中图分类号：**TP391.4**

论 文 编 号：**10006SY2039125**

北京航空航天大學

硕 士 学 位 论 文

基于上下文分析的智能家居事件安全性研究

作 者 姓 名　徐骁赫

学 科 专 业　网络空间安全

指 导 教 师　毛剑　　副教授

培 养 院 系　网络空间安全学院

[– This page is a preset empty page –]

Context-based smart home event security analysis

A Dissertation Submitted for the Degree of Master

Candidate : Xu Xiaohe

Supervisor: Prof. Mao Jian

School of Cyber Science and Technology

Beihang University, Beijing, China

[– This page is a preset empty page –]

中图分类号：**TP391.4**

论 文 编 号：**10006SY2039125**

硕 士 学 位 论 文

基于上下文分析的智能家居事件安全性研究

作者姓名　徐骁赫　　申请学位级别　工学硕士

指导教师姓名　毛剑　　职　　称　　副教授

学科专业　网络空间安全　　研究方向　　物联网安全

学习时间自　2020 年　09 月　01 日　　起至　　2023 年　01 月　15 日止

论文提交日期　2018 年　01 月　10 日　论文答辩日期　2018 年　03 月　01 日

学位授予单位　北京航空航天大学　　学位授予日期　　年　　月　　日

[– This page is a preset empty page –]

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期： 年 月 日

学位论文使用授权

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期： 年 月 日

指导教师签名：_____ 日期： 年 月 日

[– This page is a preset empty page –]

摘　要

摘要是学位论文内容的简短陈述，应体现论文工作的核心思想。论文摘要应力求语言精炼准确。博士学位论文的中文摘要一般约800～1200字；硕士学位论文的中文摘要一般约500字。摘要内容应涉及本项科研工作的目的和意义、研究思想和方法、研究成果和结论。博士学位论文必须突出论文的创造性成果，硕士学位论文必须突出论文的新见解。

关键字是为用户查找文献，从文中选取出来揭示全文主体内容的一组词语或术语，应尽量采用词表中的规范词（参考相应的技术术语标准）。关键词一般3～5个，按词条的外延层次排列（外延大的排在前面）。关键词之间用逗号分开，最后一个关键词后不打标点符号。

为了国际交流的需要，论文必须有英文摘要。英文摘要的内容及关键词应与中文摘要及关键词一致，要符合英语语法，语句通顺，文字流畅。英文和汉语拼音一律为Times New Roman体，字号与中文摘要相同。

关键字： 北航，学位论文，博士，硕士，中文

Abstract

What were you doing 500 years ago? Oh, that's right nothing, because you didn't exist yet. In fact, several generations of your family had yet to leave their mark on the world, but one very special shark may already have been swimming in the chilly North Atlantic at that time, and the incredible animal is somehow still alive today.

Scientists studying Greenland sharks observed the particularly old specimen just recently, and after studying it they've determined that the creature is approximately 272 to 512 years old. That's an absolutely insane figure, and if its age lands towards the higher end, it makes the animal the oldest observed living vertebrate on the entire planet.

Greenland sharks are an incredible species in a number of ways, but most notable is its longevity. The sharks are well over 100 years old before even reaching sexual maturity, and regularly live for centuries. This particularly old specimen, along with 27 others, were analyzed using radiocarbon dating. The reading came back at around 392 years, but potential margin of error means the animal's true age is somewhere between 272 and 512.

The shark, which is a female, measures an impressive 18 feet long. That's pretty large, but it might not sound particularly large for an ocean-dwelling creature that lives hundreds of years. That is, until you consider that the Greenland shark only grows around one centimeter per year. With that in mind, 18 feet is actually downright massive.

As for how this particular shark species manages to live so incredibly long, scientists attribute a lot of its longevity to its sluggish metabolism, as well as its environment. The frigid waters where the sharks thrive is thought to increase overall lifespan in a variety of ways. Past research has shown that cold environments can help slow aging, and these centuries-old sharks are most certainly benefiting from their chilly surroundings.

— Online news *Scientists find incredible shark that may be over 500 years old and still kicking*, 12.16.2017. (http://bgr.com/2017/12/14/oldest-shark-greenland-512-years-old/).

Key words: News, BGR, Shark

目　录

图 清 单

[– This page is a preset empty page –]

表 清 单

[– This page is a preset empty page –]

主要符号表

E 能量

m 质量

c 光速

P 概率

T 时间

v 速度

[– This page is a preset empty page –]

第一章　绪论

1.1　课题来源

国家自然科学基金面上项目：基于多源事件复合推演的物联网安全溯源与异常检测机理研究。

1.2　研究背景

智能家居（Smart Home）（又称为自动化家居、智慧家居）这一概念在1984年由American Association of House Builders提出[1]，是一种物联网（Internet of Things，IoT）的典型应用场景，由一系列控制家居环境的智能设备组成。起初智能家居被用于照明和供暖、制冷等系统的自动控制，而随着技术发展，目前智能家居基本涉及到了用户房屋内几乎所有种类的部件（包括门窗、电灯、智能音箱、各种开关、各种传感器等等）。此外，智能家居可以实现对房屋环境的实时监控，以及对智能设备的远程控制，用户以直接独立操作设备或设定自动化规则的形式参与其中[2]。通常情况下，智能家居提供的功能包括舒适度、安全性、可靠性、远程控制能力和能源节约能力[3]。

从结构上来讲，智能家居可用以下三层结构来描述[4]：感知层、网络层、应用层。感知层包括智能设备和传感器，用于感知物理环境的状态；网络层包括网关、移动设备和服务器等硬件设施，通过构建家庭网络来为设备间通信提供可能；应用层包括在移动设备或服务器上运行的App等形式的智能逻辑，负责提供用户UI接口，或通过自动化规则进行智能决策。作为智能家居系统运转的基础，感知能力保证了家居环境状态能够被各类传感器和设备实时收集，从而进一步为用户和应用层逻辑（通常以App的形式）提供准确信息以供决策。现有大多数成熟的智能家居平台（如SmartThings[6]与Home Assistant[7]）为事件驱动的（Event-driven），即通过事件实现信息传递，其中事件（Event）是由设备发出的一种网络消息，用于描述相关的设备状态或环境信息[2]。

智能家居系统通过事件总线（Event Bus）[7]来收集汇总事件，通过发布（Publish）-订阅（Subscribe）机制来控制事件的流向，设备通过发布操作向事件总线添加新事件，应用层逻辑通过订阅操作在事件总线中登记，当有相应事件的发布时，事件总线将这

一消息转发给所有订阅过此事件的应用层逻辑[2]。图1为一个简单的智能家居事件示意图，当智能设备“灯”被用户或被系统远程打开时，其会向智能家居云发布一个“开灯”事件，云中运行的App通过订阅此事件来感知智能电灯的状态，从而进一步做出智能决策。总体来说，事件是现实生活中发生事件的抽象表示，并将物理空间中的智能设备实体映射到网络空间中，使设备能够参与智能决策的过程、提供信息或接受指令[8]，是智能家居系统的核心要素。

事件是智能家居实现感知能力的基础，然而事件存在诸多安全隐患，这些安全隐患可能导致系统出现逻辑上的异常，从而出现意外行为；或是被攻击者利用，使敏感设备受控。文献[9]整理总结了智能家居事件可能出现的安全问题，其中包括事件丢失、事件截断、事件错误、虚假事件等。文献[10]以SmartThings为例指出，事件消息是缺乏保护的，一旦攻击者得到设备ID等敏感信息便可轻易地伪造事件消息，此攻击称为事件欺骗攻击（Event Spoofing Attack）。此研究进一步指出，绝大多数设备均会向系统发布事件，这些事件均会面临被伪造的风险。事件欺骗攻击的实施手段多种多样，包括节点妥协[15]、恶意App植入[16]、中间人攻击[17]、流量包伪造[18]以及超声波语音助手攻击[19][20][21]等攻击方式，故对攻击者而言非常容易实施。

上述伪造的虚假事件会造成系统对设备状态的错误感知，导致系统执行错误的决策，而通过伪造事件来有目的性地触发自动化规则，攻击者可以在未经授权的前提下控制一些敏感设备。例如，攻击者可以利用“用户到家则解锁门锁”的自动化规则，向系统注入“到家”的虚假事件，从而使门锁打开。以上案例说明，诸如事件欺骗攻击一类的攻击可以使系统产生虚假的事件，从而使系统自动化规则被意外触发，使敏感设备受控。

除此之外，由于智能家居的使用环境以及用户或App设置的关联规则的影响，事件之间存在错综复杂的因果性触发规律，构成了极为复杂的事件依赖关系。其中跨应用程序干扰[11][12]由不同App对同一设备进行同时操作引起，会造成命令冲突，引起意外的系统行为，使命令未按照用户期望执行；而危险事件依赖[13][14]可能来源于某些隐式事件依赖，进而引起危险的触发动作，造成敏感设备受控，或家居环境的危险物理状态，如扫地机器人触发动作传感器，进而使系统认为有人在家，进而使门锁打开。上述两种安全隐患的共同之处在于，其均违背了智能家居的正常运行模式，故本文将上述两种安全隐患统称为“异常事件依赖”。

综上所述，智能家居事件的安全问题会造成严重的安全后果，故针对智能家居事件安全的研究是至关重要的。本文拟从上述讨论触发，从虚假事件、异常事件依赖两方面安全问题出发，展开相关虚假事件检测、事件依赖提取、异常事件依赖检测方面的研究，旨在提高智能家居事件的安全性，保证智能家居的正确感知能力与逻辑正确性，从而为整体系统的安全运行提供基础保障。

1.3 研究意义

针对上述讨论，本论文的研究意义主要表现在以下三方面：

1. 为了解决虚假事件问题，有效检测事件欺骗攻击，本论文研究一种基于物理上下文的事件真实性验证方案，对现有方案的准确率进行改进，同时考虑到实际部署情况，通过缩减传感器数量达到轻量化的目的；
2. 为了对事件依赖进行全面提取和展示，本论文研究一种基于复合上下文建模的事件依赖图构建方案，实现了事件依赖的全面、准确提取和直观表示，提升用户对系统的感知能力；
3. 为了检测异常事件依赖，本论文研究一种基于事件依赖特征的异常检测方法，保证了系统中的事件依赖关系在语义上是准确的、安全的。

1.4 本文研究内容以及论文构成

本文基于智能家居的上下文信息，研究事件安全性相关问题，旨在提升智能家居系统的安全性。具体来说，考虑可能出现的虚假事件，设计并实现一种基于物理上下文的事件真实性验证方法，保证智能家居事件的真实性；考虑可能存在的异常事件依赖，设计并实现基于复合上下文建模的事件依赖图构建方法，以及基于事件依赖特征的异常检测方法，实现事件依赖的全面提取与静态异常检测，分析系统可能出现的安全隐患，提高智能家居安全性。

本论文的研究内容如下：

1. 智能家居自动化数据收集实物平台搭建：
2. 基于物理上下文的事件真实性验证：研究通过智能家居事件物理上下文建立事件指纹的方法，并利用指纹实现事件验证，优化现有方案效果；
3. 基于复合上下文建模的事件依赖图构建：研究通过逻辑上下文提取显式事件依赖

的方法，研究通过事件物理上下文的事件物理关系建模方法，利用物理关系提取隐式事件依赖，并实现事件依赖图的构建；

4. 基于事件依赖特征的异常检测：研究事件依赖图结构特征的提取方法，研究事件依赖语义的表达方法以及语义特征提取方法，将事件依赖向量化，研究用于事件依赖异常检测的机器学习模型，最终实现基于事件依赖特征的异常检测。

第二章　智能家居事件安全性现状分析

本章对智能家居事件安全相关研究做一整理总结。其中第一节介绍单一事件本身的安全问题相关研究，第二节介绍事件依赖安全问题相关研究。

2.1　事件安全问题相关研究

2.1.1　事件安全问题总结

文献[16]基于SmartThings用户论坛以及相关安全研究整理总结了智能家居系统可能出现的安全问题。其中包括事件丢失、事件截断、事件错误、虚假事件等。其中事件丢失、事件截断指由于网络或攻击者等原因造成了丢包，从而使控制系统不能实时准确地感知设备，事件错误指由于设备自身故障造成错误事件的产生，虚假事件指恶意攻击者通过某种攻击手段进行了事件注入。

文献[14]提到，SmartThings没有对产生事件的过程做访问控制，也没有对SmartApps提供验证事件完整性的方法，故所有SmartApp或SmartDevice都有权限发出任意事件通知。基于这一发现，恶意app可以向Hub发送错误的事件通知，从而导致错误的自动化规则被触发；或者可以阻止发送正确的事件通知，导致正确的规则没有触发。这一攻击方式被称作事件欺骗。文中还指出，此种攻击方式可以让恶意app越权控制其它设备。如恶意app可以对位置事件进行篡改，在用户在外时发送回家的物理事件，从而触发所有订阅此事件的app做出相应改变，导致门锁打开等规则的触发，达到越权控制的目的。

尽管本文使用SmartThings中的SmartApp说明事件欺骗攻击是如何产生的，且SmartApp已经不是SmartThings自动化规则的主流实现方式，但现实系统中的事件不仅仅来源于App发送的事件通知。文献[17][18][19]给出了一些通过改变物理环境来影响智能设备的感知，进而向智能家居系统中注入虚假事件的攻击手段。在这些攻击中，攻击者通过注入超声波来潜在地改变智能家居系统的物理环境，使得智能音箱感知到一个错误的事件，由于人耳无法感知超声波，故用户完全无法察觉攻击的发生。

文献[13]总结了通过事件欺骗攻击来达成潜在攻击目标的方式。攻击者的攻击目标、攻击设备多样，后果严重。但文中同样提出这些设备可以被附近的传感器感知，从而用作事件验证。如门的状态可以被附近的加速传感器、光线传感器、气压传感器和

麦克风感知，这些传感器就可以用作有关开门、关门事件的验证。

为了识别错误事件或虚假事件，可以为每个事件制定指纹。每个指纹与现实生活中的事件一一对应。当Hub收到app或设备的事件通知时，可以通过验证指纹信息确定事件在现实生活中是否真实发生，从而验证事件的正确性。以下展开介绍事件指纹的相关研究。

2.1.2　事件指纹相关研究

由于本文着重研究有关错误或虚假事件的防御方案，即事件指纹构建方案，故本节列举并介绍了现有事件指纹相关研究。目前关于构建事件指纹的方法主要分为两种：基于网络流量和基于传感器数据。

在基于网络流量构建事件指纹方面，文献[20]提出了一种用加密的网络流量检测家居设备身份和行为的方法，基于此思路攻击者可以获得用户的隐私信息。论文在识别设备身份方面，使用ZigBee网络流量包的统计特征（包括平均包长度、平均到达间隔和包长度的标准差）作为分类依据，选择算法为K近邻分类。在识别设备行为方面，作者基于活动设备的数据包发送率会显著增加这一想法，将数据包序列划分为长度为W的窗口，提取统计特征，作为监督学习的训练数据集，算法分别选用随机森林分类和K近邻分类，两者效果相似。作者还对设备行为的具体分类和用户行为分类进行了研究，其基本思路相近，均为提取流量包的相关特征，以机器学习算法进行分类。基于上述四个步骤，攻击者可以对加密流量进行分析，获得被攻击者家居环境内设备、用户行为的信息，造成隐私泄露。

文献[3]介绍了Homonit系统，此系统同样是基于网络流量的物理事件指纹构建。与上述文献的工作不同的是，此文献的目的是构建监视系统，识别来自app的错误事件通知。作者认为所有智能家居app的行为都遵循DFA（Deterministic Finite Automation，确定有穷自动机）模型，自动机的每个状态代表app和响应家居设备的状态，自动机的状态转换代表app和设备的交互。作者对app行为构建自动机，从无线通信环境中提取事件特征作为指纹。作者指出，此系统是独立于智能家居系统之外的第三方监视系统，在提高系统安全性的同时未对系统做任何修改。但本系统仍然不能在智能家居系统已经被攻击的情况下正确工作，这是由于若设备被控制，它发出的网络信号则不具有正确性。故此系统不能识别来自设备的错误事件通知。

文献[21]提出了PingPong，可以自动从加密网络流量中提取设备事件的数据包级签名。文中指出，流量包长度的唯一序列通常可以描述某些特定简单事件，并可以利用他们来作为唯一事件指纹。首先，PingPong使用Android Debug Bridge（ADB）和shell脚本在智能手机屏幕上模拟触屏输入，来模拟用户发起的设备事件，同时在路由器上使用tcpdump抓包。之后对抓到的数据包进行过滤，只保留源或目的IP为智能家居设备的，同时选择一个时间窗，只保留事件发生后一小范围内的包，然后过滤掉所有TCP或TLS的控制包（如TCP握手挥手和TLS密钥协商）。之后通过构建数据包对，并使用无监督学习算法提取与事件相关的数据包对，并通过相关数据包对构建事件数据包级签名。

在基于传感器数据的指纹方面，文献IoT-CAD提出了IoT-CAD，通过IoT系统中的传感器来捕捉物理环境状态从而识别异常。通过在IoT系统中安装持续性监测的传感器，并定期在时间窗口内产生数据快照，构建快照向量，来作为指纹描述此时刻IoT系统的状态。

文献[22]指出，尽管智能家居内的传感器具有异构性，但它们受相同的物理事件影响都有一定的表现方式。文献[23]提出了基于异构传感器数据来获得事件指纹的方法。作者收集除了摄像头视频数据之外全部传感器的数据，之后提供了两种机器学习的方法：人工标定训练数据集的标签，然后进行两层的SVM分类；缺少人工标定的标签，则进行非监督式机器学习，首先对数据降维，之后使用最大期望算法进行聚类。

文献[13]设计了一个办公室的日常场景，使用树莓派自行构建异构传感器体系，收集了时间跨度为两周的传感器数据，并记录物理世界中发生的真实事件为其标签，作为监督学习的训练数据集。由于收集的时间数据是连续的，作者使用网格搜索选定了针对不同事件的时间窗，提取各个传感器与事件的相对互信息量作为指纹特征，并选择指纹特征大的数据进行SVM分类。这一指纹选取方案可以保证在智能设备被控制时仍有效地检测事件欺骗攻击。

2.1.3 现有研究不足

表 1总结了上述提到的智能家居事件指纹相关研究。总体来说，现有研究的方法大多局限于从特征工程到机器学习的方案，将指纹数据来源直接看做一个大型数据集，从其中提取并筛选特征，然后选用适当的学习算法。

表 1 智能家居事件指纹相关研究总结

指纹来源	文献	网络流量包特征				传感器特征		学习算法
		TS	L	AD	P	TD	FD	
网络流量数据	Peek-a-boo	✓	✓	×	×	-	-	KNN
	Homonit	✓	✓	✓	×	-	-	DFA
	PingPong	✓	×	×	✓	-	-	DBSCAN
传感器数据	IoT-CAD	-	-	-	-	✓	×	RNN
	Synthetic sensors	-	-	-	-	✓	✓	SVM/EM
	Peeves	-	-	-	-	✓	×	SVM

注: TS–时间戳; L–包长度; AD–地址; P–协议; TD–时域; FD–频域

然而，由于智能家居系统中事件的多样性，使用一套通用方法论很难适用于所有事件。例如，Peeves[13]对开门、关门等具有瞬时影响的事件的准确率几乎达到了100%，然而这一效果无法泛化到开加热器等具有持续性影响的事件。

除此之外，某些特征工程的手段可能并不适用于智能家居场景。具体来说，减小特征数量的手段可以分为两种，特征选择（feature selection）和特征抽取（feature extraction），前者是选取原始特征集合的一个有效子集，后者是将原始高维特征空间映射到一个低维空间上。对智能家居场景来说，使用特征选择的好处在于可以减小模型线上验证时需要的数据来源，如使用特征选择的手段对传感器数据特征进行筛选，可以减小用于事件真实性验证的传感器数量，减小部署成本。而特征抽取手段（如Synthetic sensors[23]的自动编码器）没有这一优势。

2.2　事件依赖安全问题研究

2.2.1　事件依赖提取与表示

由于智能家居系统的事件依赖由物理环境和用户设置的自动化规则决定，故不同场景下的事件依赖是截然不同的。为了研究某智能家居系统的事件依赖安全问题，需要先进行此特定系统事件依赖的提取和表示。本节首先着重介绍现有研究中提取事件依赖的方法。

对大部分智能家居平台而言，由于显式事件依赖直接由系统中的触发-响应自动化规则直接定义，而规则通常是由用户设定，或在app中包含，故这些规则很容易通过代码分析、配置文件解析等手段获取[9][14][26][27]。如SmartThings平台的规则通常包含在SmartApps里或.json配置文件中，Home Assistant平台的规则通常包含在一个.yaml文件中。然而，隐式事件依赖在系统中没有类似代码或配置文件这样的显式定义，故需

要结合智能家居物理环境进行考虑[9][14]，或是直接在真实环境中进行动态测试来提取[13]。

在显式事件依赖提取方面，通常使用代码分析的手段，包括静态分析（AST、调用关系图等）或动态分析（代码插桩、建模、符号执行等）。以下介绍现有显式事件依赖提取相关的研究。

文献[24]提出了IoTSan，通过静态分析的方法从IoT应用程序源代码中提取依赖图（dependency graph），依据依赖关系建立状态模型。具体来说，IoTSan首先通过一个App依赖分析模块从SmartApp的groovy源码中寻找事件订阅函数subscribe()、读设备状态的API以及通过schedule()方法定义的定时任务来提取输入事件，并通过调用的API提取输出事件，从而构建完整的事件依赖。然后本研究使用Bandera Tool Set工具对源码进行代码行为建模，由于此工具只接收java代码输入，故本研究开发了一个groovy转java的翻译器，由上述步骤得到一个由Promela语言描述的代码逻辑模型。使用基于应用程序逻辑的建模方法可以更加准确、细粒度地描述智能家居中的事件依赖关系，但由于方法限制，其只能提取到应用程序内定义的规则，而无法衡量物理环境参与其中后的影响，也就是其不能对隐式事件依赖进行提取或识别。

文献[26]提出了Soteria，通过静态分析的手段验证物联网系统中的App或其他行为的安全性和正确性。Soteria使用一种中间表示（Intermediate representation，IR）来描述物联网应用程序的逻辑。具体来说，与之前介绍的静态分析类似，Soteria检测源代码中调用智能设备的函数以及定义触发-响应的函数。然后本研究通过中间表示IR提取应用程序的状态模型，包括智能设备的状态以及状态转换（即事件）。本研究通过模型检查来判断事件的行为是否符合安全属性。与IoTSan[24]类似，由于只从物联网应用程序中提取事件依赖关系，所以本研究无法对隐式事件依赖关系进行识别。

显式事件依赖可直接基于App源代码或配置文件进行提取，较为简单、直观、准确；而隐式事件依赖需要结合智能家居的物理环境进行考虑，这些方法通常需要挖掘不同事件和物理环境的语义并进行综合，通常较为复杂，不够准确。以下介绍隐式事件依赖提取相关研究。

文献[25]提出了IoTMon，来识别和分析IoT App之间的交互链，从而得到事件依赖关系。本研究首先使用静态分析方法从SmartApp的groovy源码中提取应用内交互规则，具体地，首先对App构建抽象语法树（Abstract Syntax Tree，AST），然后寻找声

明自动化规则的代码段，如SmartApp中的installed()和updated()函数，根据其中定义的事件订阅函数subscribe()以及调用的handle函数总结出触发-响应规则。之后，本研究使用NLP对SmartApp的描述做分词和词性标注工作，提取其中的名词，并对这些名词作聚类，选择每个类别中具有代表性的进行人工审查，最终的结果作为物理通道名（如温度、湿度）。之后根据具体事件和物理通道的关系进行不同App之间的关联，从而识别隐式的事件依赖关系。

文献[16]提出了HAWatcher，使用相关性（Correlation）的概念来表达事件与事件之间、事件与物理环境状态之间的语义信息。与上述研究类似，本研究首先SmartApp的代码进行符号执行来获取中间表示IR，并将每个app的语义都以触发-条件-响应的形式来表示。之后，本研究通过NLP对物理环境进行提取，并以假设的方式提出设备属性和物理属性之间的相关性，最后再通过事件日志对上述所有假设的相关性进行检验。

文献[27]提出了IoTGaze，通过无线流量特征和App的描述和UI构建行为基准模型，并检测运行时安全违规行为。本研究跳出物联网平台和应用程序内部的角度，从无线流量的角度重新思考IoT安全和隐私问题，通过收集无线射频环境内的通信数据包，构建基于程序的无限上下文。具体来说，本研究首先提取无线数据包的特征来将通信流量与事件相关联，然后通过发现事件时间性依赖来为物联网系统构无线上下文，最后通过用户界面（UI）提取实际用户预期的上下文，并与无线上下文进行对比，来发现物联网的行为异常。由于通过无线上下文来研究事件和事件间依赖，IoTGaze同样可以检测和识别到隐式事件依赖。

除常规的基于语义和机器学习的方法外，文献[15]提出了IoTSafe，通过设计一个动态测试方法来对物联网系统中的显式和隐式交互行为进行准确挖掘。与IoTMon类似，本研究首先通过代码分析提取App内交互规则，并通过NLP进行分词、聚类、人工审查，提取物理通道信息。然后设计动态测试样例的生成方法，目的是遍历地测试所有设备的所有运行状态对物理环境的影响，同时对于单房间不同设备的并行和多房间并行测试做出优化，减少动态测试时间。通过动态测试来补充设备与环境、环境与设备之间的交互关系，从而完善交互规则。IoTSafe是目前第一个使用动态测试的方法对事件依赖关系进行提取的研究，其优点是能够直接准确地提取传统静态分析方法难以提取到的隐式事件依赖，缺点是由于直接使用现实环境进行测试，若无敏感设备或状态的限制，可能会对用户的人身或物理财产造成影响。

除此之外，文献[28]提出了Hєlion，通过识别用户驱动的家居自动化规则序列的规律性，来通过语义的方式表达用户活动规律。基于人类使用的语言通常是可预测的这一假设，本研究使用统计语言模型，来通过已发生的事件预测未来发生事件的可能性。本研究首先通过用户调查的方式收集以某些特定结构表达的用户使用规律，之后通过解析上述自然语言得到事件令牌，然后使用一个n-gram方法来学习用户在智能家居自动化规则中的规律性。

在事件依赖的可视化表达方面，IoTMon[14]给出了自己的表达方案，使用有向图（文中称为交互链图）来表达App内部和App之间的依赖关系。其中，图的节点有许多类型，包括触发（形式可以是事件、传感器）、响应（形式为事件）、物理通道等，尽管以人的角度实现了一条链的可读性，但结点的类型过多也复杂化了此有向图。此外，HAWatcher[16]将智能家居系统的相关性分为两种：事件到事件，以及事件到状态。具体来说，事件到事件的相关性表示为$E \rightarrow E$，即某一事件触发另一事件的发生；事件到状态的相关性表示为$E \rightarrow S$，即某一事件的发生导致了某传感器的数值以某规律变化。从具体含义上来看，HAWatcher对相关性的定义与本研究对事件依赖的定义类似。

本论文拟在使用一种更统一、准确的表达方式，通过统一有向图节点的类型，并在图的边中加入信息予以辅助说明，来实现有向图的简化，完善事件依赖表达方案，并与后续的异常事件依赖检测进行对接。

2.2.2 异常事件依赖检测

异常事件依赖指超出代码原本逻辑或超出用户期望的事件依赖，其可能会破坏智能家居所在的物理环境，造成安全隐患；或被攻击者利用，通过事件依赖链发起跨事件的攻击。异常既可以由用户自身造成（如误配置或规则冲突等），也可以由潜在的攻击者造成（如恶意代码等）。为了检测这种超出用户期望的异常事件依赖，现有研究对智能家居系统中的正常行为进行建模，利用机器学习方法进行分类或聚类进行异常范围的划分[14][30]；或进行程序级别行为的精确建模，然后动态地监控系统或代码的行为，对基线之外的事件依赖判定为异常[9][26][28][31]。

文献[29]提出6thSense，一种基于上下文感知的入侵检测系统，用于全面检测智能设备中基于传感器的攻击手段。本研究通过实时观察传感器数据，并根据智能设备确定当前的传感器是否是恶意的，来确定设备的上下文；同时，本研究通过观察用户不

同任务的传感器数据变化，将传感器数据与用户活动相关联；然后该方法使用了三种基于机器学习的检测机制（马尔科夫链、朴素贝叶斯和LMT）进行基于传感器的恶意行为的检测。本研究主要侧重于通过传感器来对正常行为进行建模，来上层地、通用地检测一些攻击手段，而非针对事件依赖异常的检测。

IoTMon[25]通过之前提取的应用程序交互关系来检测可能会对物理环境造成危险影响的交互关系，首先对所有的交互进行向量化，然后进行行为建模和聚类，将离群的交互行为认为是有危险性的。具体来说，IoTMon首先分析了不同物理通道的出现频率；然后从出现频率最高的物理通道开始，循环地为与当前物理通道的相关性最高的物理通道赋递增、等间隔的值，其中相关性由共同出现的频率计算；然后对每个事件依赖构建向量，每一维代表其对某物理通道的影响。从结果上来看，此方法有着77%的精准率，其效果不是很理想的原因主要在于构建的事件依赖向量过于简单，每一维的值是固定的。且本研究只对单一事件依赖进行异常的检测，而并没有考虑多层事件依赖可能带来的危险。

除上述静态异常检测外，一些研究也提出了动态运行时的异常检测方法。IoTSan[24]对IoT中可能出现的安全问题进行了检测，包括物理状态异常和敏感信息的泄露。本研究使用一个输出分析器，使用基于启发式的算法，来验证智能应用程序的安装阶段和运行阶段。在安装阶段，输出分析器枚举所有可能的配置并对实际配置进行对比，来判断应用程序是否是恶意的；在之后的阶段，新应用程序会与已安装的应用程序进行分析，观察是否有异常情况，从而界定是否安装此App。此方法能够提取由不同App带来的规则冲突或恶意规则问题，从而对用户误配置或攻击者的恶意代码注入进行一定程度的防御。

HAWatcher[16]通过提取出的智能家居相关性，设计了一个异常检测模块，此模块订阅了家居系统的所有事件，并与家居自动化系统并行运行。利用之前提取的事件、传感器的相关性作为行为模型，收集实时的智能家居设备事件，判断其前序和后序事件进行上下文检查，从而检测偏离行为模型的异常事件关联。文献[30]提出了IoTGuard，一个基于策略的物联网动态安全防护系统。本研究通过模拟应用程序的生命周期，修改IoT应用程序的源代码，收集应用程序运行时信息并存储在动态模型中，然后直接对应用程序运行环境中的不安全和不期望的状态进行阻止，通过监视IoT和“触发-响应”平台应用程序的行为，保护用户免受不安全的设备状态的影响。

文献[31]提出了HomeGuard系统，用于检测跨应用程序干扰（Cross-App Interference，CAI）问题。在安装每个新的App时，此系统都会介入，从已安装的App出发进行规则干扰的检测。具体来说，本研究通过设计实现了一个符号执行器，从SmartApps中提取规则语义信息，利用可满足型模型理论检测冲突。除此之外，本研究还将App的规则语义与用户配置的规则结合以进一步检测威胁。

文献[32]首先全面分析了出发物联网平台规则间漏洞的空间，将现有研究提出的漏洞进行分类和系统化，然后本研究提出了iRuler，使用形式化方法来检测这些漏洞。具体来说，通过可满足性模型理论（Satisfiability Modulo Theories，SMT）来进行模型检查，从而发现规则间漏洞。最后，本研究同样通过NLP来对App描述来进行处理，从而推断规则间信息。

文献[33]检测了IFTTT平台小程序（applet）可能出现的一些问题，其中IFTTT也是一个以触发-响应为自动化规则模板的智能家居平台。本研究通过两种基于URL的攻击展示了IFTTT小程序的隐私泄露问题，并对在野的IFTTT小程序进行了分析，根据其来源的敏感程度进行了分类，发现了30%的applet可能侵犯了隐私。除此之外，本文还提出了一种信息流控制的框架，中断从私密source到敏感sink的访问控制机制，同时监测小程序输出的信息流，来加强小程序的隐私保护。

对上述动态异常检测的研究而言，由于其模型刻画准确的系统行为，故其准确性强，可解释性强，但只能实时做出异常检测，设计一个通用的防御策略比较难以实现。

2.2.3 现有研究不足

表 2 智能家居事件依赖安全相关研究总结

文献	主要方法	事件依赖提取		异常事件依赖检测	
		显式	隐式	静态	动态
IoTSan[26]	代码分析、模型检查	✓	×	-	✓
Soteria[27]	代码分析、模型检查	✓	×	×	×
IoTMon[14]	代码分析、NLP、聚类	✓	✓	✓	-
HAWatcher[9]	代码分析、NLP、假设检验	✓	✓	-	✓
IoTGaze[28]	NLP、事件转移图	✓	✓	×	×
IoTSafe[13]	动态测试	✓	✓	×	×
6thSense[30]	马尔可夫链、朴素贝叶斯、LMT	×	×	✓	-
IoTGuard[31]	代码插桩、代码运行监控	✓	×	-	✓
HomeGuard[12]	代码分析、SMT	✓	×	-	✓
[32]	形式化方法、SMT	×	×	-	✓

表 2总结了现有事件依赖提取和异常依赖检测相关研究。对静态异常事件依赖检测方法而言，其优点是能够全面地分析系统中的所有事件依赖，并且在系统启动或正式

运行之前发现潜在的威胁，但缺点是由于引入通用的机器学习模型来刻画不同种类的事件依赖，其准确性严重依赖于特征工程和模型选择，而现有研究在特征工程方面没有很细粒度的设计，这是由于事件依赖本身的语义比较复杂，故难以设计特征的计算方法；而对动态异常分析而言，由于其模型刻画准确的系统行为，故其准确性强，可解释性强，但只能实时做出异常检测，设计一个通用的防御策略比较难以实现。

此外，现有研究大多数集中于两事件之间依赖的异常检测，如IoTMon[14]，没有考虑多跳的事件依赖链的异常，例如用户配置了$E_{away\ home} \rightsquigarrow E_{light\ off}$以及$E_{light\ off} \rightsquigarrow E_{window\ open}$两条事件依赖，其分别都是符合用户预期的，但若没有多余条件限制，$E_{away\ home} \rightsquigarrow E_{light\ off} \rightsquigarrow E_{window\ open}$的事件依赖链成立，则会导致用户离开家后窗户被打开，造成可能的危险情况。

第三章　小型拟真智能家居实物实验平台

3.1　小型拟真智能家居实物实验平台需求调研

3.1.1　典型开源智能家居平台介绍

3.1.2　现有公开数据集调研

近年来，智能家居越来越受到研究者的重视，学术界逐渐有一些研究者自行搭建智能家居实验环境，收集用于自己学术研究的数据集并将其公开，供后续研究使用。然而，这些学术研究的侧重点不同，导致其数据集通常不能覆盖所有智能家居的使用场景。本节调研了现有学术界面向智能家居场景的公开数据集，分析其数据特性及适用场景，为本文后续研究中的数据集选择，以及实验平台搭建需求分析做支撑。由于本文研究主要关注事件安全性问题，所用数据主要涵盖各智能设备状态流、控制中枢的事件日志、以及自动化规则等，故本节着重从上述方面进行调研。

英国牛津大学Birnbach等人在其研究[?]中收集了一组数据集并将其公开[?]，他们在一个真实办公室环境内，使用路由器、智能手机、笔记本、树莓派以及各种高精度传感器电路模组构成了覆盖12种感知能力的传感器组合，并以不同的数量、不同的位置布置在办公室的各个位置。这些传感器不仅包括温度、湿度、光照强度、声音强度等对物理通道状态的感知，还包括接触传感器、开关等对设备状态的感知。在实验持续的14天内，办公室内的用户均手动地使用屋内各种设备，各个传感器不间断收集数据，并存储在本地U盘中。实验结束后研究者对数据进行了打包并公开。数据分传感器以csv文件的形式储存，由二元组(t, v)的形式组成，其中t为时间戳，v为传感器读数。此数据集的优势在于其完全利用自制传感器，传感器汇报率、精度较高，便于进行细粒度的数据分析以及复杂机器学习模型的训练，并且其采样持续时间长，传感器类型丰富；然而，其并没有引入智能家居平台作为中枢，即其不能体现智能家居的触发-响应特性，故不适用于事件依赖的相关研究。

类似地，Chimamiwa等人[?]公开了其智能家居传感器序列数据集，该数据集从多个环境传感器收集而来，主要目的为捕获人类的日常生活活动。传感器类型包括温湿度、开关、光传感器等，收集持续时间为6个月，采样率为1Hz。同时，此数据集包括一个公寓内多个房间采集的数据。数据格式方面，传感器以数据库的形式分别存储在五个

表中，并以csv文件的形式公开。传感器数据按整型数据和浮点型数据分别存储在两个表中，剩余三个表则为传感器名称等基本信息。每一数据条目由四元组(id, t, v, s)的形式组成，其中t为时间戳，v为传感器读数，s为传感器名称。从使用场景来看，此数据集相比上述牛津大学数据集更加丰富，且时间跨度更大。但此数据集每个房间内的传感器密度、种类较小，且同样没有考虑智能家居的触发-响应特性。

此外，华盛顿州立大学自适应系统高级研究中心（Center for Advanced Studies in Adaptive Systems，CASAS）公开了其智能家居环境内采集的传感器数据集[?]，通过使用数据分析及人工智能技术来对家居环境进行解释、建模和预测，主要用于用户活动感知、自动健康诊断、节能家居自动化等领域，目标为改善用户的生活。此数据集主要面向用户活动引起的智能家居设备状态改变活动，采集的数据为总日志条目的形式，每一条目以三元组(t, d, s)组成，其中中t为时间戳，d为设备名称，s为设备状态。此数据集相比上述牛津大学数据集，用户活动以及场景更加丰富，但传感器精度、采样率、种类略有不足，故其更适合用于更上层的建模分析工作，如用户活动感知等。同样地，此数据集亦没有引入智能家居平台作为中枢，没有体现触发-响应特性。

除此之外，一些民间技术爱好者将数据集分享至知名数据分析竞赛平台Kaggle[?]上，这些数据集主要关注智能家居场景中某单一方面的数据，如各设备用电功率数据、智能家居控制指令数据等。总体来说，这些数据集涵盖事件类型、物理通道种类较少，并且缺少数据采集时的实验细节，不适合用于学术研究。

3.1.3　实验平台需求分析

上述介绍的现有智能家居公开数据集，绝大多数局限于传感器读数的记录，即在一个设定好的智能家居场景内安装传感器，并按不同的时间间隔连续采集数据。在事件指纹的研究方面，Peeves[?]进行的是类似的工作，故本文在事件指纹方面的研究可用此数据集进行；而由于本文还需要对事件依赖安全进行研究，在显式事件依赖方面，需要考虑智能家居场景中设定的触发-响应形式的自动化规则吗，而现有数据集均未考虑这一设定；而在隐式事件依赖方面，需要细粒度、定量地建模不同事件对物理通道的影响，故需要触发单一事件并连续收集物理通道的影响，而现有数据集中多个事件可能同时或以很小的时间间隔触发，造成其物理通道影响杂糅交错，会造成后续数据分析工作的误差。因此，现有公开数据集均不适合作为本文事件依赖相关研究的支

撑。

此外，除上述公开数据集的研究之外，现有智能家居安全相关研究（尤其是事件指纹、事件依赖等相关研究），绝大多数都未采用现成公开的数据集，而是自行搭建智能家居环境并设计应用场景来进行数据采集，导致这些数据集的泛用性较低，使用场景局限。同时，这些研究绝大多数均未公开自己的数据，这使得后续研究仍需重新设计并采集新的数据集，导致工作量加大，同时也无法进行有效、公平的横向对比。

综上所述，为了使智能家居场景灵活化，数据收集的过程方便化，本文设计并实现了小型拟真智能家居实物实验平台。本实验平台的目标为支撑本文后续研究中需要的数据收集工作，并作为后续多样的相关研究工作提供灵活的数据来源。具体来说，本实验平台的设计需求如下：

（1）拟真性：首先，不同于现有数据集并未引入真实的智能家居平台，本实验平台需要尽量地贴合真实的智能家居使用场景。具体来说分为三个方面：在控制中枢方面，本实验平台需要引入一个真实的智能家居中枢系统，并设计合理的自动化规则来模拟用户的使用场景；同时，在设备方面，本实验平台需要设置贴合真实场景的传感器与智能设备组合；此外，本实验平台还需模拟真实智能家居环境的其他特性，如家居物理环境、数据传输方式、协议类型等。

（2）灵活性：其次，本实验平台需要具有灵活多变的场景模拟能力。具体来说，本实验平台需要能够灵活改变各类用户属性及配置，如自动化规则等，用于满足用户的各种使用场景，以满足后续不同研究工作的数据需求。同时，本实验平台需要具有一定的设备扩展能力，即在后续出现新的需求时，可在不影响现有系统的前提下添加新的设备，来扩展平台的使用场景。

（3）易用性：最后，本实验平台需要具有丰富的上层功能设计，以使数据采集的过程更加方便、直观。具体来说，本实验平台需要为数据采集过程设计并实现丰富的前台功能，研究者只需输入与实验设置有关的一些基本信息（如设定哪些规则、触发哪些事件、收集哪些时间段的数据等），实验平台即可自动化完成数据采集与打包工作，无需用户向智能家居平台或数据库等后端应用进行对接。同时，本实验平台需要设计直观、易操作的前端用户UI界面，实时展示状态的同时也能方便用户控制。

3.2　实验平台设计

基于上述实验平台需求，本节从架构、功能两方面分别介绍本实验平台的设计。

3.2.1　硬件架构设计

架构图

智能家居平台：树莓派+Home assistant，mysql数据库

设备：DIY设备/传感器模组，ESP32模块，I2C或GPIO连接

3.2.2　软件架构设计

web app

1. 连接配置模块 2. Home Assistant RESTful API对接模块 3. 数据库自动化读取模块

3.2.3　实验平台功能设计

1. 自定义服务发送时间和状态 2. 自定义数据收集时间段 3. 智能家居状态实时展示

3.3　实验平台实现

3.3.1　感知层设备选择与部署

3.3.2　智能家居平台部署

3.3.3　自动化数据收集功能实现

3.4　平台使用方法与案例

第四章　智能家居轻量级事件认证

[– This page is a preset empty page –]

第五章　示例

5.1　参考文献引用

5.1.1　数字标注

`\cite{knuth86a}` ⇒ [**?**]

`\citet{knuth86a}` ⇒ **?**]

`\citet[chap.~2]{knuth86a}` ⇒ **?** , chap. 2]

`\citep{knuth86a}` ⇒ [**?**]

`\citep[chap.~2]{knuth86a}` ⇒ [**?** , chap. 2]

`\citep[see][]{knuth86a}` ⇒ [see **?**]

`\citep[see][chap.~2]{knuth86a}` ⇒ [see **?** , chap. 2]

`\citet*{knuth86a}` ⇒ **?**]

`\citep*{knuth86a}` ⇒ [**?**]

`\citet{knuth86a,tlc2}` ⇒ **? ?**]

`\citep{knuth86a,tlc2}` ⇒ [**? ?**]

`\cite{knuth86a,knuth84}` ⇒ [**? ?**]

`\upcite{knuth86a,knuth84}` ⇒ [? ?]

`\citet{knuth86a,knuth84}` ⇒ **? ?**]

`\citep{knuth86a,knuth84}` ⇒ [**? ?**]

`\cite{knuth86a,knuth84,tlc2}` ⇒ [**? ? ?**]

5.1.2　数字标注-上标形式

`\upcite{knuth86a}` ⇒ [?]

`\upcite{knuth86a,knuth84,tlc2}` ⇒ [? ? ?]

实现源码：`\newcommand{\upcite}[1]{\textsuperscript{\cite{#1}}}`。

5.1.3　著者-出版年制标

`\cite{knuth86a}` ⇒ **?**

`\citet{knuth86a}` ⇒ **?**

`\citet[chap.~2]{knuth86a}` ⇒ **?**, chap. 2

`\citep{knuth86a}` ⇒ (**?**)

`\citep[chap.~2]{knuth86a}` ⇒ (**?**, chap. 2)

`\citep[see][]{knuth86a}` ⇒ (see **?**)

`\citep[see][chap.~2]{knuth86a}` ⇒ (see **?**, chap. 2)

`\citet*{knuth86a}` ⇒ **?**

`\citep*{knuth86a}` ⇒ (**?**)

`\citet{knuth86a,tlc2}` ⇒ **??**

`\citep{knuth86a,tlc2}` ⇒ (**??**)

`\cite{knuth86a,knuth84}` ⇒ **??**

`\citet{knuth86a,knuth84}` ⇒ **??**

`\citep{knuth86a,knuth84}` ⇒ (**??**)

5.1.4　其他形式的标注

`\citealt{tlc2}` ⇒ **?**

`\citealt*{tlc2}` ⇒ **?**

`\citealp{tlc2}` ⇒ **?**

`\citealp*{tlc2}` ⇒ **?**

`\citealp{tlc2,knuth86a}` ⇒ **? ?**

`\citealp[pg.~32]{tlc2}` ⇒ **?** , pg. 32

`\citenum{tlc2}` ⇒ **?**

`\citetext{priv.\ comm.}` ⇒ [priv. comm.]

`\citeauthor{tlc2}` ⇒ **?**

`\citeauthor*{tlc2}` ⇒ **?**

`\citeyear{tlc2}` ⇒ **?**

`\citeyearpar{tlc2}` ⇒ [**?**]

5.2 浮动体

5.3 算法环境

模板中使用 `algorithm2e` 宏包实现算法环境。关于该宏包的具体用法请阅读宏包的官方文档。

————↓————Space Check————↓————

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
  read current;
  if *understand* **then**
    go to next section;
    current section becomes this one;
  else
    go back to the beginning of current section;
  end
end

算法 1: A How to (plain).

————↑————Space Check————↑————

算法 2: A How to (ruled).

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
  read current;
  if *understand* **then**
    go to next section;
    current section becomes this one;
  else
    go back to the beginning of current section;
  end
end

5.3.1 三线表

推荐使用三线表的方式，如表 4。

————┼————Space Check————┼————
————┼————Space Check————┼————

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
    read current;
    if *understand* **then**
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end

算法 3: A How to (boxed).

算法 4: A How to (boxruled).

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
    read current;
    if *understand* **then**
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end

表 3 表的标题

操作系统	TeX 发行版
所有	TeX Live
macOS	MacTeX
Windows	MikTeX

表 4 让我们看看一个长标题长什么样。还不够长？那我再多写一点。还是不够长？那我再多写一点点。**OK**，就是长这样的**!**

操作系统	TeX 发行版
所有	TeX Live
macOS	MacTeX
Windows	MikTeX

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

5.4 长表格

超过一页的表格要使用专门的 `longtable` 环境（表 5）。

————↓————Space Check————↓————

表 5 长表格演示

名称	说明	备注
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC

续下页

表 5 长表格演示（续）

名称	说明	备注
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC
AAAAAAAAAAAA	BBBBBBBBBBB	CCCCCCCCCCCCCC

————↑————Space Check————↑————

5.5　插图

————↓————Space Check————↓————

图 1 测试图片
第二行题注

————↑————Space Check————↑————

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

5.6 数学环境

5.6.1 数学符号

模板定义了一些正体（upright）的数学符号：

符号	命令
常数e	`\eu`
复数单位i	`\iu`
微分符号d	`\diff`
$\arg\max$	`\argmax`
$\arg\min$	`\argmin`

更多的例子：

$$\mathrm{e}^{\mathrm{i}\pi} + 1 = 0 \tag{5.1}$$

$$\frac{\mathrm{d}^2 u}{\mathrm{d}t^2} = \int f(x)\,\mathrm{d}x \tag{5.2}$$

$$\arg\min_{x} f(x) \tag{5.3}$$

5.6.2 定理、引理和证明

定义 5.1. If the integral of function f is measurable and non-negative, we define its (extended) **Lebesgue integral** by

$$\int f = \sup_{g} \int g, \tag{5.4}$$

where the supremum is taken over all measurable functions g such that $0 \leq g \leq f$, and where g is bounded and supported on a set of finite measure.

例 5.1. Simple examples of functions on $\mathbf{R}^d$ that are integrable (or non-integrable) are

given by

$$f_a(x) = \begin{cases} |x|^{-a} & \text{if } |x| \leq 1, \\ 0 & \text{if } x > 1. \end{cases} \tag{5.5}$$

$$F_a(x) = \frac{1}{1+|x|^a}, \qquad \text{all } x \in \mathbf{R}^d. \tag{5.6}$$

Then f_a is integrable exactly when $a < d$, while F_a is integrable exactly when $a > d$.

引理 5.1 (Fatou)**.** Suppose $\{f_n\}$ is a sequence of measurable functions with $f_n \geq 0$. If $\lim_{n\to\infty} f_n(x) = f(x)$ for a.e. x, then

$$\int f \leq \liminf_{n\to\infty} \int f_n. \tag{5.7}$$

注. We do not exclude the cases $\int f = \infty$, or $\liminf_{n\to\infty} f_n = \infty$.

推论 5.2. Suppose f is a non-negative measurable function, and $\{f_n\}$ a sequence of non-negative measurable functions with $f_n(x) \leq f(x)$ and $f_n(x) \to f(x)$ for almost every x. Then

$$\lim_{n\to\infty} \int f_n = \int f. \tag{5.8}$$

命题 5.3. Suppose f is integrable on $\mathbf{R}^d$. Then for every $\epsilon > 0$:

1. There exists a set of finite measure B (a ball, for example) such that

$$\int_{B^c} |f| < \epsilon. \tag{5.9}$$

2. There is a $\delta > 0$ such that

$$\int_E |f| < \epsilon \qquad \text{whenever } m(E) < \delta. \tag{5.10}$$

定理 5.4. Suppose $\{f_n\}$ is a sequence of measurable functions such that $f_n(x) \to f(x)$ a.e. x, as n tends to infinity. If $|f_n(x)| \leq g(x)$, where g is integrable, then

$$\int |f_n - f| \to 0 \qquad \text{as } n \to \infty, \tag{5.11}$$

and consequently

$$\int f_n \to \int f \qquad \text{as } n \to \infty. \tag{5.12}$$

证明. Trivial. □

5.6.3 自定义

Axiom of choice. Suppose E is a set and E_α is a collection of non-empty subsets of E. Then there is a function $\alpha \mapsto x_\alpha$ (a "choice function") such that

$$x_\alpha \in E_\alpha, \qquad \text{for all } \alpha. \tag{5.13}$$

Observation 5.1. Suppose a partially ordered set P has the property that every chain has an upper bound in P. Then the set P contains at least one maximal element.

A concise proof. Obvious. □

Observationvar2 5.2. Suppose a partially ordered set P has the property that every chain has an upper bound in P. Then the set P contains at least one maximal element.

A concise proof. Obvious. □

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

我们在这儿插入一行字；

我们在这儿再插入一行字；

结　论

学位论文的结论单独作为一章，但不加章号。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。

* 嗯，这就是你的论文了 *

[– This page is a preset empty page –]

附　录

下列内容可以作为附录：

1) 为了整篇论文材料的完整，但编入正文又有损于编排的条理和逻辑性，这一材料包括比正文更为详尽的信息、研究方法和技术更深入的叙述，建议可以阅读的参考文献题录，对了解正文内容有用的补充信息等；
2) 由于篇幅过大或取材于复制品而不便于编入正文的材料；
3) 不便于编入正文的罕见的珍贵或需要特别保密的技术细节和详细方案（这中情况可单列成册）；
4) 对一般读者并非必要阅读，但对专业同行有参考价值的资料；
5) 某些重要的原始数据、过长的数学推导、计算程序、框图、结构图、注释、统计表、计算机打印输出文件等。

* 嗯，自由发挥吧 *

[– This page is a preset empty page –]

攻读硕士学位期间取得的学术成果

对于博士学位论文，本条目名称用“攻读博士学位期间取得的研究成果”，一般包括：

攻读博士学位期间取得的学术成果：攻读博士学位期间取得的学术成果：列出攻读博士期间发表（含录用）的与学位论文相关的学位论文、发表专利、著作、获奖项目等，书写格式与参考文献格式相同；

攻读博士期间参与的主要科研项目：列出攻读博士学位期间参与的与学位论文相关的主要科研项目，包括项目名称，项目来源，研制时间，本人承担的主要工作。

对于硕士学位论文，本条目名称用“攻读硕士学位期间取得的学术成果”，只列出攻读硕士学位期间发表（含录用）的与学位论文相关的学位论文、发表专利、著作、获奖项目等，书写格式与参考文献格式相同。

* 嗯，研究生不列科研项目 *

[– This page is a preset empty page –]

致　谢

致谢中主要感谢指导教师和在学术方面对论文的完成有直接贡献及重要帮助的团体和人士，以及感谢给予转载和引用权的资料、图片、文献、研究思想和设想的所有者。致谢中还可以感谢提供研究经费及实验装置的基金会或企业等单位和人士。致谢辞应谦虚诚恳，实事求是，切记浮夸与庸俗之词。

* 嗯，感谢完所有人之后，也请记得感谢一下自己 *