_j}{\|V\| \cdot \|S_j\|} \tag{3.11}$$

其中 $\|V\|$,$\|S_j\|$ 和 $V \cdot S_j$ 定义如下：

$$\begin{aligned} \|V\| &= \sqrt{\sum_{i=1}^{n} {v_i}^2}, \\ \|S_j\| &= \sqrt{\sum_{i=1}^{n} {s_{j_i}}^2} \\ V \cdot S_j &= \sum_{i=1}^{n} v_i s_{j_i} \end{aligned} \tag{3.12}$$

根据上述定义可以得到平均余弦相似度的计算总共需要 $(3n+2)m$ 次乘法运算，$(3n+1)m$ 次加法运算和 1 次除法运算。如果将 $\|v\|$ 进行存储，那么可以减少 nm 次乘法运算和 nm 次加法运算。因此计算平均余弦相似度的时间复杂度为 $\mathcal{O}(nm)$。一般而言，在分类任务中，特征向量维度 n 是相对固定的。并且有时候为了优化算法，还会对特征进行约简。而对于参数 m 的变化趋势则正好相反。样本集会被不断地扩展以提高算法的预测准确率。因此参数 m 会最终成为算法性能的瓶颈。因此本文针对参数 m 进行了算法优化，使得时间复杂度降低为 $\mathcal{O}(m)$。优化过程如下。首先定义 $S_{j_{norm}}$ 为 S_j 归一化后的结果。

$$S_{j_{norm}} = \frac{S_j}{\|S_j\|} \tag{3.13}$$

因此结合公式 (3.11) 可以得到，

$$\begin{aligned} similarity &= \frac{V}{\|V\|} \cdot \frac{1}{m} \sum_{j=1}^{m} S_{inorm} \\ &= V_{norm} \cdot S_{pre} \end{aligned} \tag{3.14}$$

其中 V_{norm} 和 S_{pre} 定义如下:

$$\begin{aligned} V_{norm} &= \frac{V}{\|V\|} \\ S_{pre} &= \frac{1}{m} \sum_{j=1}^{m} S_{jnorm} \end{aligned} \tag{3.15}$$

可以发现 S_{pre} 和 V_{norn} 是相互独立的。基于此，可以对样本集进行预处理得到 S_{pre}，然后计算根据公式3.14计算平均余弦相似度。其中 S_{pre} 只需要计算一次便可以重复使用。与模型在上线后的识别消耗的总时间相比，S_{pre} 的计算开销可以忽略不计。因此平均余弦相似的时间复杂度降为 $\mathcal{O}(n)$。

3.4　实验仿真与分析

3.4.1　数据集

本文用于测试评估设备指纹算法性能的数据来自一个开源的 IoT 流量数据集。该数据集是研究人员监听了 27 种不同类型的 31 个物联网设备的启动阶段的网络流量得到的，其中有 4 种类型的设备有两个。这些设备包括智能照明电器、健康监测设备、家用电器和安全摄像头等。本文整理这些设备的相关描述见表3.5。研究人员在网关处通过 tcpdump 软件进行流量监听，然后通过 MAC 地址进行过滤后以 pcap 文件格式存储。每一个设备都被重复试验了 20 次，因此总共包含了 620 个 pcap 文件。

3.4.2　数值分析

为体现 TSMC-SVM 对 OvA-SVM 的改进，本文首先挑选了编号为 1-10 的 10 类设备进行试验，该设备集的特点是包含了一些来自同一设备产商的相似型号的产品，例如 TP-LinkPlugHS100 和 TP-LinkPlugHS110，EdimaxPlug1101W 和 EdimaxPlug2101W。实验结果解释了 TSMC-SVM 提高了对于上述相似型号设备的识别准确率。然后 TSMC-SVM 会与其它多分类算法进行试验比较，例如随机森林（Random Forest，RF），逻辑回归（Logistic Regression，LR）和 K-邻近（K nearest neighbor，KNN）等。同时实验最后对 TSMC-SVM 的可扩展性进行测试，即算法的识别准确率随着设备类型增加的变化情况。

实验首先需要确定的是设备启动阶段的时间窗口大小，窗口的大小以设备启动后的网络报文个数来度量。确定时间窗口的目的是为了节省存储空间以及优化算法。根据3.2.2节的介绍，一段流量经过特征提取后最终得到一个 $24+7n$ 维的特征向量。在实验分析中发现，窗口大小 n 与平均识别准确率并不是呈线性关系的。如图3.4是 OvA-SVM

表 3.5: 27 种设备类型介绍

编号	设备类型	类型描述
1	Aria	Fitbit Aria WiFi-enabled scale
2	HomeMaticPlug	Homematic pluggable switch HMIP-PS
3	Withings	Withings Wireless Scale WS-30
4	MAXGateway	MAX! Cube LAN Gateway for MAX! Home automation sensors
5	HueBridge	Philips Hue Bridge model 3241312018
6	HueSwitch	Philips Hue Light Switch PTM 215Z
7	TP-LinkPlugHS110	TP-Link WiFi Smart plug HS110
8	TP-LinkPlugHS100	TP-Link WiFi Smart plug HS100
9	EdimaxPlug1101W	Edimax SP-1101W Smart Plug Switch
10	EdimaxPlug2101W	Edimax SP-2101W Smart Plug Switch
11	EdnetGateway	Ednet.living Starter kit power Gateway
12	EdnetCam	Ednet Wireless indoor IP camera Cube
13	EdimaxCam	Edimax IC-3115W Smart HD WiFi Network Camera
14	Lightify	Osram Lightify Gateway
15	WeMoInsightSwitch	WeMo Insight Switch model F7C029de
16	WeMoLink	WeMo Link Lighting Bridge model F7C031vf
17	WeMoSwitch	WeMo Switch model F7C027de
18	D-LinkHomeHub	D-Link Connected Home Hub DCH-G020
19	D-LinkDoorSensor	D-Link Door & Window sensor
20	D-LinkDayCam	D-Link WiFi Day Camera DCS-930L
21	D-LinkCam	D-Link HD IP Camera DCH-935L
22	D-LinkSwitch	D-Link Smart plug DSP-W215
23	D-LinkWaterSensor	D-Link Water sensor DCH-S160
24	D-LinkSiren	D-Link Siren DCH-S220
25	D-LinkSensor	D-Link WiFi Motion sensor DCH-S150
26	SmarterCoffee	Smarter SmarterCoffee coffee machine SMC10-EU
27	iKettle2	Smarter iKettle 2.0 water kettle SMK20-EU

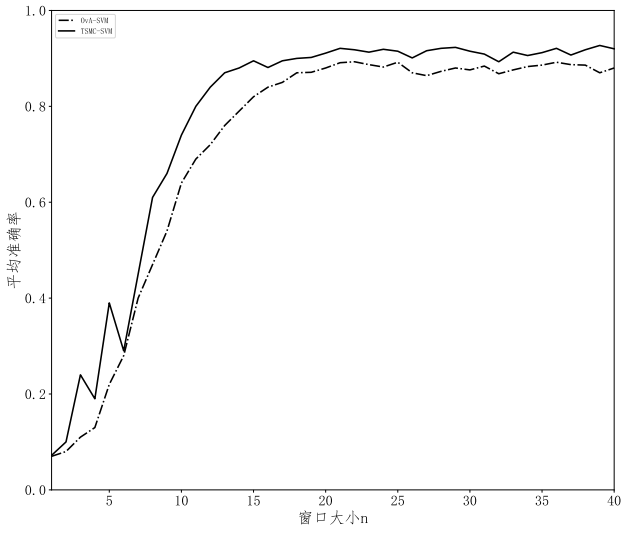


图 3.4: 启动窗口与平均识别准确率关系图

和 TSMC-SVM 算法的识别准确率和窗口大小 n 之间的关系图。对于 OvA-SVM 算法，当 n 小于 15 时，窗口大小和平均识别准确率近似呈线性关系。但是 n 当大于 15 之后，平

均识别准确率的增长开始变慢，并在 n 大约等于 20 时开始处于稳定状态。TSMC-SVM 也有相似的趋势，不同的是 TSMC-SVM 的是平均识别准确率稳定值比 OvA-SVM 的大。因此，过多的报文不但无法提高算法的识别性能，而且只会占用更多的存储空间和计算资源。基于此，本文后续实验的 n 值固定为 23，即设备启动后的 23 个报文被用于特征提取。因此特征向量的维度为 175。此外，从图中还可以发现，TSMC-SVM 算法经历了三个阶段：不稳定期，快速增长期和稳定其。当 n 小于 7 时，平均识别准确率呈现剧烈的抖动，本文称该阶段为不稳定期。随后，平均识别准确率快速增长，直到 n 值接近 20 时进入了平稳状态。TSMC-SVM 平均识别准确率的整体趋势是依赖于 OvA-SVM 的，并且其始终高于 OvA-SVM。这主要是由于 TSMC-SVM 模型的结构导致的，即 TSMC-SVM 是对 OvA-SVM 的分类重叠结果进行进一步的识别。

图3.5所示是当设备启动窗口大小 n 值固定为 23 时，SVM 和 TSMC-SVM 算法对每一类设备的识别准确率。从图中可以发现，TSMC-SVM 对每一类设备的识别准确率相比 SVM 都有提高。这很容易理解，因为 TSMC-SVM 是对 SVM 的分类结果实施进一步的识别，因此识别准确率只会提高或者不变，不会下降。但同时也可以发现，对于前六种设备，TSMC-SVM 的识别准确率的提高相比于后四种设备来讲不是很显著。经过仔细分析发现，前六种设备的类型区别很明显，它们或是来自不同产商的设备，或是来自同一厂商的完全不同类型的设备。例如 D-linkCam，D-linkswitch 和 D-linkwatersensor 虽然是来自同一厂商的设备，但是它们分别是摄像头、交换机和水质检测设备，因此 OvA-SVM 已经能提供较好的识别效果。但是对于 TP-linkPlugHS100 和 TP-LinkPlugHs110 是来自 TP-Link 的相同类型不同型号的设备，它们表现出了相似的网络行为, 因此导致 SVM 产生分类重叠的问题。同样的情况也发生在了 EdimaxPlug1101W 和 EdimaxPlug2101W 之间。但是可以发现上述问题经过 TSMC-SVM 第二阶段的余弦相似度识别后得到了缓解，识别准确率得到了大幅度的提高。

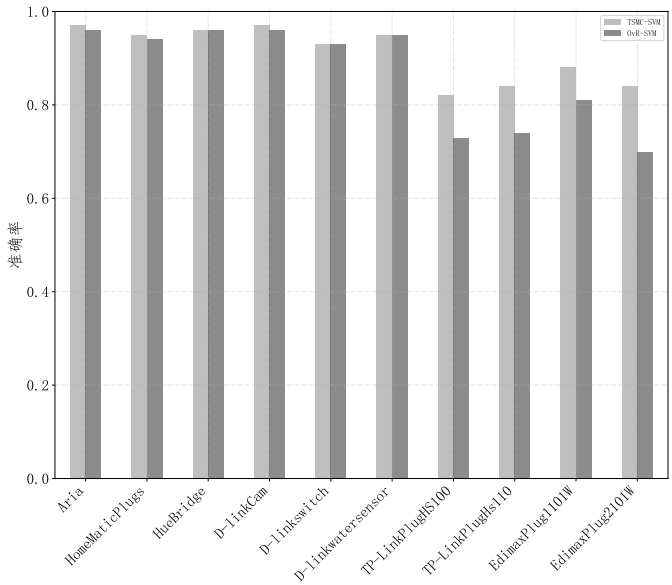


图 3.5: TSMC-SVM 和 SVM 的识别准确率对比图

表3.6 和表4.1 分别是对图3.5 在混淆矩阵层面的解释。它们描述了识别算法是如何错误地识别了设备类型。例如在 SVM 算法测试过程中，有 19% 的 TP-LinkPlugHS100 被错误识别成了 TP-LinkPlugHS110，同时有 21% 的 TP-LinkPlugHS110 被错误识别成了 TP-LinkPlugHS100。同样，在 EdimaxPlug1101W 和 EdimaxPlug2101W 之间，有 17% 的 EdimaxPlug1101W 被错误识别成了 EdimaxPlug2101W，同时有 26% 的 EdimaxPlug2101W 被错误识别成了 EdimaxPlug1101W。有而在 TSMC-SVM 算法测试过程中，上述的误识别率降低到了 12% 和 11%，10% 和 13%。

表 3.6: OvA-SVM 识别算法的混淆矩阵

真实值/预测值	1	2	3	4	5	6	7	8	9	10
1	96%	1%	1%	0%	0%	2%	0%	0%	0%	0%
2	1%	94%	4%	0%	1%	0%	0%	0%	0%	0%
3	0%	1%	96%	1%	2%	0%	0%	0%	0%	0%
4	1%	0%	1%	96%	0%	0%	0%	0%	2%	0%
5	0%	0%	2%	0%	93%	2%	0%	3%	0%	0%
6	0%	2%	0%	0%	0%	95%	2%	1%	0%	0%
7	0%	0%	3%	0%	0%	3%	73%	19%	0%	2%
8	0%	1%	0%	0%	0%	4%	21%	74%	0%	0%
9	0%	1%	0%	0%	0%	0%	0%	1%	81%	17%
10	0%	0%	1%	0%	0%	2%	1%	0%	26%	70%

表 3.7: TSMC-SVM 识别算法的混淆矩阵

真实值/预测值	1	2	3	4	5	6	7	8	9	10
1	96%	1%	1%	0%	0%	2%	0%	0%	0%	0%
2	1%	95%	3%	1%	0%	0%	0%	0%	0%	0%
3	0%	1%	96%	1%	2%	0%	0%	0%	0%	0%
4	0%	0%	1%	97%	0%	1%	0%	0%	1%	0%
5	0%	0%	2%	0%	93%	2%	0%	3%	0%	0%
6	0%	2%	0%	0%	0%	95%	2%	1%	0%	0%
7	0%	0%	2%	0%	0%	3%	82%	12%	0%	1%
8	0%	1%	0%	0%	0%	4%	11%	84%	0%	0%
9	0%	1%	0%	0%	0%	0%	0%	1%	88%	10%
10	0%	0%	1%	0%	0%	1%	1%	0%	13%	84%

表3.8 是对 OvA-SVM 和 TSMC-SVM 的相关统计，包括平均识别准确率，各个分类情况统计值。其中在 OvA-SVM 识别过程中，有 91.3% 的测试用例是可分类的（包括分类正确的和分类错误的）；而 1.6% 的测试用例不可分，系统将它们直接作为异常类型处理；5.1% 的测试用例发生了分类重叠现象。同时经过 TSMC-SVM 的余弦相似度识别后，原分类重叠的测试用例都转变成了可识别。但是通过对比平均识别准确率可以发现上述的转化过程中只有 3.6% 对准确率产生了贡献，而其它的 2.5% 转化成了分类错误的。

上述实验的窗口大小（n）和设备集大小（m）都是固定的，分别是 23 个报文和 10 类设备。本文接下来的实验将探索参数 n 和 m 对算法的平均识别准确率的影响。图3.6 所示是 TSMC-SVM 与其它分类算法在不同的参数值下的平均识别准确率的变化曲线。

表 3.8: 统计信息

算法	平均识别准确率	分类情况	统计值
OvA-SVM	89.6%	可分类	93.3%
		不可分类	1.6%
		分类重叠	5.1%
TSMC-SVM	93.2%	可分类	98.4%
		不可分类	1.6%
		分类重叠	0%

用于对比的算法包括 TSMC-SVM，Random Forest（RF），Logistics Regression（LR），K-Nearest Neighbor（KNN），Bayesian 和 Adaboost。本文首先将上述算法都训练成二分类器，然后经过 OvA 模型设计成多分类器。每一张图都表示在固定参数 n 时，各个算法的平均识别准确率随着参数 m 变化的曲线。从图中可以发现所有算法的平均识别准确率都随着设备类型的增加而下降，但 TSMC-SVM 始终位于其它算法之上。

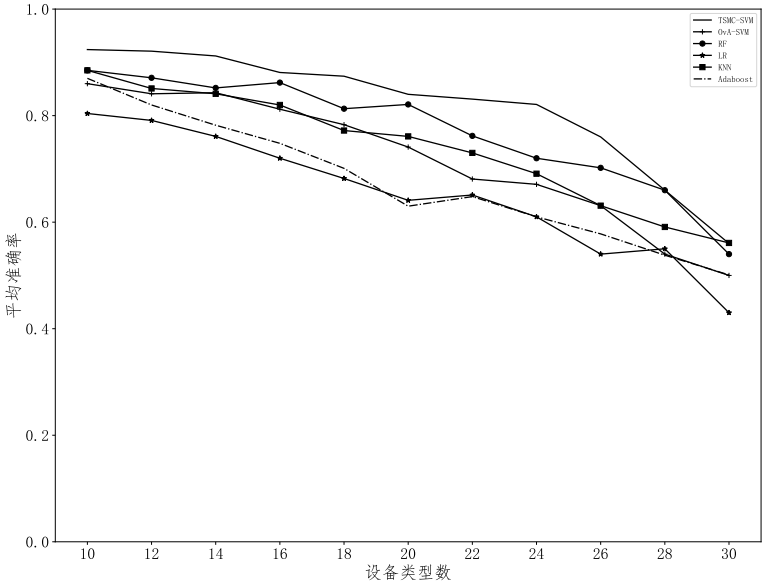


图 3.6: 增加设备类型对识别准确率的影响

3.5 本章小结

本章针对现有的设备指纹识别算法无法有效识别同一生产商的相似型号的设备类型，提出了 TSMC-SVM 设备指纹识别算法。该算法通过将修正余弦相似度概念引入 OvA-SVM 算法，提高了对上述设备类型的识别准确率。此外基于对设备网络流量特征提取的时间窗口选择问题的探讨，将该窗口固定在设备启动阶段。实验结果表明，当窗口大小为 23 个报文时，设备指纹的平均识别准确率就达到了 93.2%。最后实验对比了 TSMC-SVM 与其它指纹识别算法，其识别准确率明显优于其它算法。

第四章　基于设备行为可信度的访问控制模型

4.1　引言

访问控制的目的是实现对可信任用户的资源访问权限的分配。由于 IoT 设备的物理和计算资源匮乏，无法实施传统的基于强加密协议和复杂模型的认证机制，因而本文第三章提出了基于网络流量特征的设备指纹识别技术。该技术为实现基于 IoT 设备的访问控制系统提供了新的思路和技术支持。由于指纹提供的是设备类型信息，因此在基于角色的访问控制（RBAC）模型中，系统将根据指纹以设备类型为粒度分配角色，进而实现对设备个体的权限分配。但是，基于设备类型的粗粒度角色分配策略无法实现个性化的权限管理，即对根据设备个体的上下文分配特定的权限。此外，RBAC 模型固有的静态权限指派缺陷，即一旦完成权限与角色的指派后，该指派关系将不可变直到管理员进行重新指派。这导致了模型无法灵活适应设备的行为变化。因此，本文提出了一种基于行为可信度的授权机制（Bebavior Trust-Based Access Control，BTBAC）。该机制采用基于角色和可信度的访问控制模型，通过评估设备行为可信度和权限可信度阈值来动态调整授权状态。设备行为可信度是通过度量设备历史行为和当前行为的偏离程度得到的。而权限可信度阈值是根据资源的当前上下文动态设定。

一般来讲，对事物的可信度评价具有主观性和模糊性。为将主观因素客观化，以及量化模糊因素，本文提出了一种基于模糊综合评价的设备行为可信度生成方法。通过从多个维度度量设备行为的偏离值，以此来量化设备行为的异常程度。然后基于模糊集合理论对设备行为在各个维度的偏离程度进行综合评价，生成最终的可信度。

4.2　基于行为可信度的授权机制

在 RBAC 模型中，将一组权限与角色关联，用户通过成为适当角色的成员而得到这些角色的权限，简化了权限的管理。而针对本文以指纹作为设备身份认证的访问控制过程中，由于指纹只能识别设备的类型，因此角色的指派单位将不再是用户个体，而是同类型设备群体。这就造成了一个潜在的管理问题。如果由于某类型设备集合中的一台设备被恶意控制或者系统紊乱而发生异常操作资源的行为，系统管理将取消角色对该设备类型的指派以降低安全风险。那么这将导致该类型的所有设备失去被取消指派角色的权限，形成服务的大面积瘫痪。上述问题的根本原因是角色的粗粒度指派造成的，即角色被指派的单位是同类型设备群体，而不是设备个体。其次 RBAC 的权限指派过程是静态的。当角色被指派给用户时，权限与用户立即形成了绑定，直到该指派被取消为止。在绑定期间，用户的任何的正常或异常行为都不会影响该绑定关系。这将导致系统无法阻止合法设备在获取权限之后被恶意控制而造成的异常操作资源的行为。因此本文提出了

BTBAC 模型。该机制采用基于角色和可信度的访问控制模型，通过评估设备行为可信度和权限可信度阈值来动态调整授权状态。在 BTBAC 模型中，上述角色粗粒度指派问题和权限静态指派问题将被得到解决。

4.2.1　模型定义

图 4.1 描述了 BTBAC 模型的基本结构。和 RBAC 模型相比，该模型在角色与权限的指派过程中增加了信任授权节点。模型依据设备行为可信度和权限可信度阈值动态授予设备相应的权限。其中设备行为可信度是基于度量设备历史行为和当前行为的偏离程度得到的。而权限可信度阈值是根据资源当前所处的环境上下文设定的。下面是对 BTBAC 模型的元素定义和指派关系描述。BTBAC 模型的元素定义：

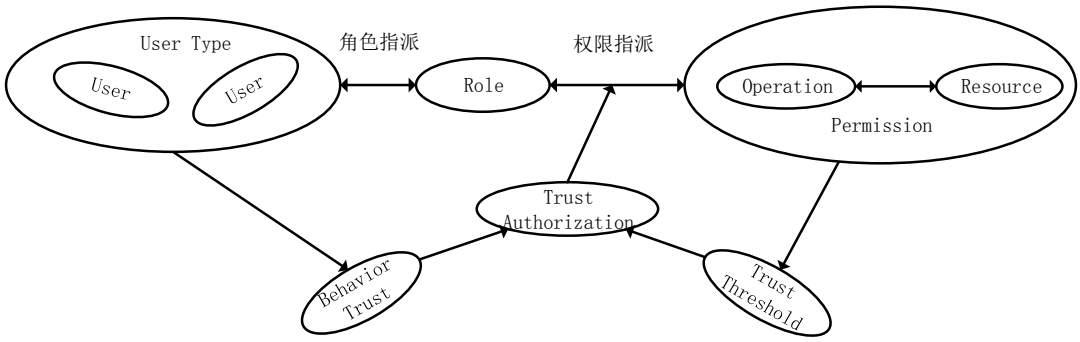


图 4.1: BTBAC 模型的基本结构

(1) 用户类型（*User Type*，*UT*）：用户类型代表同类型设备集合，是角色指派的最小单位。用 *UTs* 表示全体类型集合；

(2) 用户（*User*）：用户是访问资源的主体，即物联网设备个体。用 *Users* 表示全体用户集合，即全体物联网设备集合；

(3) 角色（*Role*）：角色是一组权限的集合，是连接用户和权限的桥梁，表示了对某一指定任务的能力。用 *Roles* 表示角色集合；

(4) 资源（*Resource, Res*）：模型中被主体访问的客体，主要是受保护的各种信息资源。用 *Ress* 表示资源集合；

(5) 操作（*Operation, Oper*）：使用资源的行为；例如在物联网设备访问网络的过程中，网络是资源，而访问代表一种操作；

(6) 权限（*Permission, Per*）：权限是由资源和操作构成的二元组 $Per =< Oper, Res >$，它表示将操作作用在资源上的资格；

(7) 行为可信度（*Behavior Trust*）：行为可信度是系统综合设备的历史行为和当前行为而做出可信度评估。它描述了设备当前行为与历史行为的偏离程度，用 *BT* 表示；

(8) 可信度阈值（*Trust Threshold*）：用户将操作作用于资源的最低行为可信度值，用 *TT* 表示。该阈值应该根据资源所在环境上下文的变化进行动态调整；例如随着时间变迁，资源的重要性和安全等级降低。此时可以降低资源的可信度阈值，使得用户可以凭借一个较低的行为可信度就可以操作该资源；

(9) 信任授权（*Trust Authorization*）：信任授权是由行为可信度和可信度阈值构成的二元组 $TA =< BT, TT >$，代表一个动态授权过程。该过程定义如下：

$$TA = \begin{cases} 0, & BT < TT \\ 1, & BT \geq TT \end{cases} \tag{4.1}$$

只有当 *BT* 大于等于 *TT* 时，用户才能行使对资源的操作资格。

BTBAC 模型的指派关系描述：

(1) 角色指派（*Role Assignment, RA*）：$RA \subset UTs \times Roles$。角色指派是指角色到用户类型的多对多映射关系。一个角色可以同时分别指派给不同的用户类型，同样一个用户类型可以同时接受多个不同角色的指派；

(2) 权限指派（*Permission Assignment, PA*）：$PA \subset Pers \times Roles$。权限指派是指权限到角色的多对多映射关系。一个权限可以同时分别指派给不同的角色，同样一个角色可以同时接受多个不同权限的指派；

(3) 可信度指派（*Trust Assignment, TA*）：可信度指派是指权限和用户的多对多动态映射关系。其初始指派关系由角色指派和权限指派确定。而在设备运行过程中，系统根据设备行为可信度和权限可信度阈值动态调整指派关系。可信度指派是 BTBAC 模型解决角色粗粒度指派和权限静态指派问题的关键。

从上述描述可以发现，BTBAC 模型中对用户的描述是两级的，分别是用户类型和用户个体。其中用户类型是角色指派的最小单位，而用户个体依然是操作资源的主体。产生这样的划分是设备指纹只能在类型层识别和访问控制模型需要更精细化管理的矛盾造成的。访问控制模型希望得到设备个体信息，以实现对设备个体的权限分配，但是设备指纹却只能提供设备的类型信息。因此，在 BTBAC 模型中，当设备发起资源访问请求时，系统首先根据设备类型为设备指派角色，实现权限的初始化分配。然后依据设备行为可信度动态调整设备与权限的绑定关系，实现动态的信任授权。

BTBAC 模型为权限设置了可信度阈值属性，设置该属性有两个目的：第一个目的是配合设备行为可信度以实现对用户个体级别的权限管理。当设备的行为可信度小于权限的可信度阈值时，系统将解除设备与对应权限的绑定。而当设备的行为行为可信度阈

值大于或者等于，系统将维持或者重新建立设备与对应权限的绑定关系。第二个目的是根据资源环境上下文动态设置权限可信度阈值。由于时间的变迁以及环境改变等因素，资源本身的重要性和安全等级也会随着变化。系统和管理人员应该根据这些上下文的变化来动态调整权限的可信度阈值，使得用户能够凭借一个较低的行为可信度访问该资源。

从上述描述中可以发现，当设备的行为可信度降低而失去了对资源的操作权限，同类型其他设备对该资源的正常操作不会受到影响。这样就解决了角色粗粒度指派的问题，实现了在用户个体级别实现对权限的管控。而且系统和管理人员可以根据资源所属的环境上下文动态调整权限的可信度阈值，克服 RBAC 模型的权限静态指派的问题。

4.2.2　授权流程

BTBAC 模型是以设备类型指纹和设备行为可信度为授权依据的访问控制模型。同 RBAC 模型一样，BTBAC 模型的角色指派单位也是同类型设备群体。但不同的是，在设备与权限绑定期间，系统通过设备行为可信度来动态调整该绑定关系。其中，行为可信度是依据设备历史行为和当前行为的偏离程度得到的。接下来，我们将详细介绍 BTBAC 模型的授权过程，图 4.2 是授权过程流程图。

(1) 初始化：初始化工作包括确定角色指派关系，权限指派关系和设置初始权限可信度阈值。其中角色指派和权限指派共同完成了权限到设备类型的预分配。只有当系统将角色正真分配给设备时，设备才能获得这些预分配的权限，并且获取的权限会随着设备的行为变化而动态调整。而权限可信度阈值应该根据资源当前环境上下文进行设置，因为它会影响到设备是否能够获得预分配的权限；

(2) 设备启动：设备指纹识别发生在设备的启动阶段。后续系统会根据指纹类型为设备分配相应的角色；

(3) 设备指纹识别：利用设备启动阶段网络流量的时域和频域特征，实现对设备类型的识别，并进行合法性判断。合法性判断的依据是 TSMC-SVM 分类器的输出结果是否为已知设备。如果是合法设备，将在步骤 (3) 中进行相应的角色分配，反之，拒绝授权；

(4) 角色分配：根据步骤 (2) 中识别出的设备类型分配相应的角色。该分配规则是提前拟定的，即初始化阶段的预分配过程。其依据是在设备获取角色的权限后能够提供完整的服务。至此，设备启动阶段的授权过程已经完成。由于系统对于新启动设备的默认态度是可信的，即针对当前时刻，步骤 (5) 的结果总是“是”，因此此时设备可以享有指派角色的所有权限；

(5) 行为可信度计算：行为可信度计算过程具有周期性和持续性。在物联网设备运行阶段，系统会周期性计算设备当前行为和历史行为的偏离程度，以此度量设备行为的

异常程度。一旦设备行为可信度小于权限的可信度阈值时，系统将终止设备和该权限的授权关系。

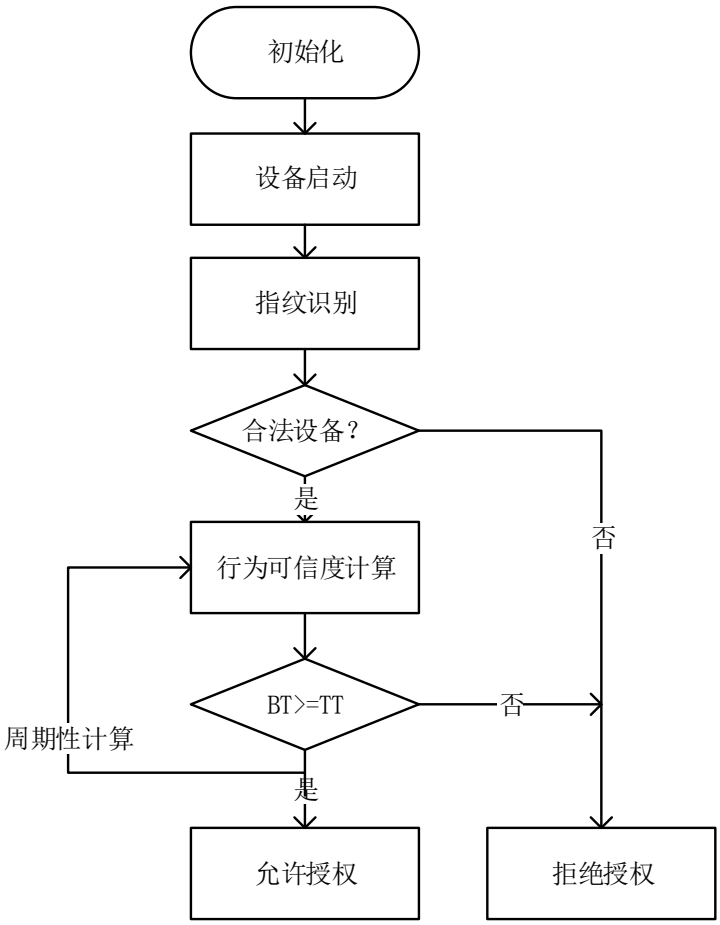


图 4.2: BTBAC 模型授权流程

4.2.3 权限可信度阈值

在 BTBAC 模型中，权限可信度阈值是权限的一个属性。其含义是设备获得权限所需要的最小行为可信度值。引入权限可信度阈值有两个目的：第一个目的是配合设备行为可信度以实现对用户个体级别的权限管理；第二个目的是根据资源环境上下文动态设置权限可信度阈值。

对于目的一，在 BTBAC 模型中，对用户的定义是分级的，即用户类型和用户个体。其中用户类型是角色指派的最小单位，而用户个体是资源操作的主体。这样设计的原因设备指纹识别技术无法提供设备个体粒度的信息，进而无法将角色直接指派给设备个体。当引入设备行为可信度和权限可信度阈值之后，系统会根据它们的值动态调整设备和权限的绑定关系。当设备的行为可信度小于权限的可信度阈值时，系统将解除设备与

对应权限的绑定。而当设备的行为行为可信度阈值大于或者等于，系统将维持或者重新建立设备与对应权限的绑定关系。这样，设备的行为可信度下降并不会影响同类型其它设备对权限的享有。

对于目的二，在传统的 RBAC 模型中，权限指派过程一旦完成，权限与角色所指派的用户之间将形成固定的绑定关系，直到该指派关系被解除为止。在绑定期间，资源本身的重要性变化无法在这种静态的指派关系中体现出来。引入权限可信度阈值之后，系统或者管理人员应该根据资源当前的环境上下文设置该阈值，以反映资源的重要性和安全等级的变化。

4.2.4　行为可信度

行为可信度描述了系统对设备的信任程度。行为可信度值越大，系统对设备越信任，并对设备开放的权限越多。因此，对于 RTABC 模型，行为可信度是使其有别于传统的 RBAC 模型的关键因素。接下来对行为可信度的属性进行介绍：

(1) 主观性：在物联网环境中，访问控制系统对设备的信任是具有主观性的。这些主观因素是由网络管理员引入的。在网络管理中，管理员会根据自身的经验、设备的历史行为以及任务需求等因素评估设备操作资源的合法性。并依据上述评估指标对设备分配相应的权限。因此，对于不同的管理员、不同的设备历史行为以及不同的任务需求，会产生不同的评判标准；

(2) 模糊性：访问控制系统对设备的行为可信度是具有模型性的，有很强的随机性和不确定性。该性质一方面是由主观性带来的；另一方面是由于设备的行为本身具有不确定性。在物联网环境中，设备的数量庞大，种类繁多以及相互之间的交互网络复杂，管理人员无法为每一个设备指定精确的行为规则；

(3) 动态性：行为可信度的动态性是由设备行为随着时间变迁和环境的变化造成的；物联网设备的物理部署环境相对开放，行为的动态变化可能是设备行为的周期性变化造成的，也有可能是被恶意毁坏或者控制而造成的异常行为；

(4) 可度量性：需要将主观的，模糊的信任进行量化，生成客观的、明确的数值。以该度量值来评估系统对设备的信任程度；量化过程应该从设备行为的多个维度进行，然后综合评估各个维度的量化值；

从上述属性可以发现，信任是一个客观的、模糊的概念，它受到多个不确定因素的影响。而在问控制系统中，系统需要将这些不确定的因素进行量化，以生成客观的、明确的可信度值。在 BTBAC 模型中，系统从不同维度提取设备的行为特征，并使用特定的算法计算行为可信度。接下来对行为可信度（BT）进行形式化定义：

$$BT = R \circ F * W \tag{4.2}$$

4.3 基于模糊理论的可信度生成方案

4.3.1 可信度评价因子

可信度评价因子是影响设备行为可信度评价的因素，在本文中指的是行为的不同维度。完备的可信度评价因子能够生成符合事实的评价结果，但是庞大的因子集会限制评价方案的可行性。在物联网时代，攻击者利用扫描攻击发现缺乏安全设计的设备。这些设备极易被控制被形成大规模的僵尸网络，并引发 DDos 攻击。因此 DDos 攻击是物联网时代的一个巨大安全难题。基于此，本文选择了一组可信度评价因子用于检测上述安全隐患，分别是上行流量端口熵（Downlink Traffic Port Entropy, DTPE），下行流量端口熵（Uplink Traffic Port Entropy，UTPE），上行流量 IP 熵（Uplink Traffic IP Entropy，UTIE），TCP 连接密度（Connection Density Trust，CDT）和历史信任（History Trust，HT）。

本文以设备视角定义流量的上行和下行，即源地址为设备 IP 的流量属于上行流量，而目的地址为设备 IP 的流量属于下行流量。对于评估因子 DTPE、UTPE 和 UTIE，它们检测的是在发生网络扫描时间时端口和 IP 的分布状态。本文引入熵的概念来表征该分布状态。熵用于表述一个系统的混乱程度，其最初源于热力学。后由 Shannon 将其引入信息论中，用于度量信息的不确定程度。信息熵的计算公式如下：

$$H = -\sum P(x) log P(x) \tag{4.3}$$

对于评估因子 DTPE，用于检测攻击者对该设备的端口扫描行为。物联网设备由于功能与服务相对单一，因此其暴露的端口趋于集中。但时当攻击者向其发起端口扫描时，会改变流量中端口的分布，使该分布变得分散，即熵增大。同理，评估因子 UTPE 和 UTIE 也是用于度量这种分布状态。但不同的是，它们描述的是不同是事件。对于 UTPE，其描述的是该设备向其他设备发起端口扫描的事件。因此它需要一个前提假设，即该设备已经被控制。而攻击者正在利用它向其他设备实施端口扫描。同样基于上述假设之下的还有 UTIE，只不过它描述的事件更符合网络发现。即设备以扫描的方式寻找周围其他活跃的设备，为后续端口扫描和实施攻击做准备。

而 CDT 度量的是 TCP 的连接密度，即在单位时间内的连接数量。当大规模被控设备实施 DDos 攻击时，每一台设备都会向目标地址发起大量无效的连接，耗尽目标主机的资源或者带宽。在这里，一个 Syn=1 的报文代表了一个链接。虽然这种指代是不精确的，但是从统计层面，已经能够反映出 TCP 连接密度的趋势了。HT 表示的是系统对设备行为的历史可信度。历史信任的定义是设备在权限请求的过程中，成功次数的比率。

4.3.2 模糊行为可信度

本文将行为可信度 BT 定义为设备当前行为和历史行为的偏离程度，其取值范围为 [0,1]。只有当 $BT \geq TT$ 时，设备才能获得对资源的操作权限。在模糊综合评价行为可

信度的过程中，首先需要建立评价因素集和评语集。本节根据 4.3.1 节定义的可信度评价因子建立行为可信度的因素集，即 U={DTPE, UTPE, UTIE, CDT, HT}。同时为信任因素建立评语集，V={不可信 (v_1)，一般可信 (v_2)，一般不可信 (v_3)，可信（v_4）}。然后基于隶属函数模糊化信任因素，并结合评语集构建可信度评价矩阵 R。因素权重向量 W 代表的是因素集中各个成员在对最终评价结果的重要性。本文采用普遍使用的层次分析法得到因素权重向量。最后通过模糊合成运算得到最终的评价结果。整个评价过程如图 4.3 所示。接下来介绍行为可信度评价的详细过程：

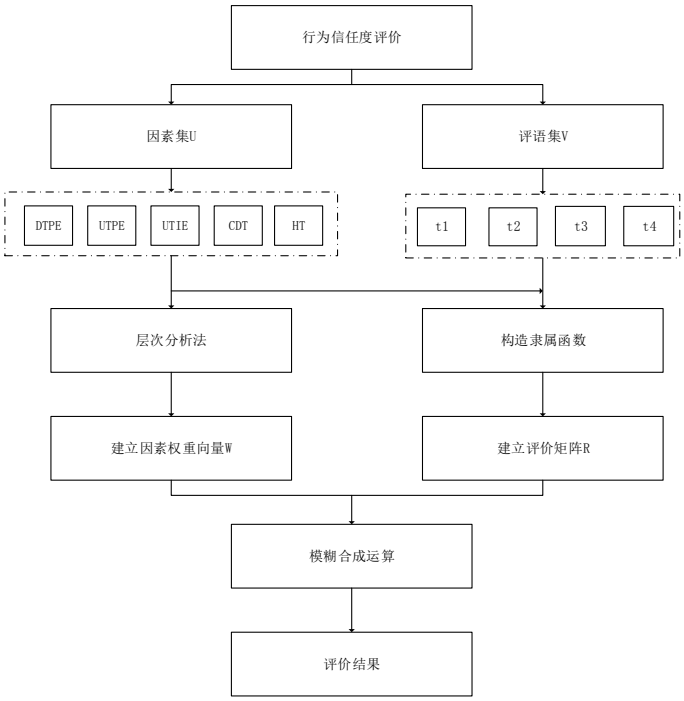


图 4.3: 模糊综合评价流程

建立因素集和评语集

根据 4.3.1 节定义的可信度评价因子建立因素集 U，同时为信任因素建立评语集 V。U 和 V 的定义如下：

$$\begin{aligned} U &= \{DTPE, UTPE, UTIE, CDT, HT\}, \\ V &= \{v_1, v_2, v_3, v_4\} \end{aligned} \tag{4.4}$$

建立评价矩阵

根据4.3.1节的定义，HT 反映的是设备的历史可信度，即设备过去权限请求成功的比率。HT 的定义如下：

$$TH = \begin{cases} \dfrac{N_t}{N_t + N_f}, & N_t + N_f \neq 0 \\ 1, & N_t + N_f = 0 \end{cases} \tag{4.5}$$

其中 N_t 表示历史权限请求成功的次数，N_f 表示历史权限请求失败的次数。若设备有较高历史可信度，那么设备当前行为的可信程度越高；反之，当前的可信程度越低。其次当设备还未产生历史行为时，即 $N_t + N_f = 0$，默认其历史行为是高可信的。为了描述 HT 的模糊性，需要使用隶属度函数将因素的量化值映射到评语集的 [0,1] 区间，用于表征该因素值隶属于某一评语的程度。HT 的隶属度函数定义如图4.4所示。它由梯形和三角形组成。

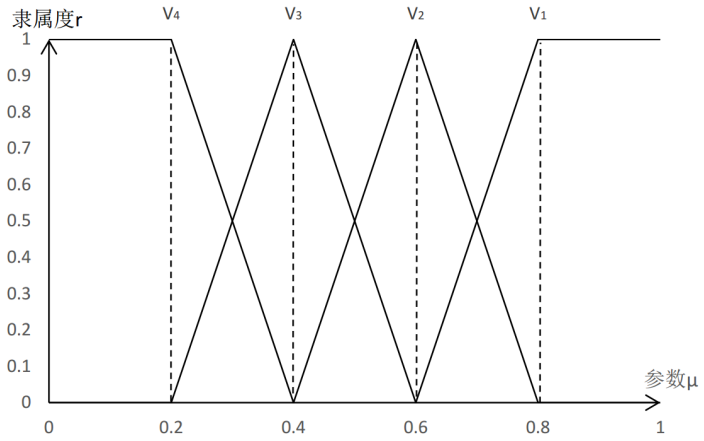


图 4.4: HT 的隶属度函数

根据图4.2描述，系统会周期性计算设备的行为可信度。对于可信度评价因子 DTPE，假设经过当前本轮的可信度计算中其值为 $dtpe_t$，那么其存在一个历史集合 $DTPE\_S$:

$$DTPE\_S = \{dtpe_1, dept_2, dept_3, ..., dept_{t-1}\} \tag{4.6}$$

为度量 $dtpe_t$ 与 $DTPE\_S$ 的偏离程度，对其进行标准化处理：

$$dtpe_t^* = \frac{dept_t - u}{\sigma} \tag{4.7}$$

其中 u 和 σ 分别是 $DEPT\_S$ 的均值和标准差。同理，对于 UTPE, UTIE 和 CDT 可以计算得到 $utpe_t^*$，$utie_t^*$ 和 cdt_t^*。为描述上述评价因子的模糊性，使用图5.1所示的隶属函数。该隶属度函数式通过正太分布图平移得到，其中的 u 和 σ 分别对评价因子历史集合的均值和均方差。

基于上述隶属度函数对可信度评价因子集 U 中的每一个元素在评语集 V 上的评价，可以得到评价矩阵 $R_{5\times4}$：

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{34} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{44} & r_{44} \\ r_{51} & r_{52} & r_{53} & r_{54} \end{bmatrix} \tag{4.8}$$

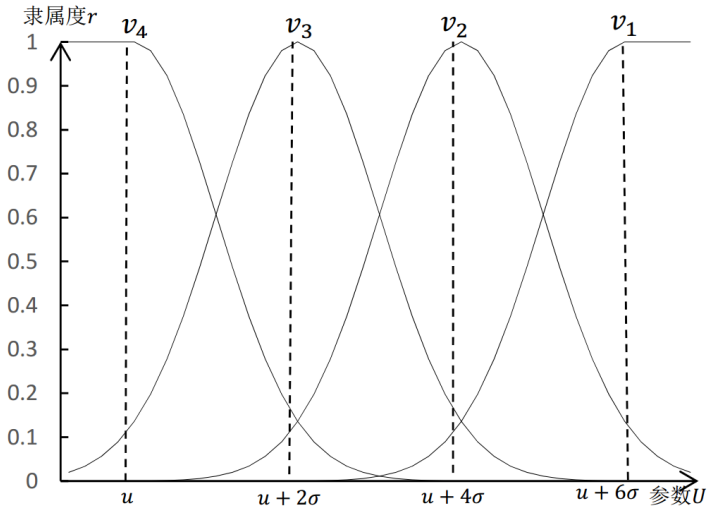


图 4.5: UTPE,UTIE 和 CDT 的隶属度函数

其中 r_{ij} 表示的是评价因子集 U 中第 i 个因子对评价集 V 中第 j 个评语的隶属度。

建立权重向量

在模糊综合评级过程中，各个评价因子对最终的评价结果的重要程度有所不同。为此，引入了权重的概念。评价因子集 U 对应的权重向量表示为 W。本文采用层次分析法建立权重向量，依据 T.L.Satty 的 1-9 标度法，生成成对比较矩阵 A:

$$A=\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{5} & 2 \\ 2 & 1 & 1 & \frac{1}{3} & 3 \\ 2 & 1 & 2 & \frac{1}{3} & 3 \\ 5 & 3 & 3 & 1 & 7 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{7} & 1 \end{bmatrix} \tag{4.9}$$

从理论上分析，成对比较矩阵 A 需要是一个完全一致性矩阵，应该满足：

$$a_{ij}a_{jk}=a_{ik}, 1\leq i,j,k\leq n \tag{4.10}$$

但是在实际情况下成对比较矩阵要完全满足上述等式是不可能的。例如对于矩阵 A，$a_{12}a_{23}\neq a_{13}$。因此需要降低对成对比较矩阵的一致性要求，即可以允许成对比较矩阵存在一定程度的不一致性。进过分析可知，完全一致的矩阵的绝对值对打的特征值等于该矩阵的维数。因此检验成对比较矩阵的一致性程度就可以转变成计算该矩阵的绝对值最大的特征值和矩阵维数的差异程度。该差异指标 CI 定义如下：

$$CI=\frac{\lambda_{max}(A)-n}{n-1} \tag{4.11}$$

对于成对比较矩阵 A，计算得到其绝对值最大的特征根为 $\lambda = 5.254$。将其带入公式4.11：

$$CI = \frac{\lambda_{max}(A) - n}{n-1} = \frac{5.254 - 5}{5-1} = 0.063 \tag{4.12}$$

通过查询平均随机一致性指标 RI 表，当 n=5 时，RI=1.12。计算 CR:

$$CR = \frac{CI}{RI} = \frac{0.063}{1.12} = 0.056 < 0.1 \tag{4.13}$$

因此，上述成对比较矩阵 A 满足一致性要求。其绝对值最大的特征根对的标准化特征向量将作为权重向量，就 $W = (0.093, 0.170, 0.210, 0.470, 0.057)$。

表 4.1: 平均随机一致性指标 RI

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.52	0.89	1.12	1.26	1.36	1.41	1.46	1.49	1.52	1.54	1.56	1.58	1.59

模糊合成运算

模糊合成运算数学定义如下：

$$RN = W \circ R \tag{4.14}$$

其中 W 是因素集的权重向量，R 是因素集的评价矩阵，运算符 $\circ$ 表示模糊合成运算规则。为演示模糊合成运算，假设 R 为：

$$A = \begin{bmatrix} 0.35 & 0.69 & 0.22 & 0.04 \\ 0.17 & 0.35 & 0.39 & 0.09 \\ 0 & 0.70 & 0.34 & 0.26 \\ 0.09 & 0.62 & 0.30 & 0.39 \\ 0.43 & 0.35 & 0.22 & 0 \end{bmatrix} \tag{4.15}$$

本文采用模糊运算规则是平均加权法，则：

$$\begin{aligned} RN &= WR \\ &= [0.093, 0.170, 0.210, 0.470, 0.057]\, A \\ &= [0.128, 0.582, , 0.333, 0.257] \end{aligned} \tag{4.16}$$

根据最大隶属原则，当前设备的行为可信度为一般可信。因此，在权限调整的过程中，设备只能获得信任度阈值为 t_1, t_2 的权限。

4.4 本章小结

本章提出了 BTBAC 模型。该模型采用基于角色和可信度实现权限访问控制，通过评估设备行为可信度和权限可信度阈值来动态调整授权状态。此外基于模糊综合评价方法提出了一种设备行为行人度评价方案。针对设备行为的不同维度建立评价因子集。每一个评价因子通过度量设备历史行为和当前行为的偏离程度得到的。然后结合权限可信度阈值实现对设备权限的动态调整。其中权限可信度阈值是根据资源的当前上下文动态设定。该模型能够有效的缓解基于设备类型的粗粒度角色分配策略无法实现个性化权限管理的问题。

第五章　基于设备指纹的物联网访问控制系统设计与实现

5.1 引言

本章介绍基于设备指纹的物联网访问控制系统的设计与实现，该系统以设备指纹和行为可信度访问控制机制为核心实现对物联网设备的权限访问控制。设备指纹实现对设备类型的识别，使得系统能够完成设备类型层面的粗粒度角色分配。而行为可信度访问控制机制依据设备的网络行为和权限的上下文变化实现对设备个体和权限之间的动态绑定关系。本文将从逻辑视图、过程视图、实现视图、物理视图和场景视图介绍系统的设计与实现过程：

(1) 逻辑视图：逻辑视图主要支持功能性需求，即在为用户提供服务方面系统所应该提供的功能。对于有明显层次特性的系统，可以对其进行分层设计。同时在每一层进行模块化或者组件化分解。以上目的都是为了给用户一个清晰的系统功能视图，明晰系统层与层的界线，降低模块与模块的设计耦合程度。对于数据驱动程度高的应用程序，可以使用 E-R 图描述数据的定义；

(2) 过程视图：过程视图描述的是系统的运行流程，或者模块（组件）之间的通信过程。对于数据驱动程度高的应用程序，也可以通过数据流转图来描述数据在系统运行过程中的流转。对于过程视图，本章将通过系统组件之间的时序图和数据流转图来描述系统的运行流程。

(3) 实现视图：实现视图关注的是系统的实现方式。对于本文的物联网访问控制系统，其核心模块为设备指纹识别、设备行为可信度计算和策略实施。其中设备指纹识别和设备行为可信度计算已经在本文的第三章和第四章展开了详尽的介绍，因此本章在该视图中将详细介绍策略实施的实现；

(4) 物理视图：对于拥有多个物理处理单元的系统，物理视图描述了它们在物理分布和通信方式。对于本文的物联网访问控制系统，存在多个不同的网络节点。例如：物联网设备、交换机、访问控制服务器等。因此在该视图中将给出系统的网络物理部署图；

(5) 场景视图：系统架构的描述可以围绕上述四个视图来组织。然后通过一些实际用例或场景来进一步进行说明，从而形成第五个视图——场景视图。对于场景视图，本章主要用于展示可视化的 web 配置中心。

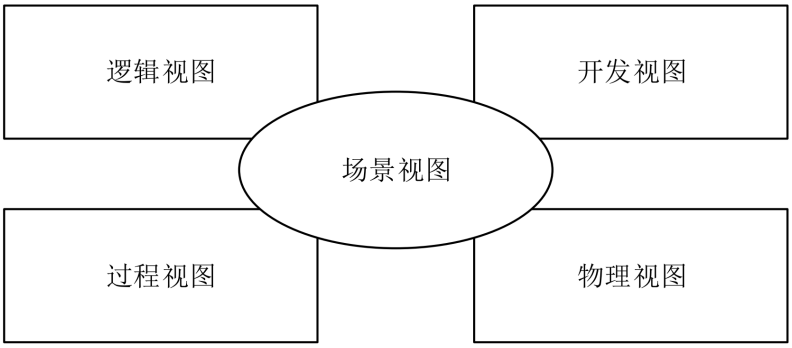


图 5.1: “4+1”视图模型

5.2 逻辑视图

5.2.1 系统整体架构

图5.2 所示是系统的整体架构图。该系统自底向上总共分为四层，分别是物理层、控制层、数据层和应用层。物理层主要包含的是物联网设备和交换机。其中，交换机是整个系统的关键节点，所有的物联网设备都需要通过交换机与其它网组的设备通信或访问资源。此外，交换机还是物理层和控制层的逻辑切面。控制层中的流量采集、设备发现和策略实施都需要通过交换机完成。流量采集模块用于提供设备指纹识别和设备行为可信度计算所需的数据，这些数据都是通过复制交换机的镜像流量得到的。而设备发现模块是为了跟踪设备的启动事件。为了避免通过轮询检查交换机端口是否有设备上线，系统采用了 SNMP TRAP 监听交换机端口的 UP 事件。当产生 UP 事件时，流量采集模块将过滤出事件相关设备的 MAC 地址的报文，未设备指纹识别模块提供数据准备。同时，SNMP TRAP 还能监听交换机端口的 DOWN 事件。该事件能够为系统提供设备下线的信息，以便于系统回收该设备所占有的资源。此外，权限控制的策略实施模块也需要借助交换机实现。由于本文的访问控制系统对终端设备具有无侵入性的特点，因此权限访问控制的策略实施将由交换机完成。权限控制的策略实施是通过调整设备所连接交换机端口隶属的 VLAN 集合实现的，该策略实施过程的细节将在5.4小节详细介绍。数据层主要是对控制层采集的流量进行处理，包括设备指纹识别（特征提取和指纹识别）和设备行为可信度计算（评价因子提取和模糊综合评价）。应用层主要的作用是接收数据层发现的知识，并依据配置中心的预设信息生成权限分配决策，然后下发给控制层的策略实施模块完成最终的权限控制。

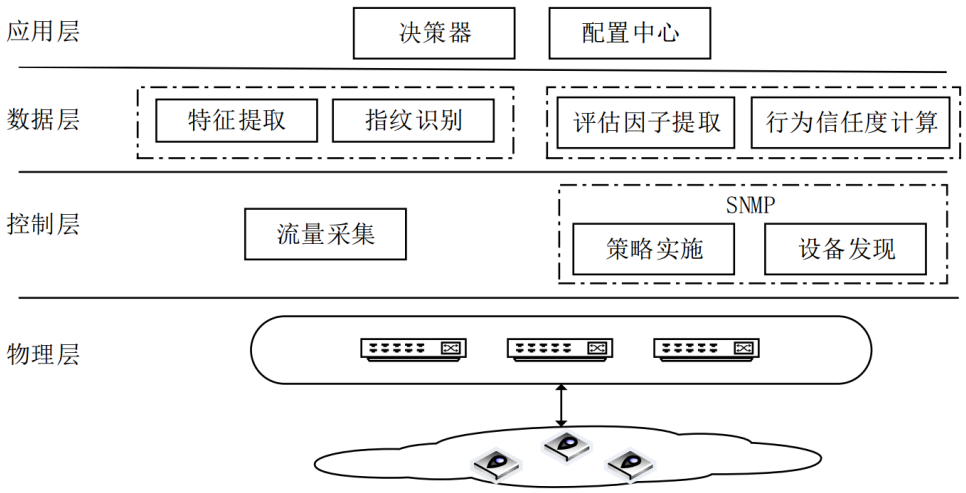


图 5.2: 系统逻辑架构

5.2.2 数据定义

表5.1 和表5.2 所示是本文的物联网访问控制系统的相关数据定义。其中表5.1 定义的是设备指纹识别相关的数据，包括样本特征向量，预处理因子和流量。样本特征向量提取自设备启动流量样本，用于 TSMC-SVM 算法的训练，以 csv 格式存储。预处理因子是对样本特征向量提前预处理得到的，其目的是用于降低 TSMC-SVM 算法在第二阶段的余弦相似度匹配式的算法时间复杂度。根据3.3节介绍，预处理因子将余弦相似度匹配的时间复杂度从 $O(nm)$ 降低到 $O(n)$。而流量可以分为两种类型，设备启动流量和实时流量。前者产生于设备启动阶段，用于设备指纹识别；而后者需要实时监听，并周期性提交并计算行为可信度，用于检测设备行为的变化。流量数据都是通过交换机的镜像流量复制得到，并以 pcap 格式存储。行为可信度定义的是模糊综合评价因子和行为可信度值，结合权限的可信度阈值能够实现设备与权限的动态绑定。表5.2定义的是 BTBAC 模型的相关数据。其中设备、设备类型、角色和权限定义了模型中的实体。其中设备是权限指派的最小单位，设备类型是角色指派的最下单位。设备类型-权限和角色-权限定义了设备类型与角色和角色与权限之间的指派关系，描述了权限到设备的分配过程。策略是完成权限控制的动作。在本文的系统中，权限控制的策略实施是通过调整设备所连接交换机端口隶属的 VLAN 集合实现的。

表 5.1: TSMC-SVM 算法数据实体定义

数据类型	数据描述
样本特征向量	TSMC-SVM 算法训练数据，以 csv 格式存储
预处理因子	余弦相似度匹配数据
设备启动流量	指纹识别所需数据，以 pcap 格式存储
实时流量	行为可信度计算所需数据，以 pcap 格式存储
行为可信度	结合权限的可信度阈值实现设备与权限的动态绑定

表 5.2: BTBAC 模型数据实体定义

数据实体	数据描述
设备	终端设备的描述、网络配置和状态相关信息集合
设备类型	同类型设备集合，是角色指派的最小单位
角色	完成特定任务的一组权限的集合
权限	操作特定资源的资格
设备类型-角色	设备类型和角色的指派关系
角色-权限	角色和权限的指派关系
策略	完成权限控制的动作

表5.1 和表5.2给出了本文系统中的关系型数据和非关系性数据的定义。其中关系型数据采用 MySQL 来存储, 如图5.3和5.4。图中描述了系统核心数据的 E-R 图。其中图5.3描述了设备、设备类型、角色和权限等实体数据，而图5.4描述了上述实体之间的关系。例如表 sys_role_device_type 描述的是角色和设备类型的指派关系；表 sys_role_acl 描述的是权限与角色的指派关系。

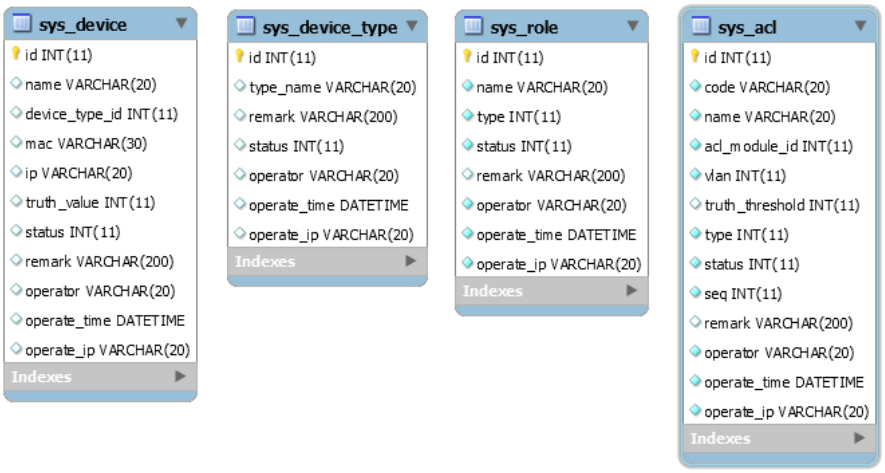


图 5.3: 系统核心数据 E-R 图

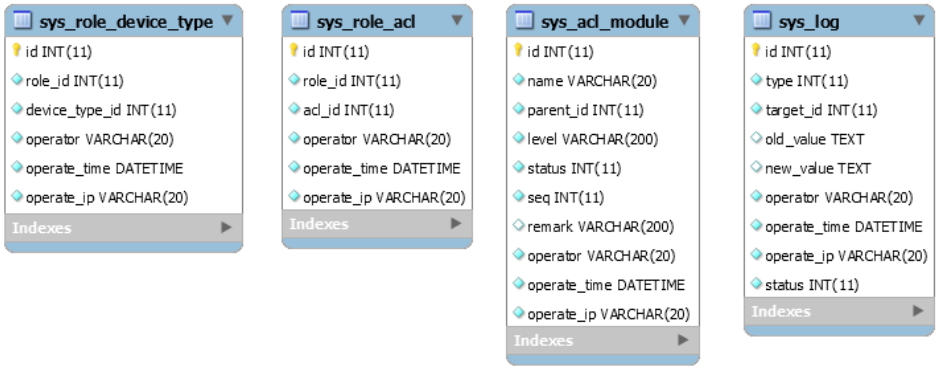


图 5.4: 系统核心数据 E-R 图

5.3 过程视图

5.3.1 数据流转视图

图5.5 所示是系统数据的流转示意图。其中终端设备的部分网络流量数据经过特征提取之后被输入设备指纹识别算法进行设备类型识别，同时另一部流量经过评价因子提取后被输入行为可信度计算算法检测设备行为与历史行为的偏离度。当设备类型信息和设备行为可信度一同流转经权限分配决策节点后输出决策结果。当然，该决策过程还需要权限规则数据参与。权限规则数据由管理人员通过 Web 管理平台进行配置，并在数据库中持久化。策略指令最后经交换机管理节点完成对交换机的管控。同时终端管理还兼顾了终端发现的职责，即利用 SNMP TRAP 跟踪换机端口的 UP/DOWN 状态，进而发现设备的上线和下线。

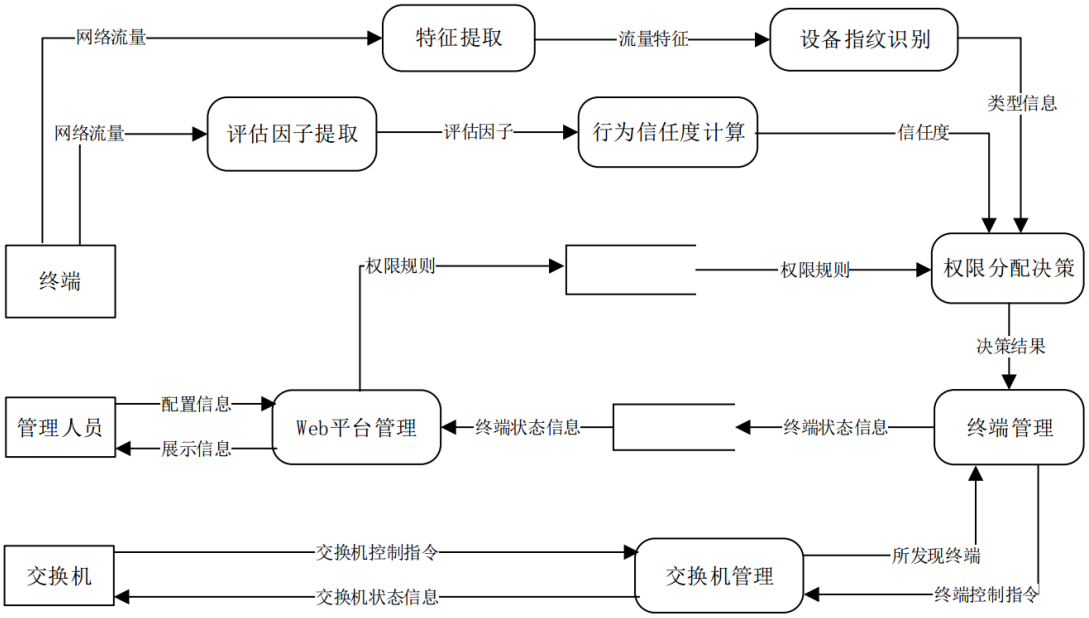


图 5.5: 数据流转图

5.3.2 系统组件时序视图

5.4 实现视图

5.4.1 基于 VLAN 的权限划分

VLAN 作为一种二层交换技术，能够实现对局域网在逻辑层面上的划分。如图5.6所示是基于 VLAN 的权限划分示意图。系统根据类型以及重要程度将不同的资源划分到不同的 VLAN 中。同时通过控制设备所连接的交换机端口加入或离开 VLAN 以实现对资源的访问控制管理。交换机端口默认只能隶属于一个 VLAN，这导致了设备在同一时刻只能获得单个 VLAN 中的资源权限。为克服该问题，需要将交换机端口设置为 TRUNK

模式。在该模式下，端口能够同时隶属于多个 VLAN。然后通过调整端口的 VLAN 隶属集合实现对资源权限的访问控制。图中设备所连接的交换机端口 EHT 在时刻 t1 时刻的 VLAN 隶属集合为 V={VLAN1,VLAN2,VLAN3,VLAN4}，相应的设备享有所有资源的权限。而在下一时刻 t2，该端口的 VLAN 隶属集合调整为了 V={VLAN2,VLAN3,VLAN4}，此时设备不再享有访问资源 A 的权限。

此外，还需要关注的是在设备启动时刻，所连接端口的状态。根据第四章的介绍，设备在通过指纹合法性校验后，其信任状态为完全可信。因此，此时的端口应该隶属于角色所拥有的 VLAN 集合。

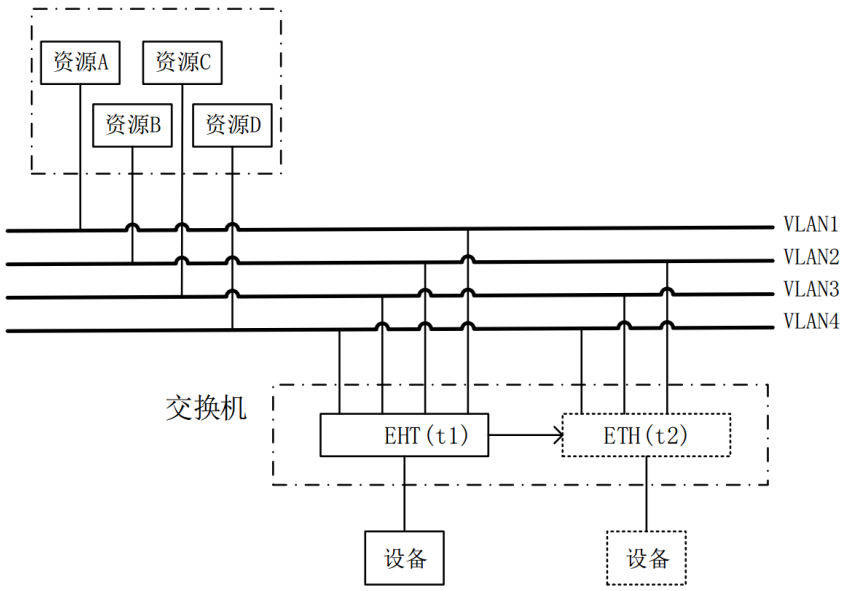


图 5.6: 基于 VLAN 的权限划分示意图

5.4.2　基于 VLAN 的策略实施过程

图5.7 所示是策略实施过程流程图。过程如下：

(1) 初始化：初始化工作包括确定角色指派关系，权限指派关系和设置初始权限可信度阈值。角色指派和权限指派共同完成了权限到设备类型的预分配。其中权限的粒度由 VLAN 的划分决定，而角色将拥有一个 VLAN 集合所对应的资源访问权限。

(2) 设备启动：设备指纹识别发生在设备的启动阶段。后续系统会根据指纹类型为设备分配相应的角色；

(3) 设备指纹识别：设备指纹识别的目的是为了识别设备的类型，进而分配预设的角色。

(4) 角色分配：根据步骤 (2) 中识别出的设备类型分配相应的角色。该分配规则是提前拟定的，即初始化阶段的预分配过程。至此，设备启动阶段的授权过程已经完成。由于系统对于新启动设备的默认态度是完全可信的。即此时设备将能够访问角色拥有的 VLAN 集合内的所有资源;

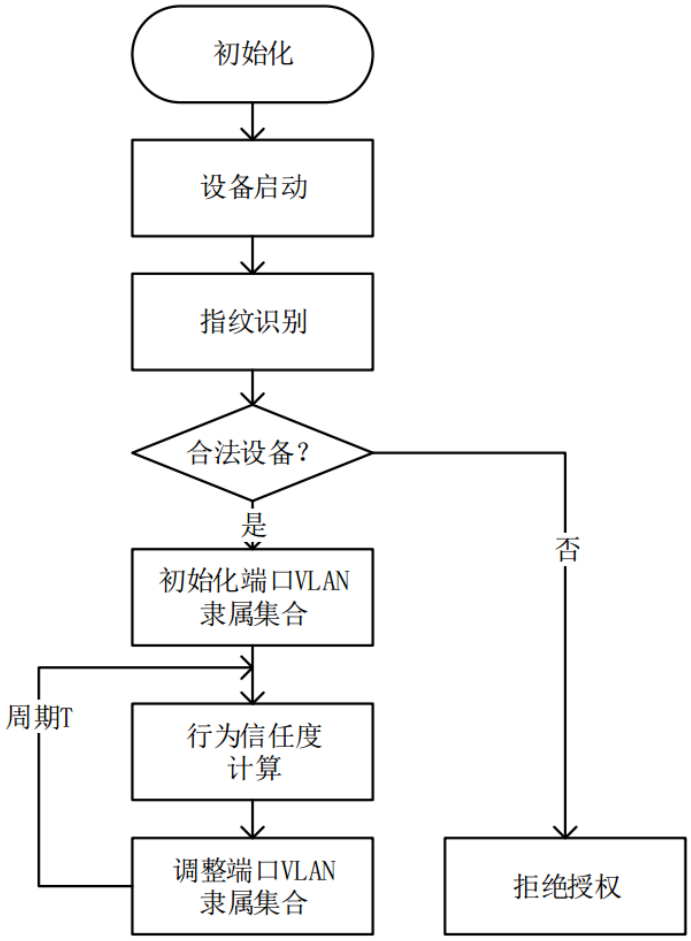


图 5.7: 基于 VLAN 的策略实施流程图

(5) 行为可信度计算：行为可信度计算过程具有周期性和持续性。在物联网设备运行阶段，系统会周期性计算设备当前行为和历史行为的偏离程度，以此度量设备行为的异常程度。一旦设备行为可信度小于权限的可信度阈值时，系统将终止设备和该权限的授权关系。

(6) 系统将根据步骤 5 的行为可信度和预设的权限可信度阈值，动态调整设备所连接端口的 VLAN 隶属集合，进而实现对资源的访问控制管理。当行为可信度值小于权限的可信度阈值时，系统将该权限对应的 VLAN 从端口的 VLAN 隶属集合中移除；同理，当行为可信度值大于等于权限的可信度阈值时，系统将该权限对应的 VLAN 加入到端口的 VLAN 隶属集合中。

5.5 物理视图

图5.8是本文的物联网设备访问控制系统的物理部署示意图。为降低带宽，流量采集与数据预处理服务被部署在了同一台服务器。经预处理后提取的特征和行为可信度评价因子被提交给了决策服务器，进行最终决策。其中可视化的 web 配置中心也部署在了

决策服务器上。管理人员可以通过浏览器访问进行信息配置。

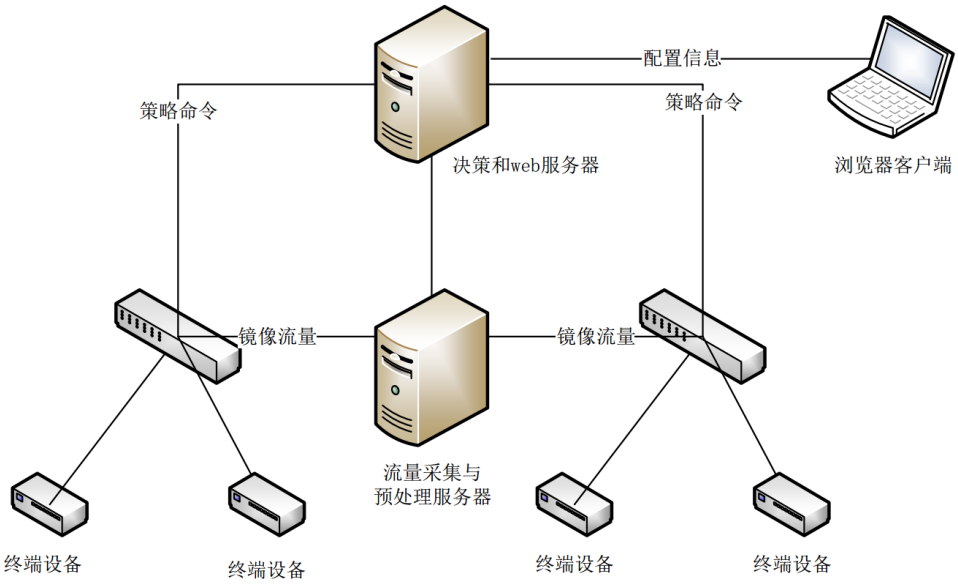


图 5.8: 系统物理部署视图

5.6　场景视图

在场景视图中本章主要介绍可视化的 web 配置中心。为便于网络管理员对设备的管理和系统维护，本文作者开发了一个可视化的 web 配置中心。该配置中心的核心模块是设备管理、角色管理和权限管理。其中设备管理包括对设备类型和设备个体的管理。管理员可以通过新增和编辑功能来添加或修改相关信息。角色管理模块维护的是角色信息、角色与权限的关系以及角色与设备类型的关系。同设备管理模块一下，管理员可以增加或者编辑相关信息。最后的权限管理模块维护的是权限的相关信息。权限定义主要根据资源所在 VLAN 进行划分。管理人员可以新增或者编辑权限相关信息。本节接下来将给出各个模块的管理面板。

5.6.1　设备管理

如图5.9是访问控制系统的权限管理页面，共分为两个面板。其中左边的是权限管理类目面板，该下拉菜单维护了四个类目：设备管理、角色管理、权限管理和权限更新记录。右边的面板是针对不同类目的信息维护。例如图5.9中显示的是设备管理类目的相关信息。设备管理分为设备类型列表和设备列表。系统支持对设备类型信息的添加、编辑和删除，对设备信息的编辑。如图5.10，图5.11和图5.12所示。

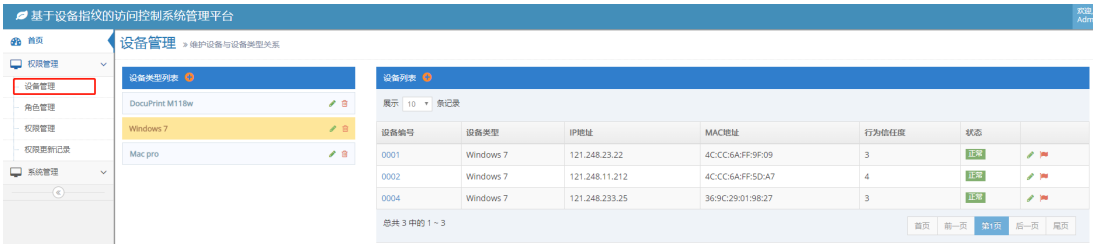


图 5.9: 权限管理页面

图 5.10: 增加设备类型信息

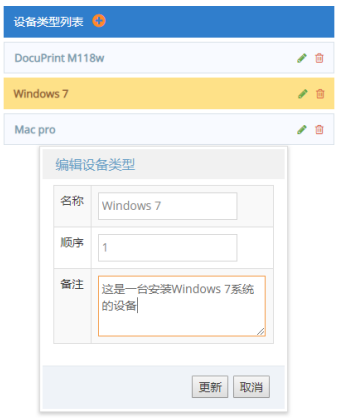


图 5.11: 编辑设备类型信息

图 5.12: 编辑设备信息

5.6.2 角色管理

如图5.13所示系统的角色管理页面。其中左边的面板维护的是设备角色列表、角色与权限的关系和角色与设备类型的关系。在角色列表中支持对角色的添加、编辑和删除。而角色与权限、角色与设备类型维护的是它们之间的指派关系，如图5.16和图5.17所示。

图 5.13: 角色管理页面

图 5.14: 增加角色

图 5.15: 编辑角色

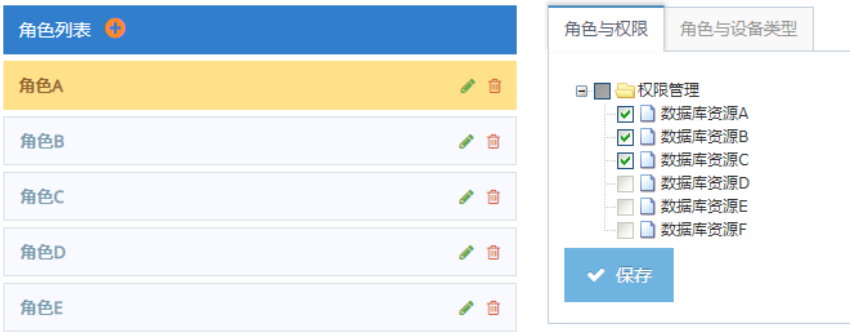


图 5.16: 角色与权限管理

图 5.17: 角色与设备类型管理

5.6.3　权限管理

如图5.18所示是系统权限管理页面。左边的面板中维护了权限模块列表信息和权限点列表信息。其中资源和 VLAN 组合是权限划分的依据。系统支持对权限点的增加、编辑和删除，如图5.19和图5.20所示。

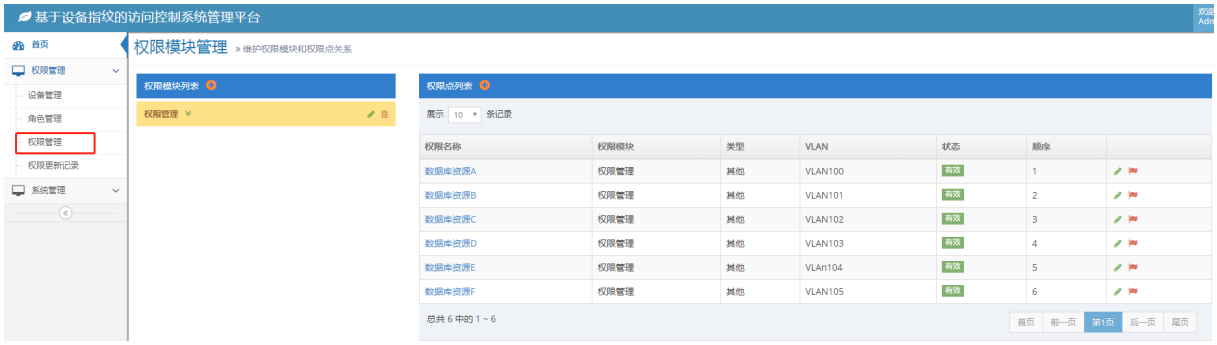


图 5.18: 权限管理页面

图 5.19: 增加权限点

图 5.20: 编辑权限点

5.7 本章小结

本文将从逻辑视图、过程视图、实现视图、物理视图和场景视图介绍系统的设计与实现过程。在逻辑视图中，介绍了系统的四层架构（物理层、控制层、数据层和应用层）。同时定义了系统相关的数据，并用 E-R 关系图描述了数据的存储结果。在过程视图中，给出了系统数据的流转描述，以及系统各个组件之间的时序描述。在实现视图中，给出了基于 VLAN 的权限划分规则和策略实施过程。在物理视图中，由于系统存在多个网络节点，因此给出了系统的物理网络部署视图。最后在场景视图中，对可视化的 web 配置中心进行展示，并详细描述了各个功能面板。

第六章 总结与展望

6.1 本文工作总结

随着物联网设备大规模接入网络，网络访问控制管理变得愈加重要。这些问题本可以通过访问控制技术得以缓解，但是传统的身份认证机制由于其复杂性而不再适用于计算和存储资源受限制的物联网设备。设备指纹技术为上述问题提供了新的解决思路。然而，现有的设备指纹识别算法无法对来自同一生产商的相似信号的设备实现有效的识别。其主要原因是这些设备在硬件、固件和软件上的相似性。基于此，本文提出了TSMC-SVM 算法，该算法将修正余弦相似性引入多分类模型。实验表明，TSMC-SVM 的平均识别精度达到 93.2‰。不幸的是，本文的设备指纹不具备正真的唯一性，即只能识别设备类型而无法识别设备个体。为了实现对设备个体的访问控制管理，本文提出了基于设备行为可信度的访问控制模型。该模型依据设备当前的网路行为和历史行为之间的偏离程度来评估设备的可信程度。模糊综合评价方法被用于计算上述的设备行为可信度。

本文首先分析了物联网安全威胁的态势以及传统访问控制系统在物联网环境中的弱势。接着介绍了设备指纹技术为物联网安全带来了新的机遇，并分析了设备指纹在国内外的研究进展以及存在的问题。然而，现有的设备指纹识别算法无法对来自同一生产商的相似信号的设备实现有效的识别。其主要原因是这些设备在硬件、固件和软件上的相似性。基于此，本文提出了 TSMC-SVM 算法，该算法将修正余弦相似性引入多分类模型。实验表明，TSMC-SVM 的平均识别精度达到 93.2‰ 同时为降低相似度匹配算法的时间复杂度，提出了一种样本预处理方法。该方法将相似度匹配算法的时间复杂度从 $\mathcal{O}(nm)$ 降低到了 $\mathcal{O}(n)$。

针对本文的设备指纹技术只能完成对设备类型的识别，而无法实现对设备个体识别的问题，提出了一种基于设备行为可信度的访问控制模型。该模型基于设备当前网络行为和历史行为的偏离程度评估其信任程度。该评估过程首先通过从设备网络行为的不同维度提取评价因子，然后利用模糊综合评价方法综合评估多个评价因子得到最终的行为可信度。其中评价因子分别为上行流量端口熵（Downlink Traffic Port Entropy, DTPE），下行流量端口熵（Uplink Traffic Port Entropy，UTPE），上行流量 IP 熵（Uplink Traffic IP Entropy，UTIE），TCP 连接密度（Connection Density Trust，CDT）和历史信任（History Trust，HT）。

为实现通过交换机完成对设备的权限管理控制，本文提出了一种基于 VLAN 的权限管理策略。在一定的规则指导下，该权限管理策略通过 VLAN 将资源划分到不同 VLAN 中，然后依据设备指纹和设备行为可信度知识调整设备与 VLAN 的隶属关系，以实现

设备对资源访问的管理。

本文的最后完成了对基于设备指纹和行为可信的物联网访问控制系统的设计和实现。并从逻辑视图、过程视图、实现视图、物理视图和场景视图介绍系统的设计与实现过程。

6.2 未来研究展望

本文设计并实现了基于设备指纹和行为可信的物联网访问控制系统，有效解决了现有的基于复杂加密协议和认证机制的访问控制技术无法适用于计算和存储资源受限制的物联网设备的问题。但是本文的系统还存在不足之处，有待后续改进：

1. 本文研究了 27 种设备类型，后续的工作是扩展设备类型数据集。同时在实验过程中也观察到了随着设备类型的增加，平均识别准确率在不断下降。而且当设备类型达到 27 种时，平均识别准确率降到了 60%。本文后续的工作是缓解该问题，使得算法能够接受更过的设备类型。
2. 在机器学习中，最终的学习结果严重依赖于特征提取工作。虽然本文从报文特征和流量统计特征两方面展开特征提取工作。但是缺少对不同特征对学习结果影响的研究，本文的后续工作将对此问题展开研究。
3. 本文提出了基于设备行为可信度的访问控制模型。虽然本文提取了流量端口熵，下行流量端口熵，上行流量 IP 熵，TCP 连接密度和历史信任以对最终的设备网络行为进行综合评价。但是针对设备行为的异常判断可以从更多的维度去综合分析。本文的后续工作是不断完善评价因子，以生成对设备行为更全面的刻画。

致谢

回首往昔，感慨万千。三年的研究生生活即将结束，此时的心情更多的是不舍与期待。在此，我想对所有给予我帮助、关系和支持的表达感激之情。

首先，我要感谢我的导师——宋宇波教授。宋老师让我学习到了什么叫做严谨求实、兢兢业业。没有宋老师耐心的指导，我也无法顺利完整这篇论文。而且宋老师幽默风趣、为人亲切，在生活上也给予我很多的帮助。在此，我要再次对宋老师致以崇高的敬意和衷心的感谢。

同时，我要感谢信息安全研究中心所有老师对我在专业学习上帮助。是你们为我进入信息安全研究领域打下了坚实的基础。

其次，我要感谢实验室的师兄师姐和师弟师妹们。是你们让我的研究生生活充满收获和欢乐！

最后，感谢在百忙之中抽出时间参与论文审阅与答辩的各位专家与老师，谢谢你们!

参考文献

[1] IoT: the next wave of connectivity and services[EB/OL]. https://www.gsmaintelligence.com/research/2018/04/iot-thenext-wave-of-connectivity-and-services/665/. Accessed March 14, 2019.

[2] GIV 2025[EB/OL]. https://www.huawei.com/minisite/giv/cn/. Accessed March 14, 2019.

[3] Gartner Says Worldwide IoT Security Spending Will Reach $1.5 Billion in 2018[EB/OL]. https://www.gartner.com/en/newsroom/press-releases/2018-03-21-gartner-says-worldwide-iot-security-spending-will-reach-1-point-5-billion-in-2018. Accessed March 14, 2019.

[4] Safety alert: see how easy it is for almost anyone to hack your child's connected toys[EB/OL]. https://www.which.co.uk/news/2017/11/safety-alert-see-how-easy-it-is-for-almost-anyone-to-hack-your-childs-connected-toys/. Accessed March 14, 2019.

[5] Consumer Alert: Consumer Affairs Warns Parents to Secure Video Baby Monitors[EB/OL]. https://www1.nyc.gov/site/dca/media/pr012716.page. Accessed March 14, 2019.

[6] Bertino, Elisa and Islam, Nayeem. Botnets and internet of things security[J]. Computer, 2017, (2):76–79.

[7] Gao, Ke, Corbett, Cherita, and Beyah, Raheem. A passive approach to wireless device fingerprinting[C]. In: 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN). 2010. 383–392.

[8] Shone, Nathan, Ngoc, Tran Nguyen, Phai, Vu Dinh, et al. A deep learning approach to network intrusion detection[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, 2(1):41–50.

[9] Wu, Mingtao, Song, Zhengyi, and Moon, Young B. Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods[J]. Journal of intelligent manufacturing, 2019, 30(3):1111–1123.

[10] Yin, Chuanlong, Zhu, Yuefei, Fei, Jinlong, et al. A deep learning approach for intrusion detection using recurrent neural networks[J]. Ieee Access, 2017, 5:21954–21961.

[11] Meshram, Ankush and Haas, Christian. Anomaly detection in industrial networks using machine learning: a roadmap[EB/OL]. 2017.

[12] Anderson, Blake and McGrew, David. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity[C]. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017. 1723–1732.

[13] Perera, Pramitha, Tian, Yu-Chu, Fidge, Colin, et al. A comparison of supervised machine learning algorithms for classification of communications network traffic[C]. In: International Conference on Neural Information Processing. 2017. 445–454.

[14] Sun, Guanglu, Liang, Lili, Chen, Teng, et al. Network traffic classification based on transfer learning[J]. Computers & electrical engineering, 2018, 69:920–927.

[15] Lotfollahi, Mohammad, Zade, Ramin Shirali Hossein, Siavoshani, Mahdi Jafari, et al. Deep packet: A novel approach for encrypted traffic classification using deep learning[J]. arXiv preprint arXiv:1709.02656, 2017.

[16] Orsolic, Irena, Pevec, Dario, Suznjevic, Mirko, et al. A machine learning approach to classifying YouTube QoE based on encrypted network traffic[J]. Multimedia tools and applications, 2017, 76(21):22267–22301.

[17] Nmap - free security scanner for network exploration & security audits[EB/OL]. https://nmap.org/. Accessed March 14, 2019.

[18] Xprobe - active OS fingerprinting tool[EB/OL]. https://sourceforge.net/projects/xprobe/. Accessed March 14, 2019.

[19] Coppi, Renato, Gil, Maria A, and Kiers, Henk AL. The fuzzy approach to statistical analysis[J]. Computational statistics & data analysis, 2006, 51(1):1–14.

[20] Bratus, Sergey, Cornelius, Cory, Kotz, David, et al. Active behavioral fingerprinting of wireless devices[C]. In: Proceedings of the first ACM conference on Wireless network security. 2008. 56–61.

[21] Sieka, Bartlomiej. Active fingerprinting of 802.11 devices by timing analysis[C]. In: CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006. 2006. 15–19.

[22] Corbett, Cherita L, Beyah, Raheem A, and Copeland, John A. Passive classification of wireless NICs during active scanning[J]. International Journal of Information Security, 2008, 7(5):335–348.

[23] Formby, David, Srinivasan, Preethi, Leonard, Andrew, et al. Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems.[C]. In: NDSS. 2016.

[24] Arackaparambil, Chrisil, Bratus, Sergey, Shubina, Anna, et al. On the reliability of wireless fingerprinting using clock skews[C]. In: Proceedings of the third ACM conference on Wireless network security. 2010. 169–174.

[25] François, Jérôme, Abdelnur, Humberto, Festor, Olivier, et al. Automated behavioral fingerprinting[C]. In: International Workshop on Recent Advances in Intrusion Detection. 2009. 182–201.

[26] Franklin, Jason, McCoy, Damon, Tabriz, Parisa, et al. Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting.[C]. In: USENIX Security Symposium. 2006. 16–89.

[27] Pang, Jeffrey, Greenstein, Ben, Gummadi, Ramakrishna, et al. 802.11 user fingerprinting[C]. In: Proceedings of the 13th annual ACM international conference on Mobile computing and networking. 2007. 99–110.

[28] Jana, Suman and Kasera, Sneha K. On fast and accurate detection of unauthorized wireless access points using clock skews[J]. IEEE Transactions on Mobile Computing, 2010, 9(3):449–462.

[29] Brik, Vladimir, Banerjee, Suman, Gruteser, Marco, et al. Wireless device identification with radiometric signatures[C]. In: Proceedings of the 14th ACM international conference on Mobile computing and networking. 2008. 116–127.

[30] Radhakrishnan, Sakthi Vignesh, Uluagac, A Selcuk, and Beyah, Raheem. GTID: A Technique for Physical DeviceandDevice Type Fingerprinting[J]. IEEE Transactions on Dependable and Secure Computing, 2015, 12(5):519–532.

[31] Yang, Kai, Li, Qiang, and Sun, Limin. Towards automatic fingerprinting of IoT devices in the cyberspace[J]. Computer Networks, 2019, 148:318–327.

[32] Aneja, Sandhya, Aneja, Nagender, and Islam, Md Shohidul. IoT Device Fingerprint using Deep Learning[C]. In: 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS). 2018. 174–179.

[33] Miettinen, Markus, Marchal, Samuel, Hafeez, Ibbad, et al. IoT Sentinel: Automated device-type identification for security enforcement in IoT[C]. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017. 2177–2184.

[34] Aluthge, Nishadh et al. IoT device fingerprinting with sequence-based features[J]. 2018.

[35] Siby, Sandra, Maiti, Rajib Ranjan, and Tippenhauer, Nils. Iotscanner: Detecting and classifying privacy threats in iot neighborhoods[J]. arXiv preprint arXiv:1701.05007, 2017.

[36] Kurtz, Andreas, Gascon, Hugo, Becker, Tobias, et al. Fingerprinting mobile devices using personalized configurations[J]. Proceedings on Privacy Enhancing Technologies, 2016, 2016(1):4–19.

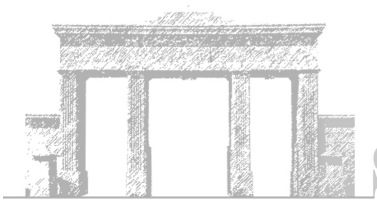
[37] Van Goethem, Tom, Scheepers, Wout, Preuveneers, Davy, et al. Accelerometer-based device fingerprinting for multi-factor mobile authentication[C]. In: International Symposium on Engineering Secure Software and Systems. 2016. 106–121.

作者攻读硕士学位期间的研究成果

发表的论文

[1] 第一作者，“A Booting Fingerprint of Device for Network Access Control”，International Conference on Circuits, System and Simulation (ICCSS 2019)，2019 年 4 月。

心求至善



SOUTHEAST UNIVERSITY