清华大学学位论文 LaTeX 模板
使用示例文档 v7.1.0

(申请清华大学工学硕士学位论文)

培 养 单 位 ： 网络科学与网络空间研究院

学　　科 ： 网络空间安全

研 究 生 ： 高　　涛

指 导 教 师 ： 李 风 华　副研究员

二○二一年七月

An Introduction to LaTeX Thesis Template of Tsinghua University v7.1.0

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Cyberspace Security

by

Gao Tao

Thesis Supervisor: Associate Professor Li Fenghua

July, 2021

学位论文指导小组、公开评阅人和答辩委员会名单

指导小组名单

李 XX	教授	清华大学
王 XX	副教授	清华大学
张 XX	助理教授	清华大学

公开评阅人名单

刘 XX	教授	清华大学
陈 XX	副教授	XXXX 大学
杨 XX	研究员	中国 XXXX 科学院 XXXXXXX 研究所

答辩委员会名单

主席	赵 XX	教授	清华大学
委员	刘 XX	教授	清华大学
	杨 XX	研究员	中国 XXXX 科学院 XXXXXXX 研究所
	黄 XX	教授	XXXX 大学
	周 XX	副教授	XXXX 大学
秘书	吴 XX	助理研究员	清华大学

摘　要

现有问题：1. 大多数现有算法基于开源数据集，很难应用到真实网络环境中；2. 真实网络环境复杂，流量异常种类多，难以找到基线；本文主要工作如下：1. 综述了 2. 提出了一种 3. 实现了系统，

论文的摘要是对论文研究内容和成果的高度概括。摘要应对论文所研究的问题及其研究目的进行描述，对研究方法和过程进行简单介绍，对研究成果和所得结论进行概括。摘要应具有独立性和自明性，其内容应包含与论文全文同等量的主要信息。使读者即使不阅读全文，通过摘要就能了解论文的总体内容和主要成果。

论文摘要的书写应力求精确、简明。切忌写成对论文书写内容进行提要的形式，尤其要避免“第 1 章……；第 2 章……；……”这种或类似的陈述方式。

关键词是为了文献标引工作、用以表示全文主要内容信息的单词或术语。关键词不超过 5 个，每个关键词中间用分号分隔。

关键词：关键词 1；关键词 2；关键词 3；关键词 4；关键词 5

Abstract

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective, brief introduction to methodology and research process, and summarization of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “the first chapter”, “the second chapter” and the like should be avoided in the abstract.

Keywords are terms used in a dissertation for indexing, reflecting core information of the dissertation. An abstract may contain a maximum of 5 keywords, with semi-colons used in between to separate one another.

Keywords: keyword 1; keyword 2; keyword 3; keyword 4; keyword 5

目 录

摘 要.....I

Abstract.....II

目 录.....III

插图和附表清单.....VI

插图清单.....VIII

附表清单.....IX

符号和缩略语说明.....X

第 1 章 引言 1

1.1 研究背景 1

1.2 论文的研究内容 2

1.3 论文内容和组织结构 3

第 2 章 相关工作综述 4

2.1 引言 4

2.2 网络流量异常的定义和分类 4

2.3 网络流量异常检测算法 6

2.3.1 基于分类的异常检测算法 6

2.3.2 基于统计的异常检测算法 9

2.3.3 基于信息论的异常检测算法 11

2.3.4 基于聚类的异常检测算法 12

2.4 异常检测领域开源数据集介绍 14

2.5 异常检测算法对比 14

2.6 现有异常检测算法存在的问题 14

第 3 章 数据集分析与对比 16

3.1 开源数据集 16

3.1.1 UNSW-NB15 16

3.1.2 CICIDS2017 19

3.2 校园网真实数据集.....21
3.2.1 全流量日志数据.....21
3.2.2 威胁告警日志.....21
3.2.3 TCP 流量和 UDP 流量日志.....22
3.3 数据预处理及评价指标.....22
3.4 现有算法在不同数据集的评估结果.....24
第 4 章 基于特征关系图的 RNN 及其网络流量异常检测算法.....25
4.1 引言.....25
4.1.1 神经网络的类型.....25
4.1.2 RNN.....28
4.1.3 LSTM.....28
4.1.4 GRU.....30
4.2 基于关系图结构的 RNN.....31
4.3 算法性能评估.....32
4.3.1 实验参数设置.....32
4.3.2 基于开源数据集的检测结果.....33
4.3.3 基于真实数据的检测结果.....33
第 5 章 流量检测系统的设计与实现.....36
5.1 引言.....36
5.2 spark 平台介绍.....36
5.3 系统整体设计.....37
5.4 流式数据特征抽取.....40
5.5 模型训练模块的设计与实现.....42
5.6 预测输出模块.....42
5.7 系统结果展示与分析.....42
第 6 章 总结与展望.....44
6.1 总结.....44
6.2 未来研究和展望.....45
参考文献.....46
附录 A 补充内容.....48
致 谢.....50

声 明.....51
个人简历、在学期间完成的相关学术成果.....52
指导小组学术评语.....53
答辩委员会决议书.....54

插图和附表清单

图 1.1 网民规模和互联网普及率 1
图 1.2 用户流量变化图 2
图 1.3 用户应用类型图 2
图 2.1 异常检测的通用框架 4
图 2.2 Replicator Neural Networks 示意图 8
图 3.1 流量数据示意图 22
图 3.2 流量数据集制作流程 22
图 3.3 NOC 平台数据样例 23
图 4.1 人工神经元模型 27
图 4.2 循环神经网络 28
图 4.3 普通 RNN 结构 29
图 4.4 LSTM 结构.....29
图 4.5 GRU 结构 30
图 4.6 结果 1 33
图 4.7 结果 1 33
图 4.8 Example figure.....35
图 4.9 Example figure.....35
图 5.1 系统架构图 36
图 5.2 Logistic regression in Hadoop and Spark 37
图 5.3 Spark 组件.....37
图 5.4 spark 工作原理.....38
图 5.5 Spark 数据源.....38
图 5.6 spark streaming 执行流程 39
图 5.7 特征提取 41
图 5.8 特征文件 42
图 5.9 Example figure in appendix 42

表 3.1 UNSW-NB15 数据集攻击类别.....17
表 3.2 与协议相关的特征 18
表 3.3 与时间相关的特征 18

表 3.4　与报文大小相关的特征 ……19
表 3.5　与连接状态相关的特征 ……19
表 3.6　额外构造的特征 ……19
表 3.7　packet 大小相关的特征……20
表 3.8　时间相关的特征 ……20
表 3.9　标志位相关的特征 ……21
表 4.1　激活函数 ……27
表 4.2　部分评估结果，待填充具体数值 ……34

插图清单

附表清单

符号和缩略语说明

PI	聚酰亚胺
MPI	聚酰亚胺模型化合物，N-苯基邻苯酰亚胺
PBI	聚苯并咪唑
MPBI	聚苯并咪唑模型化合物，N-苯基苯并咪唑
PY	聚吡咙
PMDA-BDA	均苯四酸二酐与联苯四胺合成的聚吡咙薄膜

第 1 章　引言

1.1　研究背景

网络自诞生以来，随着数十年的发展，已经成为了最重要的信息化基础设施之一。如图 4.7所示，第 46 次《中国互联网络发展状况统计报告》[1] 指出，截至 2020 年 6 月，我国的移动网民用户规模已经累计达到 9.40 亿，互联网的网络普及率已经达到高达 67.0%，互联网的主要应用领域覆盖范围已经达到包括即时移动通信、搜索结果引擎、网络即时新闻、线下线上教育、购物以及出行等各个方面，可以这么说这就是如今互联网已经和每一个人的工作日常生活息息相关密不可分。

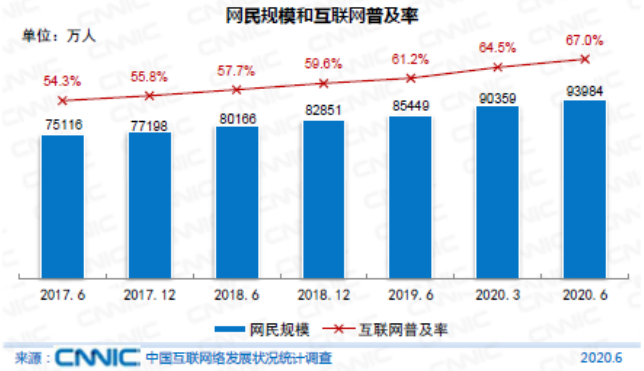

图 1.1　网民规模和互联网普及率

随着网络重要性的提升、用户规模的膨胀，管理网络的难度也越来越大。网络系统就是一个繁琐而又复杂的网络，它的设计、运营以及维护都必须依靠专业的网络运维技术人员。早期的网络运维管理工作绝大多数都是由运维工作人员手工操作来完成，此后人们逐渐发现一些重复性的工作可以用自动化脚本来实现，于是诞生了自动化运维。自动化的运维系统可以认为是一种建立在专家经验和人为制订规则的系统之上。但是随着移动互联网的规模迅速增长和巨大发展，以及其服务种类和方式的日益多样化，简单、基于人为而制定运维规则的技术和方法并不一定能够有效地解决企业进行大规模运维的困难，因此产生了智能运维。与自动化运维依赖专家知识、人工生成规则不同，智能运维更加需要强调的是利用先进机器学习算法从海量的运维管理数据中对其进行不断地综合学习和分析提炼规则。

异常检测是智能运维的关键环节，具有至关重要的意义。从网络故障管理的角度来说，做好异常检测可以提前预测故障的发生；从性能管理的角度来说，可以发现性能不佳的区域，避免因误配置、架构不合理导致性能下降；从安全管理的角度来说，在网络攻击的前期阶段，及时发现并预警后续攻击，进而做出防御措施。

因此，在复杂的网络环境中甄别出有效和异常流量尤为重要，在重大事故发生前，根据各项流量特征的变化，提前预测出即将发生的事故，提高应急响应速度，防患于未然。

本文以校园网为例，进行网络流量异常检测算法和系统的研究。清华大学校园网是全球规模最大，架构最复杂，流量场景最多变的校园网之一，具有以下几个特点：

1. 用户规模大，流量峰值高。每天有 10 万台设备活跃在线，同时在线设备最多为 7 万台，峰值流量约为 30Gbps/s，如图 1.2所示。
2. 用户应用类型多，如图 1.3所示，清华大学校园网中的用户类型远比一般的企业网复杂，在网络环境中几百种应用同时使用，这给数据分析带来了很多困难。
3. 异常流量是常态。扫描流量、攻击流量占比多。

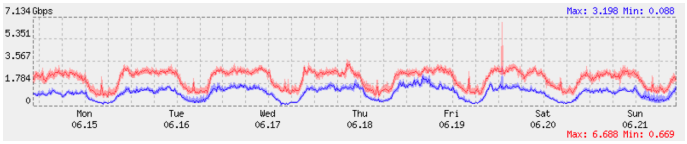

图 1.2　用户流量变化图

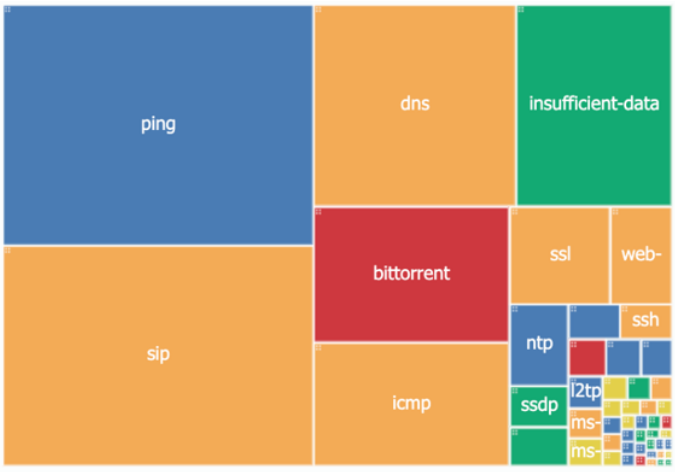

图 1.3　用户应用类型图

1.2　论文的研究内容

围绕 1.1 节中提到的清华大学校园网流量异常检测的特点和挑战，，本文分别展开了以下三方面的研究：

1. 本文分析对比了流量异常检测领域的两个经典开源数据集 UNSW-NB15 和 CICIDS2017，引入了清华大学校园网的真实数据集，并且分别在这几个数据

集上对现有的异常检测算法进行了实验对比。

2. 在现有算法的基础上，本文将特征关系引入到神经网络的训练中，提出了一种基于特征之间关系图的循环神经网络算法 (FG-RNN，Feature Graph-Recurrent Neural Network)。在复杂的网络流量环境下，流量特征种类繁多，且特征之间的相关性会随着流量变化而变化。原有的异常检测算法通常直接利用提取的特征进行训练，本文提出的 FG-RNN 算法有效利用了特征之间的相关性信息，将其加入到神经网络的训练过程中。
3. 本文对基于特征之间关系图的循环神经网络算法进行了流式改造，使之能够满足当前实时异常检测的需求。为了应对海量的校园网流量数据和高速数据流，本文设计了一个基于 spark streaming 的实时异常检测系统。该系统分为输入模块、模型模块、检测模块三部分，输入模块负责将流量数据进行窗口划分、特征选择、特征抽取、合并计算，得到的特征矩阵与预训练得到的关系矩阵一同送到模型中进行训练，模型中的参数每 2 小时更新一次，最后由检测模块对当前流量进行判别，并给出异常流量的类别。

1.3　论文内容和组织结构

第一章为引言，主要介绍本文的研究背景以及本文的研究内容。

第二章是关于网络流量异常检测的相关工作综述。首先对网络流量异常的定义和分类进行了介绍；接下来着重介绍了基于不同原理的异常检测算法；然后在开源数据集上对部分经典算法进行了复现，对比了其优缺点；最后，指出了现有的异常检测算法在当前大规模流量环境下存在的主要问题。

第三章至第五章是本文的主要研究内容。第三章分析对比了开源数据集与校园网真实数据集，并且在多个数据集上对现有的算法进行了实验对比。第四章提出了基于特征关系图的循环神经网络算法，并进行了算法性能评估。第五章设计并实现了一个基于 spark streaming 的流式处理系统。

最后，第六章对全文进行了总结，并且提出了未来展望。

第 2 章　相关工作综述

2.1　引言

本章对网络流量异常检测领域的相关工作进行综述。

异常检测是一个重要的领域，自 1980 年以来，国内外已经有无数学者在这方面做研究。Ahmed et al.[2] 将异常检测技术分为分类、统计、信息理论和聚类四类。待补充。

异常检测的通用框架如图 2.1所示。

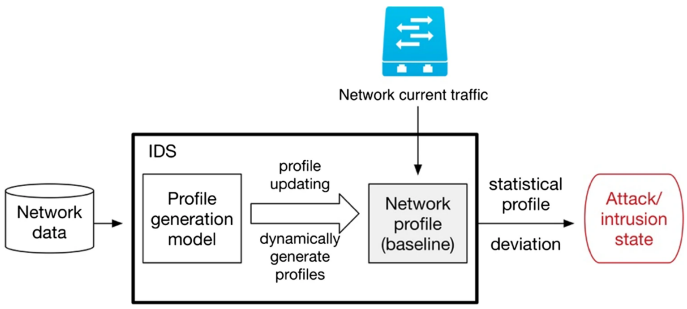

图 2.1　异常检测的通用框架

2.2　网络流量异常的定义和分类

Hawkins(1980) 给出了异常的本质性定义[3]：异常是在数据集中表现出差异的那些数据, 使人怀疑这些数据并非随机偏差，而是产生于完全不同的机制。例如在道路交通领域，某条道路的车流量突然增多，甚至多至堵塞，又或者突然减少，此时车流量数据就是一个异常。因此网络行为的异常就是指那些与正常的、标准的, 我们所预期的行为相异的表现。为了检测网络异常，网络所有者必须有一个预期或正常行为的概念，我们称其为基线。要检测网络行为的异常，就需要持续监控网络中的意外趋势或事件，那些可能改变网络流量特征或者监控指标的恶意行为。

限于篇幅，本文只关注引起网络流量特征变化的恶意行为，而对于系统权限提升，缓冲区溢出等黑客攻击手段暂时不做研究。

网络流量异常具体有哪些类别，学术界没有统一的意见。本文中关注的网络流量异常按照产生意图分为恶意和非恶意两类，其中恶意行为主要有拒绝服务攻击，网络扫描，BGP 劫持，网络蠕虫，僵尸网络等；非恶意行为主要有物理故障，突发事件等。接下来我们对这些异常分别进行介绍：

1. 拒绝服务攻击。拒绝服务 (DoS) 攻击是一种网络攻击，恶意行为者的目的

是通过中断计算机或其他设备的正常运作，使其目标用户无法使用该设备。DoS 攻击的功能通常是通过构造大量请求淹没目标机器，直到正常的流量无法处理，导致额外用户的拒绝服务。分布式拒绝服务（DDoS）攻击是一种来自许多分布式来源的 DoS 攻击，如僵尸网络 DDoS 攻击。因其攻击成本低、攻击效果明显等特点，DDoS 攻击仍然是互联网用户面临的最常见、影响较大的网络安全威胁之一。DoS 攻击通常分为 2 类。（1）缓冲区溢出攻击：一种攻击类型，内存缓冲区溢出会导致机器消耗所有可用的硬盘空间、内存或 CPU 时间。这种形式的利用通常会导致行为迟缓、系统崩溃或其他有害的服务器行为，导致拒绝服务。（2）洪泛攻击：通过用大量的数据包使目标服务器饱和，恶意行为者能够使服务器容量过饱和，导致拒绝服务。为了使大多数 DoS 泛滥攻击成功，恶意行为者必须拥有比目标更多的可用带宽。

2. 网络扫描。网络扫描黑客在进行网络攻击之前，首先要进行网络扫描，从中寻找可以攻击的目标。具体来说，黑客需要确定网络中哪些主机是活动的，活动主机上运行了哪些存在漏洞的服务等，从而决定下一步的攻击计划。网络扫描分为主机扫描和端口扫描两种。

   主机扫描主机扫描的目的是确定网络中存在哪些在线的主机或设备。端口扫描端口扫描的目的是确定活动主机上开放了哪些端口，运行了哪些网络服务。按照扫描方式来分，端口扫描分为垂直扫描和水平扫描。垂直扫描会扫描某个主机的所有端口，而水平扫描则扫描网络中所有主机的某个固定端口。

3. BGP 劫持。BGP 劫持是指攻击者恶意地对互联网流量进行重新路由。攻击者通过谎称拥有一组 IP 地址（称为 IP 前缀）的所有权来实现这一目的，而实际上他们并不拥有、控制或路由。BGP 劫持就像有人把高速公路上的所有标志都换掉，把汽车流量改道到错误的出口。

4. 网络蠕虫。网络蠕虫不同于计算机病毒。与计算机病毒不同的是，计算机蠕虫不需要附在别的程序内，可能不用用户介入操作也能自我复制或运行。计算机蠕虫未必会直接破坏被感染的系统，却几乎都对网络有害。计算机蠕虫可能会执行垃圾代码以发动分布式拒绝服务攻击，令计算机的执行效率极大程度降低，从而影响计算机的正常使用；可能会损毁或修改目标计算机的文件；亦可能只是浪费带宽。（恶意的）计算机蠕虫可根据其目的分成 2 类：

   一种是面对大规模计算机使用网络发动拒绝服务的计算机蠕虫，虽说会绑架计算机，但用户可能还可以正常使用，只是会被占用一部分运算、联网能力。另一种是针对个人用户的以执行大量垃圾代码的计算机蠕虫。计算机蠕虫多不具有跨平台性，但是在其他平台下，可能会出现其平台特有的非跨平台性

的平台版本。第一个被广泛注意的计算机蠕虫名为：“莫里斯蠕虫”，由罗伯特 · 泰潘 · 莫里斯编写，于 1988 年 11 月 2 日释出第一个版本。这个计算机蠕虫间接和直接地造成了近 1 亿美元的损失。这个计算机蠕虫释出之后，引起了各界对计算机蠕虫的广泛关注。

5. 僵尸网络。僵尸网络 (简称 “机器人网络”) 是指由感染了恶意软件的计算机组成的网络，这些计算机由单一攻击方控制，即所谓的” 僵尸继承者”。在” 机器人携带者” 控制下的每一台机器都被称为” 机器人”。攻击方可以从一个中心点，指挥其僵尸网络上的每一台计算机同时进行协调的犯罪行动。僵尸网络的规模 (许多僵尸网络由数百万个僵尸组成) 使攻击者能够进行大规模的行动，这在以前的恶意软件中是不可能的。由于僵尸网络一直处于远程攻击者的控制之下，受感染的机器可以接收更新，并在飞行中改变其行为。因此，僵尸牧民往往能够在黑市上租用其僵尸网络部分的访问权，以获取大量经济利益。
6. 物理故障是指路由器故障，链路破坏，线缆损坏，断电等不可预测的突发事件。
7. 突发事件。突发事件是指引起网络瞬时拥塞的事件，有可能是管理员配置错误，也有可能是正常的网络操作，如某网站访问量激增。

2.3　网络流量异常检测算法

本节将介绍在异常检测领域主流的一些算法，根据所依赖的技术原理的不同，将这些算法分为了基于分类、基于统计、基于聚类、基于信息论的异常检测算法。

2.3.1　基于分类的异常检测算法

基于分类的方法依赖于建立知识库的正常流量活动特征，并将偏离基线特征的活动视为异常活动。其优势在于它们能够检测到完全新颖的攻击，假设它们表现出大量的偏离正常基线的情况。此外，由于知识库中未包含的正常流量被认为是攻击，因此会有无意中的误报。因此，异常检测技术需要进行训练，以建立正常的活动基线，这个建立基线的过程通常非常耗时，而且还取决于是否有完全正常的流量数据集。在实践中，获得无攻击的流量实例是非常罕见且昂贵的。此外，在当今 s 动态和不断变化的网络环境中，保持正常基线的更新是非常困难的。在现有的大量基于分类的网络异常检测技术中，我们主要讨论以下四种技术。

2.3.1.1　SVM

Eskin et al.[4] 引入无监督 SVM 的概念来检测异常事件。常规的 SVM 的原理是推导出一个超平面，使得正类样本和负类样本之间的分离余量最大化，将特征空间中的两类数据进行分离。标准的 SVM 算法是一种监督学习方法，需要标记数据来创建分类规则。而该算法经过改进 SVM，试图将整个训练数据集从原点分离出来，找到一个以最大余量将数据实例与原点分离的超平面，

Hu et al.[5] 提出了一种忽略噪声数据的异常检测方法，该方法使用 Robust SVM(RSVM) 来开发。标准的 SVM 有一个主要假设，即所有的训练样本数据都是独立且相同分布的 (i.i.d)。但是在实际场景中，训练数据往往包含噪声，这就会导致标准 SVM 会学习出一个高度非线性的决策边界，从而导致通用性较差。基于此，RSVM 以类中心的形式加入了平均化技术，使得决策面更加平滑。此外，RSVM 另一个优点是能大大降低支持向量的数量，从而减少运行时间，提高效率。

Taeshik Shon 等人[6] 提出了一种基于增强支持向量机 (Enhanced Support Vector Machines) 的异常检测算法。增强支持向量机是该文作者提出的一种新型的支持向量机，该向量机同时具备了传统支持向量机的高性能和单类支持向量机检测新异常的能力。在预处理阶段，检测算法使用数据包过滤器滤除畸形数据包。随后，检测算法使用遗传算法对数据包头的信息进行特征选择。接下来，检测算法利用 SOM 网络对正常流量进行聚类，并用这些聚类来训练增强支持向量机，从而得到最终的异常检测器。

2.3.1.2　贝叶斯

贝叶斯网络是对包含不确定性的领域进行建模的一种有效方法。一个离散的随机变量用一个有向无环图（DAG）表示，其中每个节点反映了随机变量的状态，并包含一个条件概率表（CPT）。CPT 的任务是提供一个节点处于特定状态的概率。在贝叶斯网络中，节点之间存在着父子关系，这表明子节点所代表的变量依赖于父节点所代表的变量。由于这种网络可以用于事件分类方案，因此也适用于网络异常检测。

Kruegel 等人[7] 假设异常检测系统包含许多模型，用于分析一个事件的不同特征。他们指出了在这种系统下异常检测技术造成高误报率的两个主要原因：一是异常检测系统通过将多个概率模型的输出进行汇总，而每个模型往往只给出一个事件的常态/异常的得分或概率，从而导致高误报率；二是异常检测系统无法处理那些不正常但合法的行为，如 CPU 利用率、内存使用率突然增高等。基于贝叶斯网络的概念，[7] 提出了一种解决上述问题的方法。对于一个输入事件的有序流

($\boldsymbol{S} = e_1, e_2, e_3...$)，异常检测系统决策每个事件是正常还是异常。该决策基于 k 个模型 ($\boldsymbol{M} = m_1, m_2, ..., m_k$) 的输出 ($o_i | i = 1, 2, ..., k$) 和可能的附加信息 ($\boldsymbol{I}$)。应用贝叶斯网络来识别异常事件，引入根节点，根节点代表一个具有两种状态的变量。一个子节点用于捕捉模型的输出，子节点与根节点相连，预计当输入异常或正常时，输出事件会有所不同。

2.3.1.3　神经网络

神经网络已经被应用于各个应用领域，如图像和语音处理，但其对计算量的要求很高。神经网络对数据进行分类的优势也可被用于网络异常检测。在网络异常检测领域，神经网络通常会和其他技术进行结合，如统计方法。

Hawkins 等人[8] 提出了一个多层的前馈神经网络，该神经网络可以用来进行异常值的检测。具体来说，Replicator Neural Networks 是一个多层前馈的神经网络 (multi-layer feed-forward neural networks)，在输入层和输出层之间放置了三个隐藏层。它的目标是通过训练在输出层以最小的误差重现输入数据模式。由于该模型中间隐藏层节点的个数少于输入输出层节点的个数，这样就起到了压缩数据和恢复数据的作用。Replicator Neural Networks 的示意图如图 2.2所示。

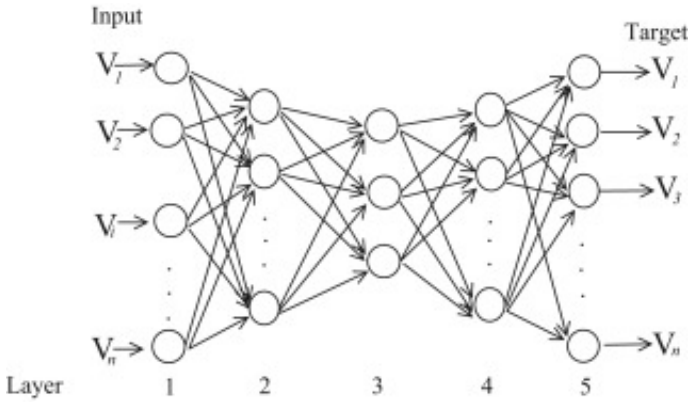

图 2.2　Replicator Neural Networks 示意图

Zhang[9] 提出了一种神经网络与统计模型相结合的分层入侵检测系统。将神经网络分类器的输出表示为一个连续变量 (t)，其中-1 表示有绝对把握的入侵，1 表示没有攻击。此外，自组织地图 (SOM) 被用于网络异常检测。Ramadas 等 (2003) 提出，利用 SOM，可以对网络流量进行实时分类。SOM 依赖于这样一个假设，即网络攻击可以由不同的神经元组来描述，这些神经元组与其他神经元组相比，在输出神经元图上覆盖更大的区域。Poojitha 等人[10] 开发了一个由反向传播算法训练的前馈神经网络，利用给定的数据集与计算机网络在正常和异常行为期间的相关信息来检测异常。

2.3.2　基于统计的异常检测算法

早期的异常检测方法往往基于统计与概率模型，也就是假设-检验的方法。首先对数据的分布做出假设，然后找出假设下所定义的“异常”，因此往往使用极值分析或假设检验。比如对最简单的一维数据假设服从正态分布，然后将距离均值某个范围以外的点当做异常点。推广到高维后，假设各个维度相互独立。这类方法的好处速度一般比较快，但是因为存在很强的“假设”，效果不一定很好。因此统计技术的主要挑战是找到减少硬阈值引起的误报产生的方法[11]。例如，可以使用统计信号处理程序来提高检测率，同时减少误报，如 Lakhina 等人在主成分分析工作中所做的工作[12-14]。

2.3.2.1　小波分析

小波分析的重点是对非稳定数据序列进行建模, 这种数据序列可能包含在长时间内振幅和频率都会变化的信号。小波变换是将傅里叶变换的基由无限长的三角函数基换成有限长的会衰减的小波基。小波是强大的基函数，在时间和频率上是局部的，允许被表示的序列和它们的系数之间有密切的联系。这样不仅能够获取频率，还可以定位到时间。

Callegari 等人[15] 提出了一种利用小波与基线相结合的实时异常检测方法。它是通过提取 NetFlow 轨迹，并将其转化为 ASCII 数据文件，进行路由器级别的分析。经过格式化后，通过哈希函数将不同的流量汇总到基线中。然后，将时间序列数据进行小波变换，若发现不连续点则视为异常点。

另一项使用小波的研究是由 Hamdi 等人[16] 产生的。它依赖于通过区分危险和非威胁性的异常来识别与攻击相关的异常。这项任务是在周期观测概念的基础上完成的，小波理论被用来分解一维信号，以分析其特殊频率和时间定位。

2.3.2.2　主成分分析

主成分分析 (Principal component analysis，PCA) 是一种广泛应用于计算机网络异常检测的统计技术。该方法的主要思想是将 n 个相关变量组成的数据集映射到一个新的、缩小的 k 个变量集上，即主成分 (PC)，其中 $k << n$。基于矩阵分解的异常点检测方法的关键思想是利用主成分分析去寻找那些违背了数据之间相关性的异常点。为了发现这些异常点，基于主成分分析（PCA）的算法会把原始数据从原始的空间投影到主成分空间，然后再把投影拉回到原始的空间。如果只使用第一主成分来进行投影和重构，对于大多数的数据而言，重构之后的误差是小的；但是对于异常点而言，重构之后的误差依然相对大。这是因为第一主成分反映了正

常值的方差，最后一个主成分反映了异常点的方差。

Lakhina 等人[12] 利用 PCA 将流量测量结果有效地分离为正常和异常子空间，解决了网络流量的异常诊断问题。该方法是基于将一组网络流量测量所占的高维空间分离成不相干的子空间，分别对应正常和异常的网络条件。PCA 的结果是 k 个子空间，对于与正常的网络流量行为，而剩余的 m 个子空间 ($m = n - k$) 则由异常和噪声组成。然后，将每一个新的流量测量值映射到这两个子空间上，这样就可以通过设置不同的阈值将这些测量值划分为正常或异常。

Pascoal 等人[17] 提出的异常检测方法采用了鲁棒 PCA 检测器与鲁棒特征选择算法合并，以获得对不同网络背景和环境的适应性。该鲁棒 PCA 与经典的 PCA 方法相反，它对异常值不敏感，并且无需使用可靠标记的数据集进行训练。

Fernandes 等人[18] 提出了一种基于统计程序主成分分析的异常检测算法。该算法首先生成一个 “使用流量分析的网络段数字签名 (DSNSF)” 的网络基线，然后将该基线与真实网络流量进行比较以识别异常事件。该系统分析了几天内的历史网络流量数据，从中找出最重要的流量时间间隔，同时对数据集进行了缩减，使新的缩减集能够有效地描述正常的网络行为。然后，以 DSNSF 作为阈值，通过得到的 PCA 参数，限制一个区间，偏差阈值的被认为是异常的。该系统共使用七个流量特征，三个 IP 流量特征 (bits/s, packets/s, flows/s) 被用来生成 DSNSF，四个流量属性（源 IP 地址、目的 IP 地址、源 TCP/UDP 端口和目的 TCP/UDP 端口）被用来生成一个包含有关异常流量间隔的报告。这种方法的缺点是只使用流量属性进行异常检测，只考虑检测基于流量的攻击。

2.3.2.3　协方差矩阵

协方差矩阵是二阶统计，已经被证明是一种强大的异常检测方法。该领域的一个有趣的方向是寻找哪些变量最能标记网络异常，提高检测性能。

Yeung 等人[19] 采用协方差矩阵分析来检测洪泛攻击。该方法将网络流量建模为协方差矩阵样本，以利用时序样本中包含的统计量达到检测泛洪攻击的目的，直接利用协方差矩阵的变化和相关特征的差异来揭示正常流量与各种类型的洪泛攻击之间的变化。

Miao Xie 等人[20] 研究了一种基于段的方式处理数据的技术。因为在无线传感器网络 (WSNs) 中，人们观察到大多数异常事件都会持续相当长的时间。由于现有的异常检测技术通常是以基于点的方式单独处理每个观测值，它们无法可靠和有效地报告在单个传感器节点中出现的这种长期异常。因此该方法采用斯皮尔曼等级相关系数（Spearman’s rank correlation coefficient 或 Spearman’s ρ）和差分压缩概念近似的样本协方差矩阵，以大幅降低计算和通信成本。

2.3.3　基于信息论的异常检测算法

信息论是一门以信息量化和冗余分析为核心的数学学科，其前身是 1948 年 Claude E. Shannon 在寻求信号处理和通信操作的数据压缩、传输和存储时提出的设想[21]。然而，它的应用扩展到许多其他领域，如电信、决策支持系统、模式识别等。常用的信息理论测量方法有香农熵、广义熵、条件熵、相对熵、信息增益和信息成本等。信息论应用于异常检测的途径主要是依靠计算流量特征的相互信息或熵值来识别异常分布。

2.3.3.1　熵

熵（entropy）是接收的每条消息中包含的信息的平均量，可以理解为不确定性的度量，因为越随机的信源的熵越大。异常检测领域中，熵可以有效地将流量特征描述为分布，例如如源/目的端口或 IP 地址，因为有某些类型的异常会对严重影响这些分布。通过这种方式，可以检测到例如由目的端口熵的变化表示的端口扫描攻击。

Behal 等人[22] 指出，由于 DDoS 攻击和突发事件会引起网络流量模式的大幅改变，而基于信息理论的熵或散度可以快速捕捉网络流量行为中的这种差异。因此，他们提出了一种利用流量之间的熵差进行异常检测的算法。通过采用了一组泛化的 ϕ-熵和 ϕ-散度，检测合法流量和攻击流量之间的信息距离。经过实验，该算法对于突发事件和 DDoS 的检测精度较高，而在其他数据集上表现一般。

David 等人[23] 提出了一种通过快速熵和基于流量的分析来增强对 DDoS 攻击的检测的方法。作者将观察到的流量汇总成一个单一的流量，并考虑到每个连接在一定时间间隔内的流量数，而不是取每个连接的数据包数。第二步基本上是计算每个连接的流量计数的快速熵。最后，根据快速熵和流量计数的均值和标准差生成一个自适应阈值。阈值随流量模式状况不断更新，提高了检测精度，同时快速熵的使用减少了计算处理时间。

2.3.3.2　KL 散度

KL 散度，又称相对熵，通常用于测量一个随机变量 X 的真实概率分布 P 与任意概率分布 Q(P 的近似) 之间的差异。设 $p(x)$、$q(x)$ 是离散随机变量 X 中取值的两个概率分布，则 p 对 q 的相对熵是：

$$D_{KL}(p||q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} = E_{p(x)} \log \frac{p(x)}{q(x)} \tag{2-1}$$

在机器学习中，P 往往用来表示样本的真实分布，Q 用来表示模型所预测的分布。

Xie 等人[24] 利用 KL 散度着重检测了无线传感器网络 (WSN) 中的一种特殊类

型的异常，这种异常会同时出现在邻近节点的集合中，并持续相当长的时间。基于节点的技术在此场景下效果和效率都不尽如人意。作者提出了基于分布式段的递归核密度估计，可以跟踪全局的概率密度函数，并连续测算其每两个时间段的差异，以便进行决策。为了以较低的通信成本实现分布式估计，作者采用 KL 散度作为度量方法。利用真实世界的数据集对算法进行评估，结果表明，该算法可以以更低的通信成本实现很好的性能。

Li 等人[25] 以检测无线传感器网络中的异常数据值为目标，提出了一种基于差分 KL 散度的异常检测方案。该方案首先将整个传感器网络划分为若干个簇，每个簇中的传感器在物理上相互接近，并且具有相似的感知值。然后，在每个簇内使用 KL 散度，以通过统计测量两个数据集之间的差异来检测异常值。他们的工作取得了良好的检测率和较低的误报率，同时比其他文献中的类似研究消耗更少的 CPU、内存等资源。

2.3.4　基于聚类的异常检测算法

聚类分析是把彼此相似的对象分成不同的组别，组内的对象是相似的（相关的），而不同组之间是不同的（不相关的）。如果组内的相似性越大，组间的差别越大，说明聚类的效果越好。因此，聚类技术可以用于离群值检测 (Outlier Detection)，识别出与正常组别相距较“远”的值，判定为异常值/离群值[26]。聚类算法通常是基于距离/密度发现异常点。其关键步骤在于给每个数据点都分配一个离散度，针对给定的数据集，对其中的任意一个数据点，如果在其局部邻域内的点都很密集，那么认为此数据点为正常数据点，而异常点则是距离正常数据点最近邻的点都比较远的数据点。通常由阈值进行距离远近的界定。

Rajasegar 等人[27] 提出了一种基于分布式超球面集群的聚类算法，用于检测无线传感器网络中的异常。该算法利用聚类对每个节点的流量数据进行建模，通过使用 k 个最近邻（KNN）集群的平均集群间距来识别异常集群，就可以将数据向量分类为正常或异常。该算法的特点是在分布式系统下进行，传感器节点上报集群聚类的信息，在与其他节点通信之前，由中间结点先行合并，从而使得通信开销最小化。

K-means 是一种经典的聚类技术，能够将数据分为不同的类别，但是它存在局部收敛性和对聚类中心点选择的敏感性等缺点。因此，许多研究者尝试将 k-means 与其他技术相结合，以克服这些缺点。Karami 等人[28] 设计了一种基于粒子群优化 (particle swarm optimization,PSO) 和 k-means 与局部优化混合的模糊异常检测系统，以确定最优的簇数。它分为两个阶段：训练阶段旨在通过将 PSO 的全局搜索的边界处理方法与 k-means 的快速收敛相结合，找到近似最优解，同时避免陷入局部最

优解的困境。在检测阶段，由于对于任何数据 (正常或攻击)，都有可能与某些集群处于近距离，因此通过引入模糊方法，可以有效降低误报率。

Carvalho 等人[29] 开发了一种主动式网络监控系统，可以检测异常事件，减少决策中的人工干预和错误概率。他们提出一种创建网络基线轮廓 DSNSF（使用流量分析的网段数字签名）的方法。该方法通过修改蚁群优化算法，使用聚类方法描述正常的网络使用情况，该方法称为 ACODS。ACODS 在大量高维输入数据中，通过无监督学习机制优化提取行为模式，对网络流量发现进行表征。然后为了检测异常行为他们首先计算每个时间区间内真实流量与正常曲线的相似度；然后计算序列之间的距离，并提供基于距离的测量方法。作者所提出的告警系统采用七种流量属性 (Bits, Bytes, Flows, Origin IP, Destination IP, Origin Port, Destination Port) 工作，利用熵来计算 IP 地址和端口特征的相关信息。当检测到异常时，ACODS 会提供一份包含 IP 流量信息的完整报告，说明每个属性对检测到的异常时间间隔的影响。ACODS 具有平方复杂度，导致解的收敛要经过多次迭代，作者试图通过使用局部搜索和信息素更新来缓解。

Dromard 等人[30] 提出了一种基于网格增量聚类算法和离散时间滑动窗口的无监督异常检测器。网格增量聚类比常规的聚类算法更有效率，因为后者只更新之前的特征空间分区，而不是每当增加或删除很少的点时，就对整个空间进行重新分区。增量网格聚类的使用有助于降低系统复杂度，从而使其在实时检测方面更加可行。最后系统合并这些更新的分区，用以识别最不相似的异常值。

Syarif 等人[31] 研究了各种聚类算法在应用于异常检测时的性能。他们使用了五种不同的方法，即 k-means、改进的 k-means、k-medoids、期望最大化（EM）聚类和基于距离的异常检测算法。下表展示了这些算法在 NSL-KDD 数据集下的性能表现。

Algorithm	Accuracy(%)	False positive(%)
k-means	57.81	22.95
Improved k-means	65.4	21.52
k-medoids	76.71	21.83
EM clustering	78.06	20.74
Distance-based anomaly detection	80.15	21.14

2.4　异常检测领域开源数据集介绍

数据集主要由 KDDCUP99, CICIDS 等。网络流量异常检测领域最为经典的数据集当属 KDD99，但是这个数据集年代过于久远，对于现在的网络环境早已不适用。NSL-KDD 是为了解决 KDD'99 数据集的一些固有问题而提出的数据集。虽然，这个新版本的 KDD 数据集仍然存在 McHugh 所讨论的一些问题，并且可能不能完美地代表现有的真实网络，但由于缺乏基于网络的 IDS 的公共数据集，我们相信它仍然可以作为一个有效的基准数据集来帮助研究人员比较不同的入侵检测方法。

此外，NSL-KDD 训练集和测试集的记录数量是合理的。这一优势使得在完整的集合上运行实验是经济实惠的，而不需要随机选择一小部分。因此，不同研究工作的评价结果将具有一致性和可比性。

CICIDS2017 数据集包含了良性的和最新的常见攻击，与真实的现实世界数据（PCAPs）相似。它还包括使用 CICFlowMeter 进行网络流量分析的结果，并根据时间戳、源和目的 IP、源和目的端口、协议和攻击（CSV 文件）对流量进行了标注。同时还提供了提取的特征定义。

生成真实的背景流量是我们构建这个数据集的首要任务。我们使用了我们提出的 B-Profile 系统（Sharafaldin，等人，2016）来对人类交互的抽象行为进行剖析，并生成自然的良性背景流量。对于这个数据集，我们基于 HTTP、HTTPS、FTP、SSH 和电子邮件协议建立了 25 个用户的抽象行为。

数据采集期从 2017 年 7 月 3 日（周一）上午 9 点开始，到 2017 年 7 月 7 日（周五）下午 5 点结束，共 5 天。其中周一为正常日，只包括良性流量。实施的攻击包括蛮力 FTP、蛮力 SSH、DoS、Heartbleed、Web 攻击、渗透、僵尸网络和 DDoS。它们在周二、周三、周四和周五的上午和下午都被执行过。

THU-IDS 清华校园网数据集，该数据集为真实流量，将于第三章进行介绍。

2.5　异常检测算法对比

对比不同机器学习方法在 NSL-KDD 数据集上的效果，

2.6　现有异常检测算法存在的问题

近些年出现了数以千计的异常检测算法，它们的检测原理，适用范围以及所使用的流量特征各不相同。通过上述实验室对比我们可以看出，在应对大规模、应用类型多、异常流量是常态且多种异常相互叠加的场景下，现有异常检测算法大多存在以下缺陷：1. 面对海量数据规模时，无法或很难做到实时性；2. 应用类型

多导致的流量特征复杂，因此很难达到较高的检测率和较低的误报率；3. 大多数异常检测算法仅能报告是否发生异常，难以对确定异常种类和定位异常来源给出指导性意见。由于上述缺陷，当前大多数异常检测算法仍处于概念验证或模拟实现的阶段，距离实际的部署和应用还有不小的距离。

第 3 章　数据集分析与对比

3.1　开源数据集

3.1.1　UNSW-NB15

UNSW-NB 15 数据集[32] 的原始网络数据包是由 IXIA PerfectStorm 工具在澳大利亚网络安全中心（ACCS）的网络范围实验室中创建的，该数据集主要是工具生成的正常活动和人为构造的多种攻击行为的混合体。相比于更加古老的 KDD99[33]，其更能代表真实的网络流量。该数据集包括 100GB 的.pcap 格式的原始网络流量，具有九种攻击类型，分别是 Fuzzers，Analysis，Backdoors，DoS，Exploits，Generic，Reconnaissance，Shellcode 和 Worms。每种攻击的具体描述和类别数目如表3.1所示。该数据集还包含 4 个经过特征提取的 csv 文件，一共 2540044 条数据。

类别	类别描述	样本数量
Normal	正常流量	2218761
Fuzzers	攻击者从命令行或以报文的形式发送大量随机生成的输入序列。攻击者试图发现操作系统、程序或网络中的安全漏洞，并使这些资源挂起一段时间，甚至可以使它们崩溃。	24246
Analysis	这类攻击是指通过端口扫描、恶意 web 脚本 (如 HTML 文件渗透) 和发送垃圾邮件等各种方式渗透到 web 应用程序的各种入侵等。	2677
Backdoor	这类攻击中攻击者可以绕过正常的身份验证并获得对系统的未授权远程访问。黑客利用后门程序安装恶意文件，修改代码或获得对系统或数据的访问。	2329
DoS	攻击者使某些计算或内存资源过于繁忙或占据全部资源而无法处理合法请求或者拒绝合法用户对计算机的访问。	16353
Exploit	利用操作系统或软件中的软件漏洞、漏洞或故障进行入侵的行为。攻击者利用软件的知识发动攻击，意图对系统造成危害。	44525
Generic	针对密码系统的攻击，试图破坏安全系统的密钥。它独立于密码系统的实现细节。不考虑块密码的结构。例如，生日攻击是一种将哈希函数视为黑盒的通用攻击。	215481
Reconnaissance	为了绕过目标计算机网络的安全控制而收集其信息的攻击。它可以被定义为一个探针，是发起进一步攻击的初步步骤。攻击者使用各种扫描手段来收集系统信息。在收集到足够的信息后，可以发起后续的攻击。	13987
Shellcode	Shellcode 作为负载在目标机器执行，来挖掘该软件的漏洞。之所以称作 Shellcode 是因为启动了受到攻击者控制的命令行 shell。	1511
Worm	蠕虫是一种恶意程序或恶意软件，它可以复制自己并传播到其他计算机。	174

表 3.1　UNSW-NB15 数据集攻击类别

UNSW-NB15 数据集一共包含 47 个特征，其中时间戳，IP 地址，端口号等特征对训练无用，因此有效的特征一共 41 个。下面对这些特征做一个概括的说明。按照数据集作者的思路，可以分为基本特征，内容特征，时间特征和额外生成的特征这几类。这里从另外一种思路进行重新归类可以分为以下几类。

表 3.2　与协议相关的特征

特征名称	特征描述
proto	传输层协议
service	应用层协议
sttl	从源发出的报文的 time to live
dttl	从目的发出的报文的 time to live 字段
stcpb	源 tcp 报文的初始序列号
dtcpb	目的 tcp 报文的初始序列号
swin	源 tcp 报文的窗口字段
dwin	目的 tcp 报文的窗口字段
res_bdy_len	http 响应内容的长度
ct_flw_http_method	会话中 http 方法字段的计数
is_ftp_login	是否有 ftp 的登录
ct_ftp_cmd	会话中 ftp 命令的计数
trans_depth	http 服务的连接深度

表 3.3　与时间相关的特征

dur	会话的持续时间
tcprtt	tcp 三次握手的 rtt（round trip time）
synack	第一次发送到确认的时间
ackdat	确认之后返回的时间
sintpkt	源报文的间隔时间的平均值
dintpkt	目的报文间隔时间的平均值
sjit	源报文间隔时间的标准差 (jitter)
djit	目的报文间隔时间的标准差
sload	源报文的吞吐量
dload	目的报文的吞吐量

表 3.4　与报文大小相关的特征

sbytes	会话中从源发出的总字节数
dbytes	会话中从目的发出的总字节数
smeansz	会话中源报文的平均大小
dmeansz	会话中目的报文的平均大小
spkts	会话中源的报文总数
dpkts	会话中目的报文总数

表 3.5　与连接状态相关的特征

sloss	源的丢包数和重传数之和
dloss	目的的丢包数和重传数之和
state	会话的状态和相应的协议

表 3.6　额外构造的特征

ct_srv_src	根据最后一条报文的时间排序，每 100 条记录中源 ip 与服务都相同的会话计数（下面省略每 100 条）
ct_srv_dst	目的 IP 与服务都相同的会话计数
ct_dst_ltm	目的 IP 相同的会话计数
ct_src_ltm	源 IP 相同的会话计数
ct_src_dport_ltm	源 IP 和目的端口都相同的会话计数
ct_dst_sport_ltm	目的 IP 和源端口都相同的会话计数
ct_dst_src_ltm	目的 IP 和源 IP 都相同的会话计数
ct_state_ttl	对于每一个状态，ttl 值的范围
is_sm_ips_ports	源 ip 与目的 ip, 源端口和目的端口是否都相同

可以看到，UNSW-NB15 数据集大多数特征都有较为清晰的定义，因此可以用做流量数据特征提取的标准。

3.1.2　CICIDS2017

CICIDS 数据集特征介绍如下：（修改格式，解释表格）

List of extracted features and descriptions:

表 3.7　packet 大小相关的特征

total Length of Fwd Packet	发送方 packet 总长度
total Length of Bwd Packet	接收方 packet 总长度
Fwd Packet Length Min	发送方 packet 最小长度
Fwd Packet Length Max	发送方 packet 最大长度
Fwd Packet Length Mean	发送方 packet 平均长度
Fwd Packet Length Std	发送方 packet 长度标准差
Bwd Packet Length Min	接收方 packet 最小长度
Bwd Packet Length Max	接收方 packet 最大长度
Bwd Packet Length Mean	接收方 packet 平均长度
Bwd Packet Length Std	接收方 packet 长度标准差

表 3.8　时间相关的特征

Flow Bytes/s	bps
Flow Packets/s	pps
Flow IAT Mean	一条流中包平均间隔时间
Flow IAT Std	一条流中包间隔时间的标准差
Flow IAT Max	一条流中两个 packet 最大间隔时间
Flow IAT Min	一条流中两个 packet 最小间隔时间
Fwd IAT Min	发送方两个 packet 最小间隔时间
Fwd IAT Max	发送方两个 packet 最大间隔时间
Fwd IAT Mean	发送方两个 packet 平均间隔时间
Fwd IAT Std	发送方两个 packet 间隔时间的标准差
Fwd IAT Total	发送方两个 packet 总间隔时间
Bwd IAT Min	接收方两个 packet 最小间隔时间
Bwd IAT Max	接收方两个 packet 最大间隔时间
Bwd IAT Mean	接收方两个 packet 平均间隔时间
Bwd IAT Std	接收方两个 packet 间隔时间的标准差
Bwd IAT Total	接收方两个 packet 总间隔时间

表 3.9　标志位相关的特征

Fwd PSH flags	发送方 PSH 标志位出现次数 (0 for UDP)
Bwd PSH Flags	接收方 PSH 标志位出现次数 (0 for UDP)
Fwd URG Flags	发送方 URG 标志位出现次数 (0 for UDP)
Bwd URG Flags	接收方 URG 标志位出现次数 (0 for UDP)
FIN Flag Count	FIN 标志位出现次数
SYN Flag Count	SYN 标志位出现次数
RST Flag Count	RST 标志位出现次数
PSH Flag Count	PUSH 标志位出现次数
ACK Flag Count	ACK 标志位出现次数
URG Flag Count	URG 标志位出现次数
CWR Flag Count	CWR 标志位出现次数
ECE Flag Count	ECE 标志位出现次数

3.2　校园网真实数据集

清华大学无线网规模巨大，具有超过 6 万名活跃用户，13 万个可用 ip 地址，是全球最大的校园无线网之一。本节共采集了三类数据集，分别是出口网关处的全流量日志数据、NOC（Network Operation Center）平台的威胁告警日志、TCP 流量和 UDP 流量日志。接下来分别介绍着四类数据。

3.2.1　全流量日志数据

校园网的原始数据为抓包（Packet Capture，pcap）流量，如下图 3.1所示，包含每个数据包的详细信息，如五元组、报文内容等。

对于机器学习模型来说，通常不是直接对原始的流量进行检测，而是通过对流量提取特征，得到样本的特征向量后进行检测和分类。因此数据集的制作流程如图所示:

3.2.2　威胁告警日志

经过第 5 章的特征提取模块，我们得到了特征数据集，但是由于缺乏标注，该数据集无法进行训练以及验证。因此为了得到可进行训练的有效数据集，我们还需要对特征数据集进行标注。标注数据是根据现有的 NOC 平台得到，需要进行数据清洗、数据预处理等步骤。下图为 NOC 平台的数据样例。

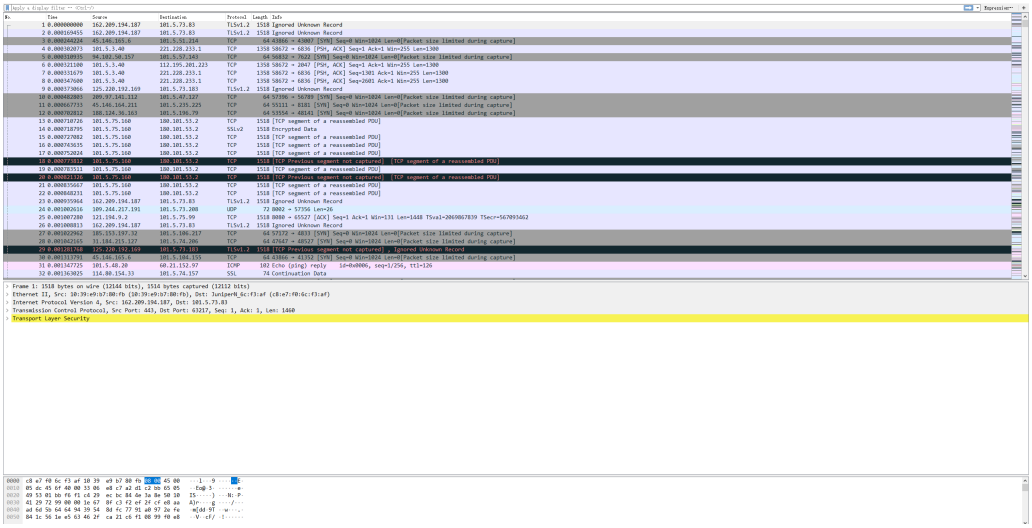

图 3.1　流量数据示意图

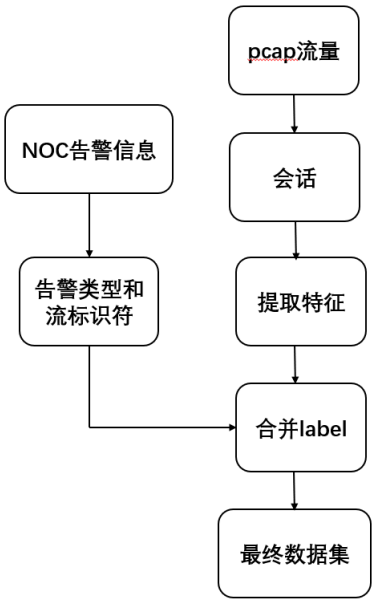

图 3.2　流量数据集制作流程

3.2.3　TCP 流量和 UDP 流量日志

3.3　数据预处理及评价指标

数据预处理阶段，首先对数据进行数据清洗，消除不必要的噪声数据。后续操作总结如下：

1. 删除套接字信息：由于原始数据集包含网络中源主机和目标主机的 IP 地址和端口号，因此删除这些信息以提供无偏检测非常重要，使用这些信息可能会导致对该套接字信息的过度训练。

2. 删除空格：数据集中的一些多类标签包含空格。由于实际值不同于同一类

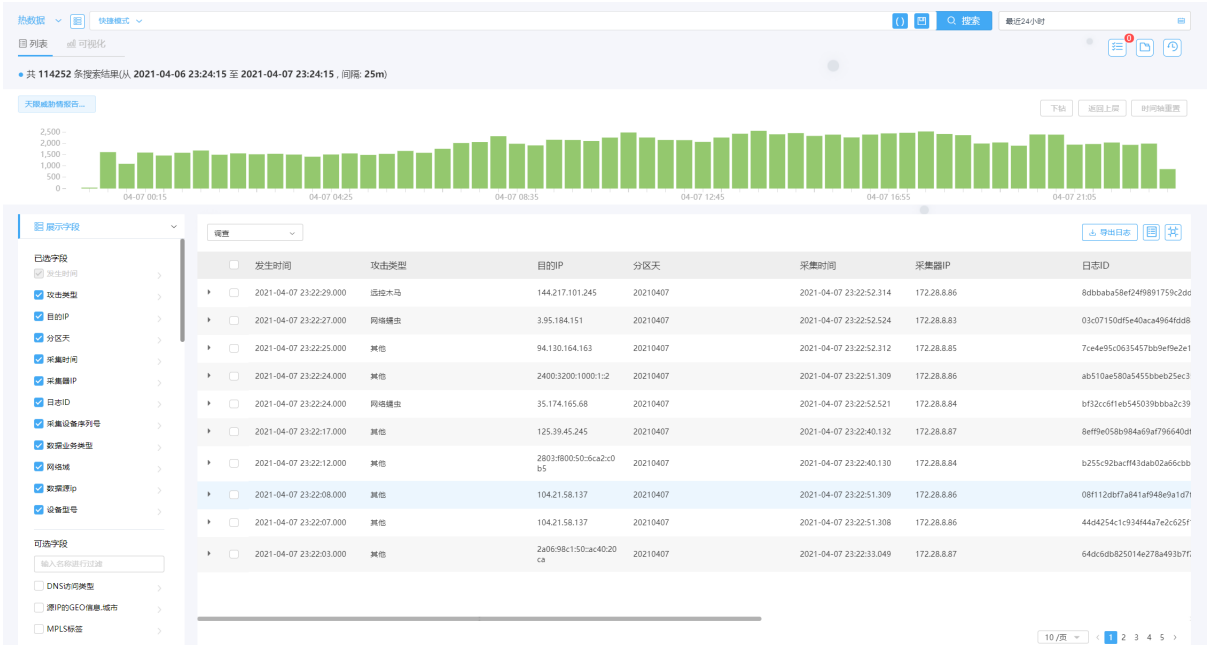

图 3.3　NOC 平台数据样例

中其他元组的标签，因此这些空白会导致不同的类。

3. 标签编码：数据集中的多个类标签都有攻击的名称，即字符串值。因此这些值编码成数值是很重要的，分类器就可以学习每个元组所属的类号。

4. 数据规范化：数据集中的数值数据属于不同的范围，这给分类器在训练过程中补偿这些差异带来了一些挑战。因此，规范化每个属性中的值很重要，这样，每个属性中的最小值为零，而最大值为一。这为分类器提供了更均匀的值，同时保持了每个属性值之间的相关性。

在实验中，本论文使用四个指标来评估模型的性能：准确率，精确度，召回率和 $F1-score$。对于一个数据样本检测的结果可分为下面的结果：

1. TP：入侵网络流量被入侵检测正确的识别为入侵网络流量。
2. TN：正常网络流量被入侵检测正确的识别为正常网络流量。
3. FN：入侵网络流量被入侵检测错误的识别为正常网络流量。
4. FP：正常网络流量被入侵检测错误的识别成入侵网络流量。

论文使用下面的方法去评估我们提出方法的性能：准确率（Accuracy）：预测正确的样本的数量与所有被预测样本数量的比值，公示如下所示。

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3-1}$$

精确度（Precision）：这个指标于分类器衡量的被错误分类的数量所惩罚的正确分类的数量，公示如下所示

$$Precision = \frac{TP}{TP + FP} \tag{3-2}$$

召回率（Recall）：针对样本而言的表示的是数据正样本中有多少被预测正确。这种度量反映了分类器检测网络攻击的能力，公示如下所示

$$Recall = \frac{TP}{TP + FN} \tag{3-3}$$

$F1 - score$：该度量精确度和召回率的调和平均值，可以平衡的反映算法的精确度，公示如下所示

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3-4}$$

3.4　现有算法在不同数据集的评估结果

本节分别在以上三个数据集上进行实验，对比了现有的经典算法，这些算法的名称及特点见下表。

待补充结果对比图

第 4 章　基于特征关系图的 RNN 及其网络流量异常检测算法

4.1　引言

神经网络（Neural Network，NN），又称人工神经网络 (Artificial Neural Network，ANN)，是 20 世纪 80 年代以来人工智能领域兴起的研究热点。它的定义有很多，其中第一批神经计算机的发明者 Robert Hecht-Nielsen 博士对神经网络的定义是“一个由许多简单的，高度互连的处理元素组成的计算系统，它们通过对外部输入的动态状态反应来处理信息。或者也可以认为人工神经网络是一种计算模型，它的灵感来自于人脑中生物神经网络处理信息的方式。”最近十多年来，针对人工神经网络的研究工作已经取得了重大进展，其在模式识别、自动控制、预测估计等领域已成功地解决了许多现代计算机难以解决的实际问题。

4.1.1　神经网络的类型

神经网络有很多类，这些类也有子类，最常用的类有如下几种。

1. 前馈神经网络. 前馈神经网络是一种人工神经网络，单元之间的连接不形成循环。在这种网络中，信息只有一个方向，即向前移动，从输入节点，通过隐藏节点（如果有的话），然后到输出节点。网络中不存在循环或环路。我们可以区分两种类型的前馈神经网络。
   (a) 单层感知器。这是最简单的前馈神经网络，不包含任何隐藏层，也就是说它只由单层的输出节点组成。之所以说是单层，是因为我们在计算层数的时候，并不包括输入层，原因是在输入层没有进行计算，输入通过一系列的权重直接反馈给输出。
   (b) 多层感知器 (MLP)。这类网络由多层计算单元组成，通常以前馈方式相互连接。一层中的每个神经元都与后一层的神经元有定向连接。在许多应用中，这些网络的单元应用一个 sigmoid 函数作为激活函数。MLP 是非常有用的，一个很好的原因是，它们能够学习非线性表示（大多数情况下，呈现给我们的数据是不可线性分离的。
   (c) 卷积神经网络 (CNN)。卷积神经网络与普通的神经网络非常相似，它们是由具有可学习权重和偏差的神经元组成的。在卷积神经网络（CNN，或 ConvNet 或移位不变或空间不变）中，单元连接模式的灵感来自于视觉皮层的组织，单元在一个被称为感受场的受限空间区域内对刺激做出

反应。感受场部分重叠，覆盖了整个视场。单元响应可以用卷积运算在数学上近似。它们是多层感知器的变体，使用最少的预处理。它们的广泛应用是在图像和视频识别，推荐系统和自然语言处理。CNNs 需要大量的数据来进行训练。

2. 循环神经网络在循环神经网络 (RNN) 中，单元之间的连接形成了一个定向循环 (它们向前传播数据，同时也向后传播数据，从较后的处理阶段到较早的阶段)。这使得它能够表现出动态的时间行为。与前馈神经网络不同，RNNs 可以利用其内部存储器处理任意输入序列。这使得它们适用于未分割、连接的手写识别、语音识别和其他一般序列处理器等任务。

神经网络的灵感来自于人脑生物神经网络的处理方式。大脑的基本计算单位是神经元。在人类的神经系统中，大约有 860 亿个神经元，它们与大约 10^{14}-10^{15} 的突触相连。神经网络的基本计算单位也是神经元，通常称为节点或单位。它从其他一些节点或外部源接收输入，并计算输出。每个输入都有一个相关的权重 (w)，该权重是根据其对其他输入的相对重要性分配的。节点对其输入的加权和应用一个函数。其想法是，突触强度（权重 w）是可学习的，并控制影响的强度及其方向：一个神经元对另一个神经元的兴奋性（正权重）或抑制性（负权重）

假设一个神经元接收 D 个输入 $x_1, x_2, ..., x_D$ 令向量 $x = [x_1; x_2; ...; x_D]$ 来表示这组输入，净输入也叫净活性值（Net Activation）. 并用净输入（Net Input）$z \in \mathbb{R}$ 表示一个神经元所获得的输入信号 x 的加权和,

$$\begin{aligned} z &= \sum_{d=1}^{D} w_d x_d + b \\ &= w^T x + b \end{aligned} \tag{4-1}$$

其中 $w = [w_1; w_2; ...; w_D] \in \mathbb{R}^D$ 是 D 维的权重向量，$b \in \mathbb{R}$ 是偏置。

净输入 z 在经过一个非线性函数 $f(\cdot)$ 后，得到神经元的活性值（Activation，

$$a = f(z) \tag{4-2}$$

其中非线性函数 $f(\cdot)$ 称为激活函数。

一个人工神经元的结构如图 4.1所示：

在图 4.1中，$\vec{x}$ 为输入向量，w 和 b 分别是权重和偏移，

神经网络主要由以下几部分组成：

- 输入节点（输入层）。在这一层中不进行任何计算，它们只是将信息传递给下一层（大部分时间是隐藏层）。

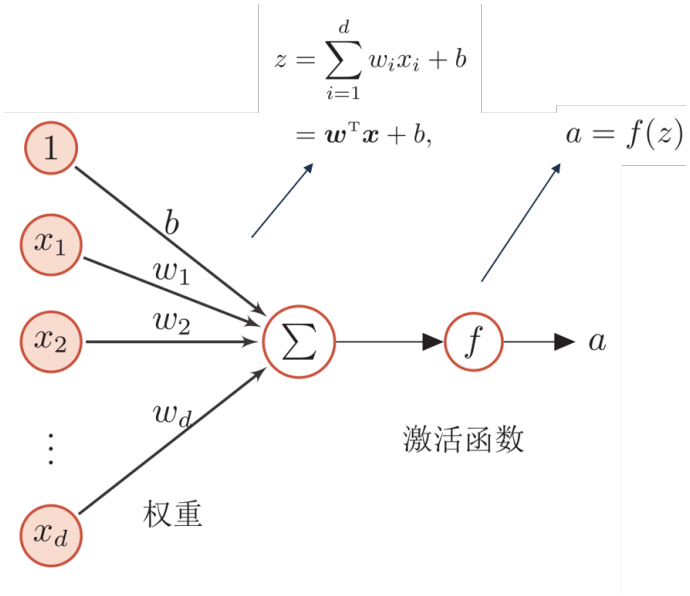

图 4.1　人工神经元模型

- 隐藏节点（隐藏层）。中间处理或计算在隐藏层中完成的，然后将输入层的权重（信号或信息）传递给下一层（另一个隐藏层或输出层）。一个神经网络也可以不包含隐藏层。
- 输出节点（输出层）。它是神经网络的最后一层，接收来自最后一个隐藏层的输入。通过激活函数可以得到合理范围内的理想数值，例如用于分类的 softmax 函数。
- 连接和权重。神经元之间会有边进行连接，每条边会有一定的权重。即每个连接将神经元 i 的输出传递给神经元 j 的输入，每个连接被赋予一个权重 W_{ij}。
- 激活函数。激活函数负责为神经网络引入非线性特征。它把值压缩到一个更小范围，即一个 Sigmoid 激活函数的值区间为 [0,1]。深度学习中有很多激活函数，如 Sigmoid、Tanh、ReLU 、Softplus、Softmax 等。下表为常见的激活函数。

表 4.1　激活函数

名称	表达式	导数
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh	$f(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ReLU	$f(x) = max(0, x)$	$f(x) = \begin{cases} 0 & \text{x<0} \\ 1 & \text{x>=0} \end{cases}$
Softplus	$f(x) = log(1 + e^x)$	$f'(x) = \frac{1}{1+e^x}$
Softmax	$S_i = \frac{e_i}{\sum_j e_j}$	

- 学习规则。学习规则是一种规则或算法，它修改神经网络的参数，以使网络的给定输入产生一个有利的输出。这个学习过程通常相当于修改权重和阈值。

4.1.2　RNN

RNN 是一种针对时序数据处理的神经网络模型。相较于普通的神经网络模型来说，更擅长处理序列数据，即该网络其具有短期记忆能力。在循环神经网络中，神经元不但可以接受其他神经元的信息，也可以接受自身的信息，形成具有环路的网络结构。

循环神经网络的通用近似定理 [Haykin, 2009]：如果一个完全连接的循环神经网络有足够数量的 sigmoid 型隐藏神经元，它可以以任意的准确率去近似任何一个非线性动力系统

下图 4.2是循环神经网络的示意图。该图显示了一个循环神经网络被展开成一

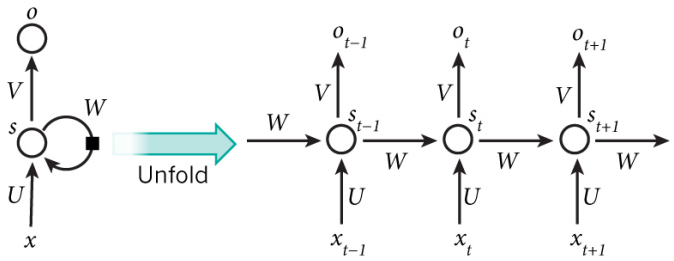

图 4.2　循环神经网络

个完整的神经网络。例如，如果输入序列是一个由 5 个单词组成的句子，那么网络就会被展开成一个 5 层神经网络，每个单词一层。这个网络在 t 时刻接收到输入 x_t 之后，隐藏层的值是 s_t ，输出值是 o_t 。关键一点是，x_t 的值不仅仅取决于 s_t ，还取决于 s_{t-1} 。用公式表示如下：

$$\begin{aligned} O_t &= g(V \cdot S_t) \\ S_t &= f(U \cdot X_t + W \cdot S_{t-1}) \end{aligned} \tag{4-3}$$

x_t 表示第 t 步的输入，例如 x_1 表示时刻 1 的特征向量。s_t 表示第 t 步隐藏层状态，也就是网络中的“记忆”。s_t 是由前一层的隐藏层状态和当前层的输入计算得到。函数 $f(\cdot)$ 通常是一个非线性函数，如 tanh 或者 ReLU。

4.1.3　LSTM

普通 RNN 有不能处理长依赖的问题，因此 Hochreiter 提出了一种长短期记忆网络-LSTM，以及它的变种，它是一种特殊的 RNN，适用于学习长期依赖。现在 LSTM 已经被广泛应用于各个领域。

所有 RNN 都具有链式形式。在普通的 RNN 中，这种循环是一种非常简单的结构，比如简单的 tanh 层。

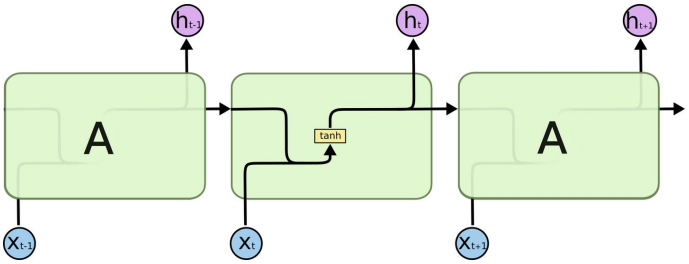

图 4.3　普通 RNN 结构

LSTM 也具有这种链式结构，但循环单元里面不再是只有单一的神经网络层，而是构建了一些“门”（Gate）。原来的 RNN，由于这种链式结构的限制，很长的时刻以前的输入对现在的网络影响非常小，后向传播时那些梯度也很难影响很早以前的输入，即会出现梯度消失的问题。而 LSTM 通过构建“门”，让网络能记住那些非常重要的信息，比如遗忘门，来选择性清空过去的记忆和更新较新的信息。LSTM 中的第一步是决定从单元状态中丢弃什么信息。通过加入一个称为忘记门

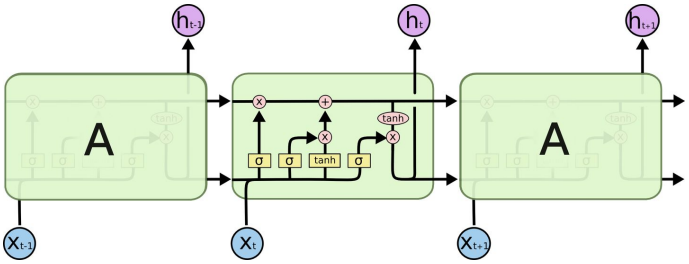

图 4.4　LSTM 结构

的 σ 层完成。遗忘门会读取上一时刻的输出 h_{t-1} 和当前时刻的输入 x_t，计算出一个维度为 n 的向量 f_t，该向量的值均在 $(0,1)$ 之间。1 表示“完全保留”上个神经元的状态信息，0 表示“完全舍弃”。

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \tag{4-4}$$

下一步是确定该神经元的哪些新状态信息被存放在单元状态中。这里包含两个部分。第一，sigmoid 层，即“输入门层”，决定 LSTM 单元将更新哪些值。然后，tanh 层创建一个新的候选值 z_t 的向量，该向量可以加入到下一层单元状态中。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4-5}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4-6}$$

最后一步是将旧单元状态 c_{t-1} 更新为新状态 c_t。把旧状态与遗忘门 f_t 相乘，丢弃掉之前需要丢弃的信息。接着与新状态进行相加。综合得出该神经元的输出的状态，也即更新单元的状态。

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{4-7}$$

4.1.4　GRU

循环门单元（Gated Recurrent Unit，GRU），由 Cho, et al. (2014) 提出。它组合了遗忘门和输入门到一个单独的“更新门”中。它也合并了 cell state 和 hidden state，并且做了一些其他的改变。结果模型比标准 LSTM 模型更简单，

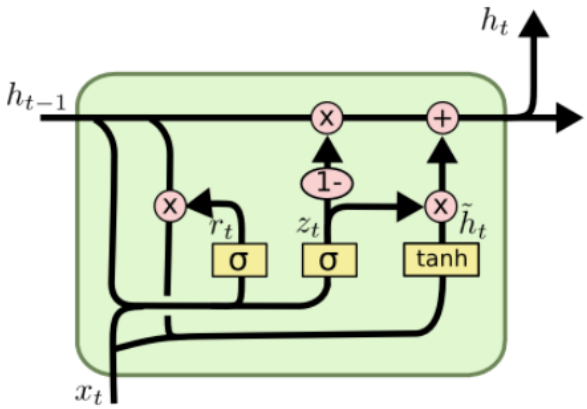

图 4.5　GRU 结构

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{4-8}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{4-9}$$

$$\widetilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{4-10}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{4-11}$$

由图中结构可以看出，GRU 是通过一个循环神经网络和“门”机制来不断更新内部参数，GRU 算法的伪代码如下表所示。

算法 4.1 *GRU*

输入： t
输出： 内部参数
1: 初始化 t 时刻单元状态
2: **for** t ← 1 to T **do**
3: $output_t$ = t
4: $state_t = output_t$
5: **end for**

算法 4.2 *LSTM 层*

输入： 一组按时间排列的向量组
输出： 按时间排列的向量组
1: $\vec{C}_0 = \vec{0}$
2: $\vec{h}_0 = \vec{0}$
3: **for** t ←1 to T **do**
4: $output_t = activation(dot(W, input_t) + dot(U, state_t) + b)$
5: $state_t = output_t$
6: **end for**

4.2　基于关系图结构的 RNN

有向权重图 $G = (V, E, W)$,V 表示特征节点的集合，其中 $|V| = N$, E 表示特征间的关联关系，即图中的边，$W \in R[N * N]$ 为特征节点的相似度的加权邻接矩阵。将流量表示为 G 的一个图信号，P 为每个节点特征数，则-时间 t 观察到的图信号。那么流量预测目的就是：给定 G 下，学得一个函数将 T’ 个历史图信号映射到未来 T 时刻的图信号：

$$h[X^{t-T+1}, X^{t-T+2}, ..., X^t; G] \Rightarrow [X^{t+1}, X^{t+2}, ..., X^{t+T}] \tag{4-12}$$

$$r^{(t)} = \sigma(\Theta_r \star G[X^{(t)}, H^{t-1}] + b_r) \tag{4-13}$$

$$u^{(t)} = \sigma(\Theta_u \star G[X^{(t)}, H^{t-1}] + b_u) \tag{4-14}$$

$$C^{(t)} = tanh(\Theta_C \star_G [X^{(t)}, (r^{(t)} \odot H^{(t-1)})] + b_c) \tag{4-15}$$

$$H^{(t)} = u^{(t)} \odot H^{(t-1)} + (1 - u^{(t)}) \odot C^{(t)} \tag{4-16}$$

其中 $X^{(t)}$, $H^{(t)}$ 表示在时间 t 的输入和输出，$r^{(t)}$ $u^{(t)}$ 分别是在时间 t 的复位门和更新门。$\star_G$ 表示在等式 2 中定义的扩散卷积，并且是对应滤波器的参数。与 GRU 相似，DCGRU 可用于构建递归神经网络层，并使用反向传播进行训练。

我们利用递归神经网络（RNN）对时间依赖性进行建模。特别是，我们使用门控循环单元（GRU）（Chung 等，2014），它是 RNN 的简单而强大的变体。我们用扩散卷积代替了 GRU 中的矩阵乘法，这导致了我们提出的扩散卷积门控递归单元（DCGRU）。

通常，计算卷积会很费时。但是，如果 G 稀疏，则可以使用总时间复杂度 $O(K \mid \varepsilon \mid \ll O(N^2)$ 递归稀疏矩阵乘法来有效地计算方程 2。

$$hl = [hl_1,\ hl_2,\ \cdots,\ hl_{n-f+1}] \tag{-}$$

4.3　算法性能评估

为了验证 FG-RNN 模型的有效性，本节首先在清华大学无线校园网真实场景下的数据集上开展有效性评估，随后在异常检测领域下的两个经典的公开数据集上进行实验，分别对比了 FG-RNN 和多个模型的评估效果，最后分析了模型参数的敏感性。

其中多分类任务的准确率指标为对于每个标签，分别计算 precision，然后取不加权平均。

4.3.1　实验参数设置

在本节中，

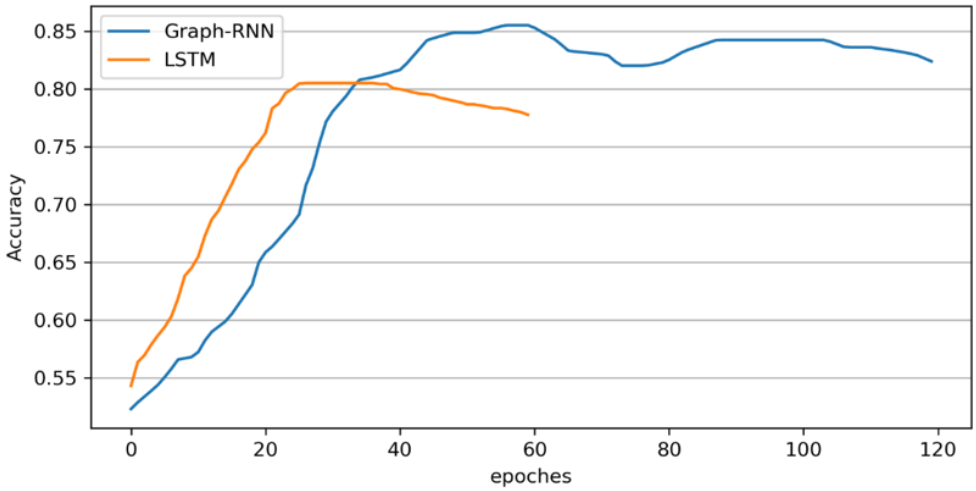

图 4.6　结果 1

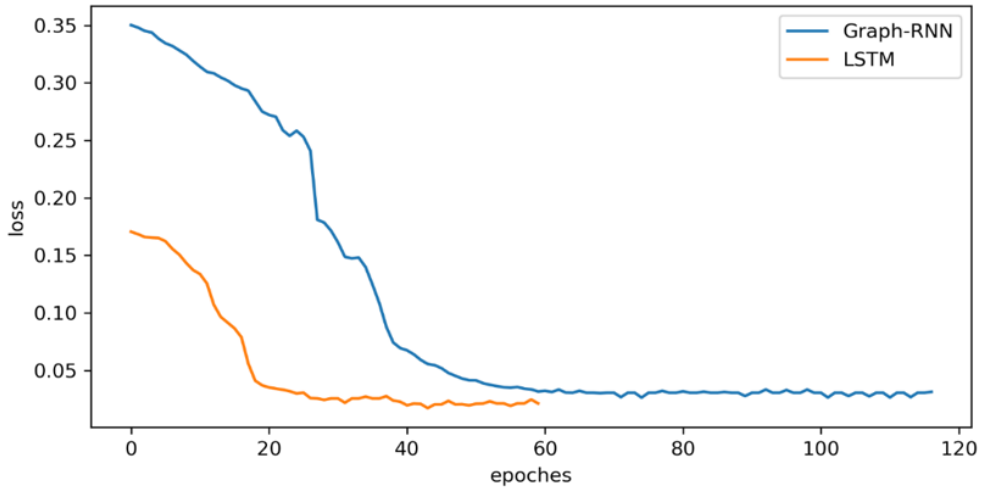

图 4.7　结果 1

TODO: 格式待调整

4.3.2　基于开源数据集的检测结果

4.3.3　基于真实数据的检测结果

交叉熵损失函数经常用于分类问题中，特别是在神经网络做分类问题时，也经常使用交叉熵作为损失函数，此外，由于交叉熵涉及到计算每个类别的概率，所以交叉熵几乎每次都和 sigmoid(或 softmax) 函数一起出现。

我们用神经网络最后一层输出的情况，来观察整个模型预测、获得损失和学

表 4.2　部分评估结果，待填充具体数值

数据集	任务	LR	NB	DT	CNN	CNN-LSTM	GRU	DCRNN-A	DCRNN-B
UNSW-NB15	二分类	0.848	79.33	78.52	80.51	62.74	68.91	**82.69**	81.66
	多分类	76.73	77.96	76.82	77.98	47.11	56.30	**81.05**	79.94
NSL-KDD	二分类	68.26	67.11	65.54	70.18	65.04	58.49	70.26	**70.88**
	多分类	67.01	67.37	66.41	68.31	56.16	53.18	68.47	**70.24**
CAMPUS	二分类	79.98	77.95	79.51	75.62	79.80	75.25	**80.69**	79.10
	多分类	76.33	77.01	77.54	73.01	76.59	59.28	78.12	**78.89**

算法 4.3 *FG-RNN*

输入： 一组按时间排序的向量组 $\vec{x}_1,\vec{x}_2,...,\vec{x}_T$, 特征间关系图 G 以及它的邻接矩阵 W
输出： 按时间排序的向量组 $\vec{h}_1,\vec{h}_2,...,\vec{h}_T$

通过 Xavier 方法初始化 q_{net1}, q_{net2} and p_{net} 的所有参数
while $\mathcal{L}$ 没有收敛 **do**
    利用公式计算每一个节点 v 的 $q_\phi(\boldsymbol{y}_v|\boldsymbol{A},\boldsymbol{X},Y_L)$ 以及 h_v^{K-1}
    for i ← 1 to m **do**
        从分布 $q_\phi(Y_U|\boldsymbol{A},\boldsymbol{X},Y_L)$ 中采样 Y_U
        在采样得到的 Y_U 基础上，利用公式计算 $q_\phi(\boldsymbol{z}|\boldsymbol{A},\boldsymbol{X},Y)$
        for j ← 1 to n **do**
            从分布 $q_\phi(\boldsymbol{z}|\boldsymbol{A},\boldsymbol{X},Y)$ 中采样 $\boldsymbol{z}$
            在采样得到的 $\boldsymbol{z}$ 基础上用公式计算 $p_\theta(\boldsymbol{y}_v|\boldsymbol{A},\boldsymbol{X},\boldsymbol{z})$
        end for
    end for
    用公式计算目标函数 $\mathcal{L}(\theta,\phi)$
    用梯度下降法更新 q_{net1}，q_{net2} 和 p_{net} 的所有参数
end while

习的流程：

神经网络最后一层得到每个类别的得分 scores；该得分经过 sigmoid(或 softmax) 函数获得概率输出；模型预测的类别概率输出与真实类别的 one hot 形式进行交叉熵损失函数的计算。

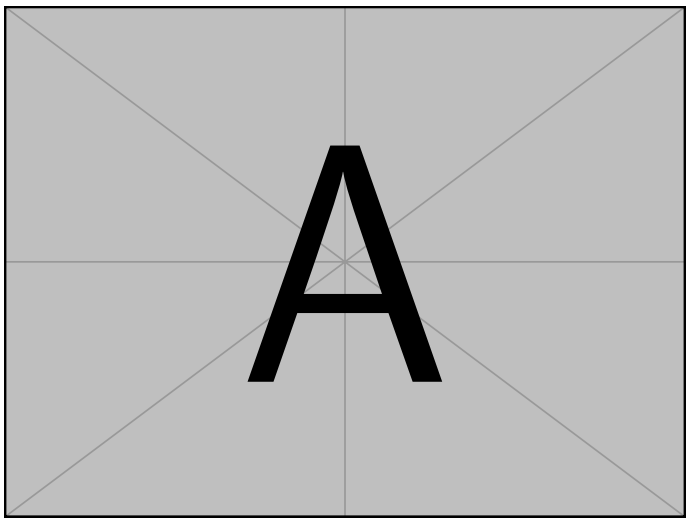

图 4.8　Example figure

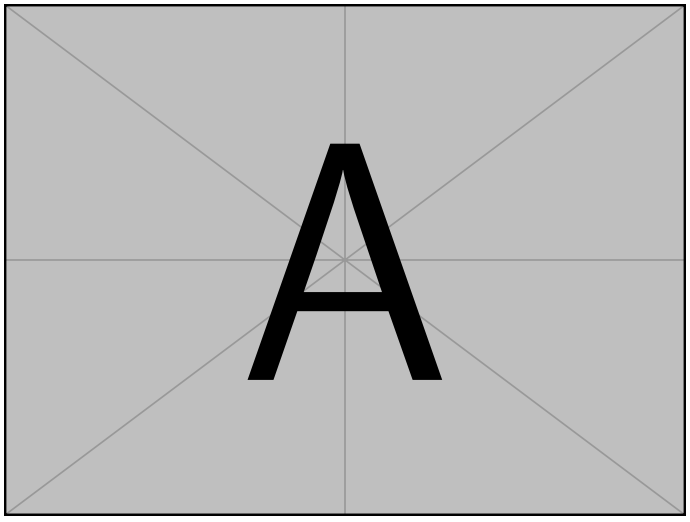

图 4.9　Example figure

第 5 章　流量检测系统的设计与实现

5.1　引言

本文实现的流量异常检测系统按照功能需求可以分为三个模块：数据采集和分析模块，异常检测模块，终端预警模块。TODO: 图中加入关系矩阵

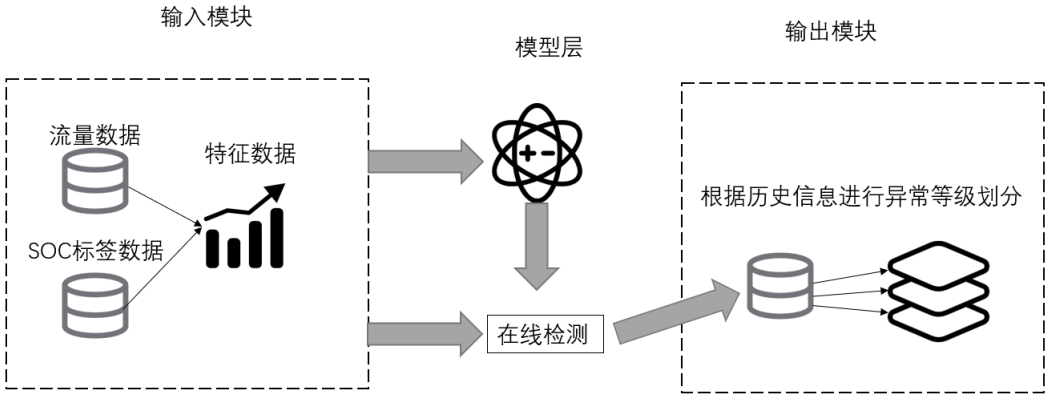

图 5.1　系统架构图

下面是对各个模块的介绍：

待补充。

5.2　spark 平台介绍

Apache Spark 是由 UC Berkeley AMP Lab 开源的用于大规模数据处理的统一分析引擎[34]，现如今已经成为 Apache 软件基金会的顶级开源项目。Spark 原理与 Hadoop 类似，都是基于 MapReduce 进行分布式计算，但是功能更加丰富，且支持 SQL 查询、流式处理、图计算等功能。Spark 具有以下几个特点：

- 速度快，Spark 的速度比 Hadoop 要快 100 倍以上，图5.2是一项逻辑回归任务在 Hadoop 和 Spark 两个系统的运行时长对比。这是因为 Spark 基于内存计算，而 Hadoop MapReduce 必须进行读取和写入磁盘，IO 操作比内存操作更加耗时。
- 易用性。Spark 提供了 80 余种高级 API，可以轻松编写高度并行的应用程序。Spark 支持 Java, Scala, Python, R, and SQL 等多种编程语言。
- 兼容性。Spark 可以运行在 Hadoop 模式、Mesos 模式、Standalone 独立模式或 Cloud 中，并且还可以访问各种数据源，包括本地文件系统、HDFS、Cassandra、

HBase 和 Hive 等。

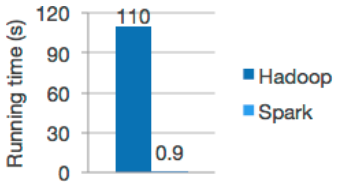

图 5.2　Logistic regression in Hadoop and Spark

如图5.3所示，Spark 项目包含多个独立组件。其最核心的模块为深蓝色的 Spark Core，其定义了核心 API，如弹性分布式数据集（Resilient Distributed Datasets，RDD），该模块也负责调度、监控计算集群中的计算任务，具有内存管理，故障恢复，与管理系统、存储系统交互等功能。为了满足分布式计算系统的可扩展性，Spark 支持在各种集群管理器之上运行，例如图中灰色的模块 Hadoop YARN，Apache Mesos 以及浅蓝色的 Spark 自身的“独立调度程序”（Standalone Scheduler）。在 Spark Core 之上，具有四个组件：Spark SQL、Spark Streaming、MLlib、GraphX。本文中的系统主要使用 Spark Streaming 组件。

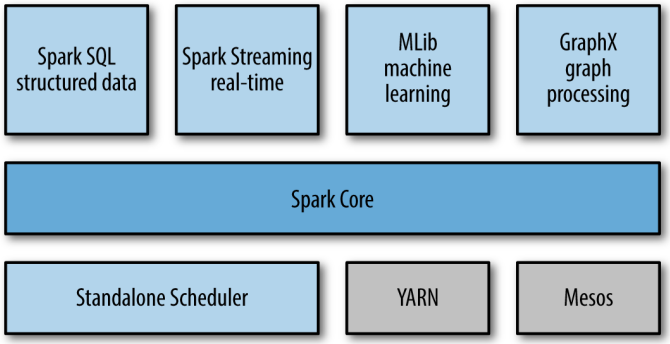

图 5.3　Spark 组件

5.3　系统整体设计

Spark Streaming 是 Spark Core 为支持流式处理的扩展，具有高容错，高吞吐的特点。Spark Streaming 在 Spark Core 的基础上继续实现 StreamingContext（负责管理 Spark Streaming 程序运行的环境），JobScheduler（负责调度 Job 任务），JobGenerator（负责生成 Job 任务），Receiver（负责接收数据）组件。

Spark Streaming 内部的基本工作原理如图 5.4所示，接收实时输入数据流，然后将数据拆分成多个 batch，比如每收集 1 秒的数据封装为一个 batch，然后将每个 batch 交给 Spark 的计算引擎进行处理，最后会生产出一个结果数据流，其中的数据，也是由一个一个的 batch 所组成的。

图 5.4　spark 工作原理

DStream 是 Spark Streaming 中对数据流的一个抽象。DStream 以哈希表的方式保存一系列连续的 RDD，哈希表的键是时间，值是 RDD。随着时间的推移，DStream 中产生新的 RDD，形成一个 RDD 流，但是由于 RDD 还是批处理方式，而且往往 RDD 中的数据量相对较小，所以 SparkStreaming 对于数据的处理又称为微批处理。此外 DStream 和 RDD一样，提供 map、foreach、flatMap、transform 等算子。

Spark Streaming 支持基本数据源和高级数据源，其中基本数据源包括 hdfs（hadoop File System）文件源，简单 socket 源等，高级数据源包括 kafka，Flume 等。Spark Streaming 在执行的过程可以分成两个阶段，数据准备阶段和数据计算阶段，以下分别介绍。

下图 5.5展示 Spark Streaming 接收数据的示意图，其中 BlockGenerator，Receiver 都是 Spark Streaming 实现的组件，blockForPushing 是一个阻塞队列。在 Spark Streaming 数据准备阶段，首先 Receiver 从外部数据源接收数据并且将数据写入 BlockGenerator 的 ArrayBuffer（内存数组）中，该过程会保证数据接收速率始终小于设置的最大速率 maxRate，这个 maxRate 也可以由 Spark 的反压模块根据 Spark 集群的处理速度自行计算。然后，BlockGenerator 包含定时器 Timer 会定时将 ArrayBuffer 中的数据生成 Block 对象（块对象，其中包含生成的数据），并且将 Block 对象写入 BlockForPushing 阻塞队列中，然后再通知 SparkStreaming 的 Driver 程序，将 Block 对象分配至 Block 队列。最终，数据在 Block 队列中等待 Spark Streaming 的计算阶段。以上便是 Spark Streaming 接收数据的全部过程。

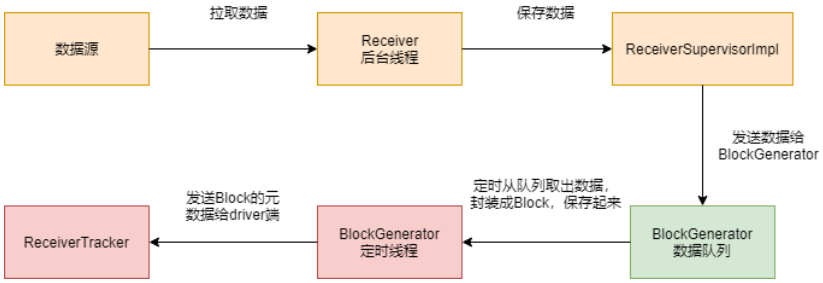

图 5.5　Spark 数据源

下图 5.6是 Spark Streaming 作业的执行流程，具体流程分为以下几步：

1. 客户端提交作业后启动 Driver，Driver 是 park 作业的 Master。
2. 每个作业包含多个 Executor，每个 Executor 以线程的方式运行 task，Spark

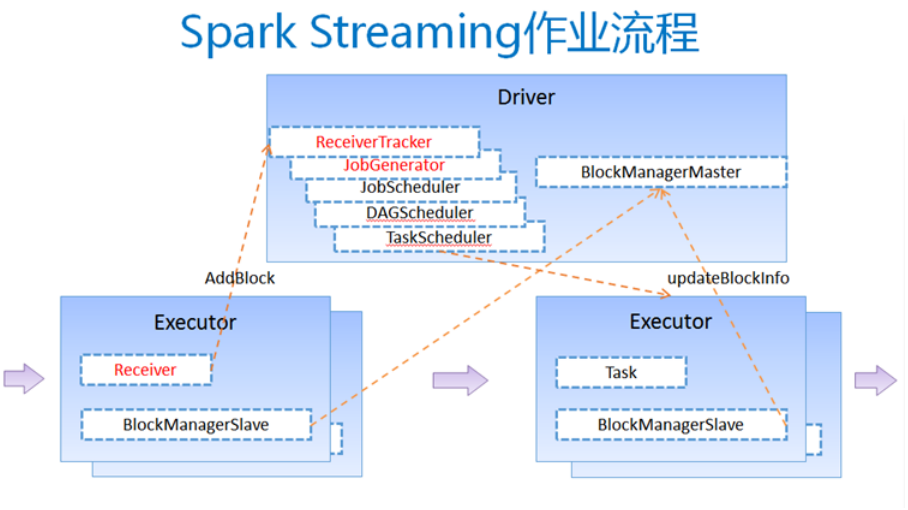

图 5.6　spark streaming 执行流程

Streaming 至少包含一个 receiver task。

3. Receiver 接收数据后生成 Block，并把 BlockId 汇报给 Driver，然后备份到另外一个 Executor 上。
4. ReceiverTracker 维护 Reciver 汇报的 BlockId。
5. Driver 定时启动 JobGenerator，根据 Dstream 的关系生成逻辑 RDD，然后创建 Jobset，交给 JobScheduler。
6. JobScheduler 负责调度 Jobset，交给 DAGScheduler，DAGScheduler 根据逻辑 RDD，生成相应的 Stages，每个 stage 包含一到多个 task。
7. TaskScheduler 负责把 task 调度到 Executor 上，并维护 task 的运行状态。
8. 当 tasks，stages，jobset 完成后，单个 batch 才算完成。

运行一个 Spark Streaming 应用程序，有下面一些步骤

有管理器的集群-这是任何 Spark 应用程序都需要的需求，详见部署指南将应用程序打为 jar 包-你必须编译你的应用程序为 jar 包。如果你用 spark-submit 启动应用程序，你不需要将 Spark 和 Spark Streaming 打包进这个 jar 包。如果你的应用程序用到了高级源（如 kafka，flume），你需要将它们关联的外部 artifact 以及它们的依赖打包进需要部署的应用程序 jar 包中。例如，一个应用程序用到了 TwitterUtils，那么就需要将 spark-streaming-twitter_2.10 以及它的所有依赖打包到应用程序 jar 中。为 executors 配置足够的内存-因为接收的数据必须存储在内存中，executors 必须配置足够的内存用来保存接收的数据。注意，如果你正在做 10 分钟的窗口操作，系统的内存要至少能保存 10 分钟的数据。所以，应用程序的内存需求依赖于使用它的操作。配置 checkpointing-如果 stream 应用程序需要 checkpointing，然后一个与 Hadoop API 兼容的容错存储目录必须配置为检查点的目录，流应用程序将

checkpoint 信息写入该目录用于错误恢复。配置应用程序 driver 的自动重启-为了自动从 driver 故障中恢复，运行流应用程序的部署设施必须能监控 driver 进程，如果失败了能够重启它。不同的集群管理器，有不同的工具得到该功能

Spark Standalone：一个 Spark 应用程序 driver 可以提交到 Spark 独立集群运行，也就是说 driver 运行在一个 worker 节点上。进一步来看，独立的集群管理器能够被指示用来监控 driver，并且在 driver 失败（或者是由于非零的退出代码如 exit(1)，或者由于运行 driver 的节点的故障）的情况下重启 driver。YARN：YARN 为自动重启应用程序提供了类似的机制。Mesos：Mesos 可以用 Marathon 提供该功能

配置 write ahead logs-在 Spark 1.2 中，为了获得极强的容错保证，我们引入了一个新的实验性的特性-预写日志（write ahead logs）。如果该特性开启，从 receiver 获取的所有数据会将预写日志写入配置的 checkpoint 目录。这可以防止 driver 故障丢失数据，从而保证零数据丢失。这个功能可以通过设置配置参数 spark.streaming.receiver.writeAheadLogs.enable 为 true 来开启。然而，这些较强的语义可能以 receiver 的接收吞吐量为代价。这可以通过并行运行多个 receiver 增加吞吐量来解决。另外，当预写日志开启时，Spark 中的复制数据的功能推荐不用，因为该日志已经存储在了一个副本在存储系统中。可以通过设置输入 DStream 的存储级别为 StorageLevel.MEMORY_AND_DISK_SER 获得该功能。

安装环境介绍（待补充）：本文将 Spark 部署在安装有 ubuntu16.04 系统的 VirtualBox 虚拟机中。

搭建 Spark 集群，需要准备以下文件及环境：jdk-8u211-linux-x64.tar.gz spark-2.4.3-bin-hadoop2.7.tgz 3 个独立的 ubuntu16.04 虚拟机系统，机器集群规划如下：

5.4　流式数据特征抽取

Spark Streaming 支持 Scala、Java 和 Python，本文中使用 Python 提交任务。使用第三章中的特征提取方法，输入为由 kafka 传入的流量数据，提取的结果都是传输层的一些统计信息，以一个 TCP 流或一个 UDP 流为一个单位。TCP 流以 FIN 标志为结束，UDP 以设置的 flowtimeout 时间为限制，超过时间就判为结束。在一个 TCP 流中有很多个数据包，先三次握手而后传输信息再四次挥手。统计一个流中的统计信息作为提取的特征。且统计的特征都分前后向，规定由源地址到目的地址为正向，目的地址到源地址为反向，例如某条 TCP 流的源 ip 地址为 192.168.31.100，源端口号为 174，目的 ip 地址为 183.232.231.174，目的端口号为 443，则为该流构建一个标志叫 Flow ID:192.168.31.100-183.232.231.174-46927-443-6, 由源地址、目的地址、源端口、目的端口、协议号组成。流量特征提取的整体流程图如下：

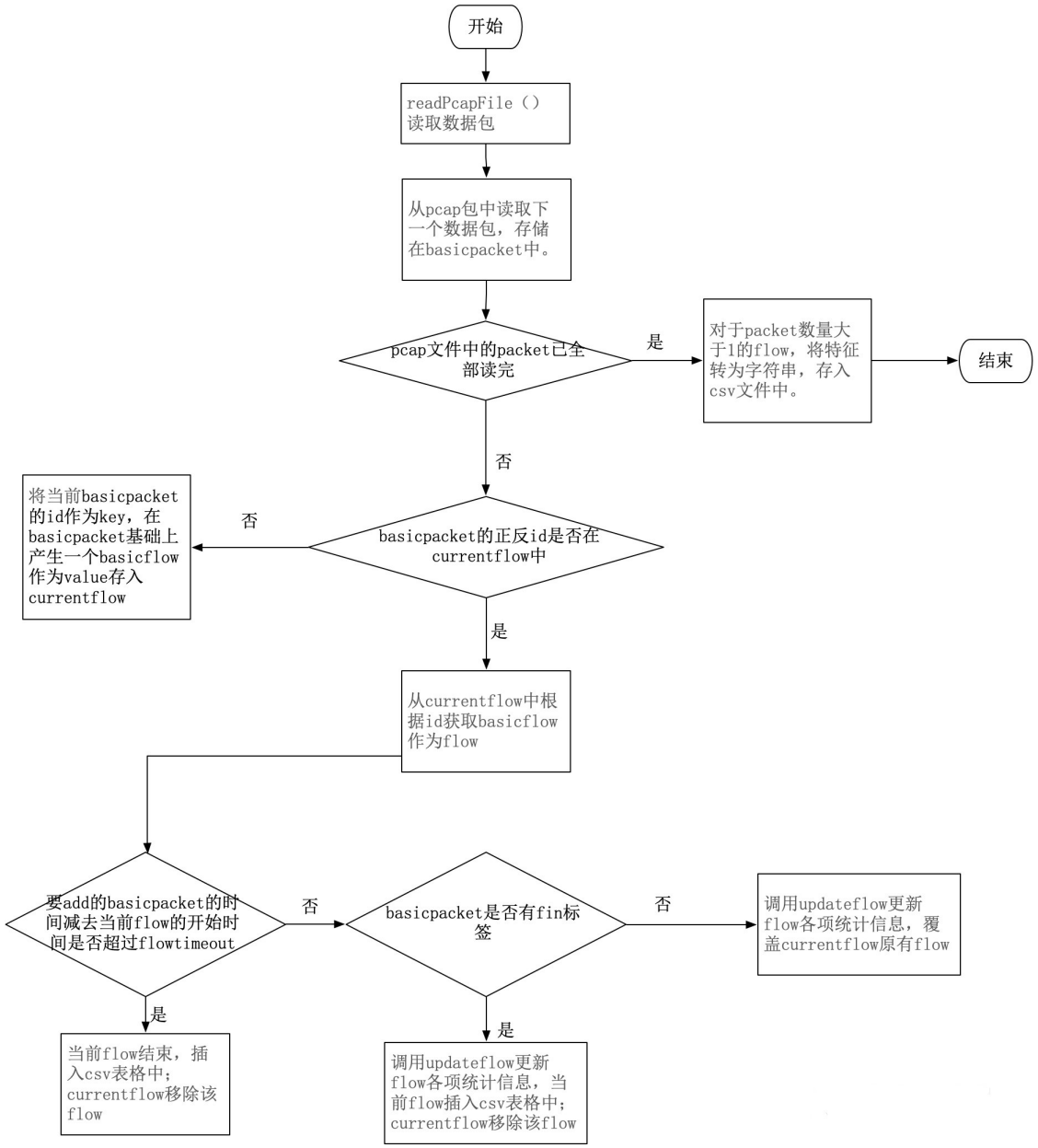

图 5.7　特征提取

从 pcap 文件中逐个读取 packet，将每个数据包添加到对应的流中在 currentFlows 存储当前还未结束得所有 TCP、UDP 流。在添加的过程中不断地更新每个流的统计特征，最终将统计特征写入 csv 文件。判断新加入的数据包是否属于当前所有未结束的流，如果属于当前流则判断正向还是反向，之后判断时间是否超时、不超时则判断是否含有 FIN 标志，如果两者都不满足，则声明一个 BasicFlow 对象，根据 id 从 currentFlows 中拿到与当前数据包对应的流，调用 addPacket 将该数据包加入到对应流中。如果前面判断不在当前所有未结束的流中，则直接创建一个新得流，里面只含当前数据包，存入到 currentFlows 中。如果属于当前某个未结束的流，且超时或存在 FIN 标志，则说明当前 flow 结束，超时则从 currentFlows

中移除对应流，新建 flow 存入 currentFlows 中，含 FIN 标志则直接从 currentFlows 中移除对应流。结束的 flow 直接调用 onFlowGenerated 函数将流打印存储起来。

使用该方法得到的 csv 文件如下所示：

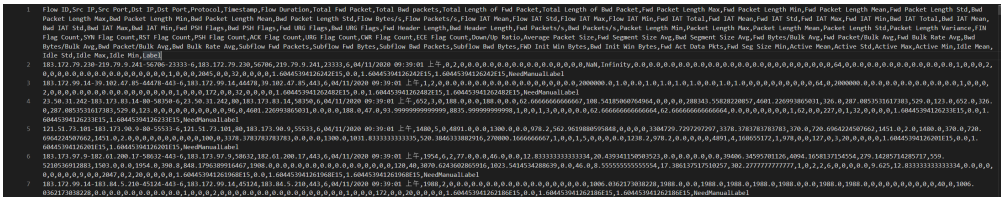

图 5.8　特征文件

5.5　模型训练模块的设计与实现

补充一些图

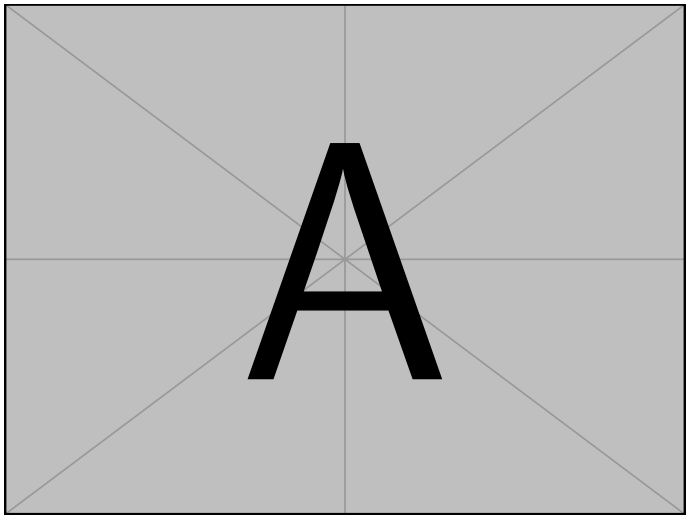

图 5.9　Example figure in appendix

5.6　预测输出模块

5.7　系统结果展示与分析

本章在第二章技术分析和介绍的基础上，先后阐述异常流量检测系统的需求分析、本文系统总体架构的设计、算法模型的设计方案，最后根据系统的总系架构和算法模型提出本文系统五大模块的设计方案。３．１需求分析３．１．１功能需求分析根据项目背景，本文设计的系统功能需求为：（１）能够通过 JnetPcap 技术采集主机流量特征，采集 pcap 类型文件中的保存的流量特征。（２）能够将采集端的采集特征在后台收集端集中，过滤，写入 kafka。（３）能够对流量预测，

本文将流量分成 4 等级，等级越高，是异常流量的可能性越大。（ 4 ）能够对预测的结果进行良好的展示。（ 5 ）预测模型具有一定效果。 3 ． 1 ． 2 系统需求分析（ 1 ）可扩展：随着后期需求的变化与增长，所以在开发过程需要考虑如何应付这些后期变化。针对本系统，需要考虑采集模块支持 Windows、Linux 系统，需要考虑预测模块中模型的更新和替换问题。（ 2 ）健壮性：在流式处理的过程中，有可能其中一个模块偶尔发生错误，不能让这种错误影响到整体系统运行，但是需要对这种错误进行记录。（ 3 ）配置分离：系统的某些属性需要以配置文件的形式分离出来，方便以后对系统的管理控制。 3 ． 2 系统总体设计 3 ． 2 ． 1 异常流量预测系统总体架构设计本系统属于流式系统，采集模块获取数据包流量的特征，并且实时对流量进行预测，最后以报表的形式展示出来。结合需求和相关技术的分析，本文系统总体设计如图所示，下面分别对各个模块的功能进行介绍。

3 ． 2 ． 2 异常流量预测系统运行流程简介在系统环境已经部署好的情况下，结合异常流量的总体架构设计，该系统运行的整体流程如图 3 - 2 所示，图中编号代表这个模块运行的步骤。（ 2 ）开启后台收集模块，收集端应该先于采集端运行，不然会导致采集端发送的数据丢失。（ 3 ）开启采集模块，采集并发送数据。（ 4 ）向 Spark 集群提交预测模块任务，从 kafka 中源源不断的读取数据，并且将预测后的结果写入一个新的 kafka topic。（ 4 ）开启报表模块展示预测结果。

第 6 章　总结与展望

6.1　总结

随着互联网的规模越来越大，网络流量的规模也急剧膨胀，网络应用的种类日益多样化，因此管理网络的难度也越来越大。传统的基于专家知识、人工生成规则的自动化运维具有很多缺陷，一是无法适应快速变化的网络流量环境，过度依赖规则库，而规则库需要不断人为添加新的规则；二是面对复杂网络情况时，很难根据流量的实时变化动态更新基线。

异常检测作为智能运维的关键环节，现有的异常检测算法具有很多缺陷，如面对海量数据规模时，无法或很难做到实时性；应用类型多导致的流量特征复杂，因此很难达到较高的检测率和较低的误报率；大多数异常检测算法仅能报告是否发生异常，难以对确定异常种类和定位异常来源给出指导性意见。

本文以清华大学校园网为例，进行网络流量异常检测算法和系统的研究。清华大学校园网是全球规模最大，架构最复杂，流量场景最多变的校园网之一，具有以下几个特点：

1. 用户规模大，流量峰值高。每天有 10 万台设备活跃在线，同时在线设备最多为 7 万台，峰值流量约为 30Gbps/s.
2. 用户应用类型多，远比一般的企业网复杂，在网络环境中几百种应用同时使用，这给数据分析带来了很多困难。
3. 异常流量是常态。扫描流量、攻击流量占比多。

总的来说，本文在以下几个方面对流量异常检测进行了研究：

1. 本文设计了一种基于特征之间关系图的循环神经网络算法 (FG-RNN)。在复杂的网络流量环境下，流量特征种类繁多，且特征之间的相关性会随着流量变化而变化。原有的异常检测算法通常直接利用提取的特征进行训练，本文提出的 FG-RNN 算法有效利用了特征之间的相关性信息，将其加入到神经网络的训练过程中。经过在开源数据集、真实数据集下分别进行的实验验证，本文提出的算法检测结果优于现有的基于机器学习、深度学习的算法。
2. 本文对基于特征之间关系图的循环神经网络算法进行了流式改造，使之能够满足当前实时异常检测的需求。为了应对海量的校园网流量数据和高速数据流，本文设计了一个基于 spark streaming 的实时异常检测系统。该系统分为输入模块、模型模块、检测模块三部分，输入模块负责将流量数据进行窗口划分、特征选择、特征抽取、合并计算，得到的特征矩阵与预训练得到的关

系矩阵一同送到模型中进行训练，模型中的参数每 2 小时更新一次，最后由检测模块对当前流量进行判别，并给出异常流量的类别。

3. 基于该实时异常检测系统，在公开数据集 UNSW-NB15 和 CICIDS2017 上用多个不同的衡量指标对 FG-RNN 和现有神经网络算法进行实验对比，以 F1-score 为例，相比其他算法的最好效果，

6.2　未来研究和展望

本文通过对清华大学校园网流量进行研究，提出了一种基于特征关系图的循环神经网络算法（FG-RNN），并设计和实现了一个基于 spark streaming 的实时异常检测系统。但是仍然有很多不足之处，本人认为未来的研究可以从以下几个方面着手。

1. 本文 FG-RNN 算法中使用比较简单的基于皮尔森相关系数的关系矩阵计算方式，该方式并不一定是最佳的关系图计算方式。在未来的工作中，可以尝试其他挖掘方法，如 Network Embedding 等。
2. 目前，大多数研究仅关注网络异常的检测和分类，而如何根据检测分类结果定位出异常的源头，指导运维人员进行快速有效地防御，甚至能够根据异常检测的结果自动化防御是未来研究的重点和难点。
3. 由于流量日志数据过于海量，系统给出的异常检测结果往往内容巨大，可能淹没运维人员真正关心的问题。
4. 与商业化的网络运维中心（Network Operation Center）平台中的威胁告警系统相比，基于各种算法的流量检测系统通常人力成本、机器开销更低，但是也有很多不足，未来可以将两者结合。

参考文献

[1] 中国互联网信息中心. 中国互联网络发展状况统计报告 [Z].

[2] Ahmed M, Mahmood A N, Hu J. A survey of network anomaly detection techniques[J]. Journal of Network and Computer Applications, 2016, 60: 19-31.

[3] Hawkins D M. Identification of outliers: volume 11[M]. Springer, 1980.

[4] Eskin E, Arnold A, Prerau M, et al. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data[M]. 2002: 77-101.

[5] Hu W, Liao Y, Vemuri R. Robust support vector machines for anomaly detection in computer security.[C]// 2003: 168-174.

[6] Shon T, Kim Y, Lee C, et al. A machine learning framework for network anomaly detection using svm and ga[C]// Proceedings from the sixth annual IEEE SMC information assurance workshop. IEEE, 2005: 176-183.

[7] Kruegel C, Mutz D, Robertson W, et al. Bayesian event classification for intrusion detection[C]// 19th Annual Computer Security Applications Conference, 2003. Proceedings. IEEE, 2003: 14-23.

[8] Hawkins S, He H, Williams G, et al. Outlier detection using replicator neural networks[J]. 2002.

[9] Zhang Z. Hide : a hierarchical network intrusion detection system using statistical preprocessing and neural network classification[C]// Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 5-6 June, 2001. 2001.

[10] Poojitha G, Kumar K N, Reddy P J. Intrusion detection using artificial neural network[J/OL]. 2010: 1-7. DOI: 10.1109/ICCCNT.2010.5592568.

[11] Cormode G, Thottan M. Algorithms for next generation networks[M]. Springer Science & Business Media, 2010.

[12] Lakhina A, Crovella M, Diot C. Diagnosing network-wide traffic anomalies[J]. ACM SIGCOMM computer communication review, 2004, 34(4): 219-230.

[13] Lakhina A, Papagiannaki K, Crovella M, et al. Structural analysis of network traffic flows[C]// Proceedings of the joint international conference on Measurement and modeling of computer systems. 2004: 61-72.

[14] Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions[J]. ACM SIGCOMM computer communication review, 2005, 35(4): 217-228.

[15] Callegari C, Giordano S, Pagano M, et al. Combining sketches and wavelet analysis for multi time-scale network anomaly detection[J]. Computers & Security, 2011, 30(8): 692-704.

[16] Hamdi M, Boudriga N. Detecting denial-of-service attacks using the wavelet transform[J]. Computer Communications, 2007, 30(16): 3203-3213.

[17] Pascoal C, De Oliveira M R, Valadas R, et al. Robust feature selection and robust pca for internet traffic anomaly detection[C]// 2012 Proceedings Ieee Infocom. IEEE, 2012: 1755-1763.

[18] Fernandes Jr G, Carvalho L F, Rodrigues J J, et al. Network anomaly detection using ip flows with principal component analysis and ant colony optimization[J]. Journal of Network and Computer Applications, 2016, 64: 1-11.

[19] Yeung D S, Jin S, Wang X. Covariance-matrix modeling and detecting various flooding attacks[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2007, 37(2): 157-169.

[20] Xie M, Hu J, Guo S. Segment-based anomaly detection with approximated sample covariance matrix in wireless sensor networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 26(2): 574-583.

[21] Shannon C E. A mathematical theory of communication[J]. The Bell system technical journal, 1948, 27(3): 379-423.

[22] Behal S, Kumar K. Detection of ddos attacks and flash events using novel information theory metrics[J]. Computer Networks, 2017, 116: 96-110.

[23] David J, Thomas C. Ddos attack detection using fast entropy approach on flow-based network traffic[J]. Procedia Computer Science, 2015, 50: 30-36.

[24] Xie M, Hu J, Guo S, et al. Distributed segment-based anomaly detection with kullback–leibler divergence in wireless sensor networks[J]. IEEE Transactions on Information Forensics and Security, 2016, 12(1): 101-110.

[25] Li G, Wang Y. Differential kullback-leibler divergence based anomaly detection scheme in sensor networks[J]. 2012: 966-970.

[26] Han J, Kamber M, Pei J. Cluster analysis: Basic concepts and methods[J]. Data Mining, 2012: 443-495.

[27] Rajasegarar S, Leckie C, Palaniswami M. Hyperspherical cluster based distributed anomaly detection in wireless sensor networks[J]. Journal of Parallel and Distributed Computing, 2014, 74(1): 1833–1847.

[28] Karami A, B G Z. A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks[J]. Neurocomputing, 2015, 149: 1253-1269.

[29] Carvalho L F, Barbon Jr S, de Souza Mendes L, et al. Unsupervised learning clustering and self-organized agents applied to help network management[J]. Expert Systems with Applications, 2016, 54: 29-47.

[30] Dromard J, Roudière G, Owezarski P. Online and scalable unsupervised network anomaly detection method[J]. IEEE Transactions on Network and Service Management, 2016, 14(1): 34-47.

[31] Syarif I, Prugel-Bennett A, Wills G. Unsupervised clustering approach for network anomaly detection[J/OL]. 2012, 293. DOI: 10.1007/978-3-642-30507-8_7.

[32] Moustafa N, Slay J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)[C]// 2015 military communications and information systems conference (MilCIS). IEEE, 2015: 1-6.

[33] Özgür A, Erdem H. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015[J]. PeerJ Preprints, 2016, 4: e1954v1.

[34] spark. spark[Z].

附录 A　补充内容

附录是与论文内容密切相关、但编入正文又影响整篇论文编排的条理和逻辑性的资料，例如某些重要的数据表格、计算程序、统计表等，是论文主体的补充内容，可根据需要设置。

A.1　图表示例

A.1.1　图

附录中的图片示例（图 A.1）。

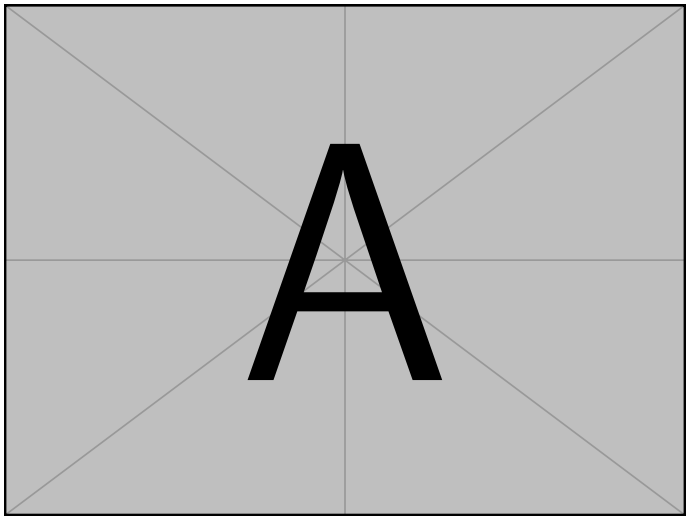

图 A.1　附录中的图片示例

A.1.2　表格

附录中的表格示例（表 A.1）。

A.2　数学公式

附录中的数学公式示例（公式 (A-1)）。

$$\frac{1}{2\pi \mathrm{i}} \int_{\gamma} f = \sum_{k=1}^{m} n(\gamma; a_k) \mathscr{R}(f; a_k) \tag{A-1}$$

表 A.1　附录中的表格示例

文件名	描述
thuthesis.dtx	模板的源文件，包括文档和注释
thuthesis.cls	模板文件
thuthesis-*.bst	BibTeX 参考文献表样式文件
thuthesis-*.bbx	BibLaTeX 参考文献表样式文件
thuthesis-*.cbx	BibLaTeX 引用样式文件

致　谢

衷心感谢导师 ××× 教授和物理系 ×× 副教授对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 Robert Field 教授热心指导与帮助，不胜感激。

感谢 ××××× 实验室主任 ××× 教授，以及实验室全体老师和同窗们学的热情帮助和支持！

本课题承蒙国家自然科学基金资助，特此致谢。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间完成的相关学术成果

个人简历

197× 年 ×× 月 ×× 日出生于四川 ×× 县。

1992 年 9 月考入 ×× 大学化学系 ×× 化学专业，1996 年 7 月本科毕业并获得理学学士学位。

1996 年 9 月免试进入清华大学化学系攻读 ×× 化学博士至今。

在学期间完成的相关学术成果

学术论文：

[1] Yang Y, Ren T L, Zhang L T, et al. Miniature microphone with silicon-based ferroelectric thin films[J]. Integrated Ferroelectrics, 2003, 52:229-235.

[2] 杨轶, 张宁欣, 任天令, 等. 硅基铁电微声学器件中薄膜残余应力的研究 [J]. 中国机械工程, 2005, 16(14):1289-1291.

[3] 杨轶, 张宁欣, 任天令, 等. 集成铁电器件中的关键工艺研究 [J]. 仪器仪表学报, 2003, 24(S4):192-193.

[4] Yang Y, Ren T L, Zhu Y P, et al. PMUTs for handwriting recognition. In press[J]. (已被 Integrated Ferroelectrics 录用)

专利：

[5] 任天令, 杨轶, 朱一平, 等. 硅基铁电微声学传感器畴极化区域控制和电极连接的方法: 中国, CN1602118A[P]. 2005-03-30.

[6] Ren T L, Yang Y, Zhu Y P, et al. Piezoelectric micro acoustic sensor based on ferroelectric materials: USA, No.11/215, 102[P]. (美国发明专利申请号.)

指导小组学术评语

论文提出了……

答辩委员会决议书

论文提出了……

论文取得的主要创新性成果包括：

1. ……

2. ……

3. ……

论文工作表明作者在 ××××× 具有 ××××× 知识，具有 ×××× 能力，论文 ××××，答辩 ××××。

答辩委员会表决，（× 票/一致）同意通过论文答辩，并建议授予 ×××（姓名）×××（门类）学博士/硕士学位。