目 录

插图和附表清单

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 1.1所示，第 46 次《中国互联网络发展状况统计报告》[?] 指出，截至 2020 年 6 月，我国网民规模达到 9.40 亿，互联网普及率达 67.0%，互联网应用涵盖即时通讯、搜索引擎、网络新闻、在线教育、购物出行等方面，可以说互联网已经和每个人的生活息息相关密不可分。

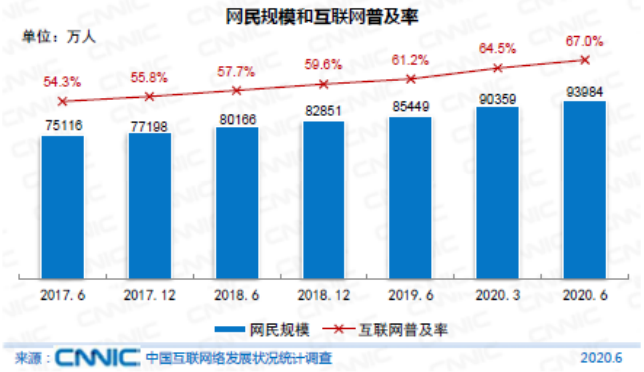

图 1.1　网民规模和互联网普及率

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。网络是一个复杂的系统，它的部署、运行和维护都需要专业的运维人员。早期的运维工作大部分是由运维人员手工完成，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。自动化运维可以被认为是基于专家经验、人为制定规则的系统。但是随着互联网规模急剧膨胀，以及服务类型的多样化，简单的、基于人为制定规则的方法并不能解决大规模运维的问题，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，智能运维强调使用机器学习算法从海量运维数据中不断学习、不断提炼规则。

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。

1.2　研究内容

1.3　论文的主要贡献

1.4　论文内容和组织结构

第一章为引言，主要介绍网络流量异常检测的研究背景，本文的研究内容以及主要贡献。

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。

第三章是基于图结构的 RNN 及其网络流量异常检测算法。

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。[?] 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。

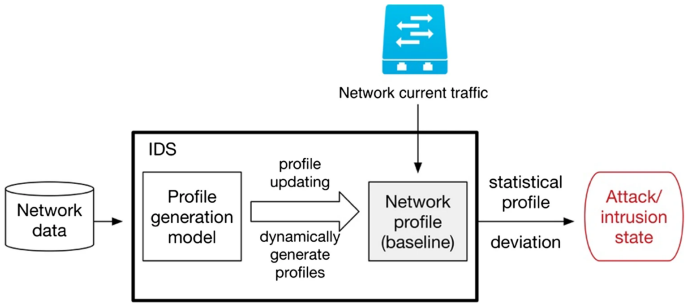

图 2.1　异常检测的通用框架

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[?]：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

?] 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，

?] 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[**?**] 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[**?**] 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,[**?**] 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[?] 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。

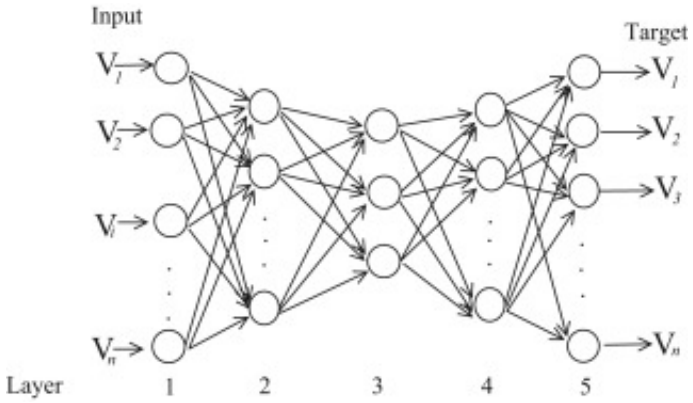

图 2.2　Replicator Neural Networks 示意图

?] 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[?] 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[?]。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[? ? ?]。

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[?] 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[?] 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[?] 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[?] 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[?] 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[?] 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[?] 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[?]。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[?] 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[?] 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[?] 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[?]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[?]。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[?]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[?]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[?] 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[?] 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[?] 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-NB15 数据集是 2015 年澳大利亚网络安全中心使用 IXIA Perfect Storm 工具模拟网络环境流量而生成的一个数据集。相比于 KDD99 是一个更新的数据集，因此更能代表真实的网络流量。UNSW-NB15 数据集中包括 100GB 的.pcap 格式的原始网络流量，同时还有 4 个经过特征提取的 csv 文件，分别是 UNSW-NB15_1.csv 、UNSW-NB15_2.csv 、UNSW-NB15_3.csv 和 UNSW-NB15_4.csv，一共是 2540044 条数据，同时该数据集提出了与 KDD99 较为不同的特征，这些特征更为符合当前的网络协议模式。UNSW-NB15 一共包含有 10 个分类，一个正常类别和 9 个攻击类别，其具体描述和类别数目如下表所示：

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

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 3.1　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 3.2　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 3.3　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 3.4　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 3.5　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
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：（修改成跨页表格）

List of extracted features and descriptions:

Flow duration Duration of the flow in Microsecond

total Fwd Packet Total packets in the forward direction

total Bwd packets Total packets in the backward direction

total Length of Fwd Packet Total size of packet in forward direction

total Length of Bwd Packet Total size of packet in backward direction

Fwd Packet Length Min Minimum size of packet in forward direction

Fwd Packet Length Max Maximum size of packet in forward direction

Fwd Packet Length Mean Mean size of packet in forward direction

Fwd Packet Length Std Standard deviation size of packet in forward direction

Bwd Packet Length Min Minimum size of packet in backward direction

Bwd Packet Length Max Maximum size of packet in backward direction

Bwd Packet Length Mean Mean size of packet in backward direction

Bwd Packet Length Std Standard deviation size of packet in backward direction

Flow Bytes/s Number of flow bytes per second

Flow Packets/s Number of flow packets per second

Flow IAT Mean Mean time between two packets sent in the flow

Flow IAT Std Standard deviation time between two packets sent in the flow

Flow IAT Max Maximum time between two packets sent in the flow

Flow IAT Min Minimum time between two packets sent in the flow

Fwd IAT Min Minimum time between two packets sent in the forward direction

Fwd IAT Max Maximum time between two packets sent in the forward direction

Fwd IAT Mean Mean time between two packets sent in the forward direction

Fwd IAT Std Standard deviation time between two packets sent in the forward direction

Fwd IAT Total Total time between two packets sent in the forward direction

Bwd IAT Min Minimum time between two packets sent in the backward direction

Bwd IAT Max Maximum time between two packets sent in the backward direction

Bwd IAT Mean Mean time between two packets sent in the backward direction

Bwd IAT Std Standard deviation time between two packets sent in the backward direction

Bwd IAT Total Total time between two packets sent in the backward direction

Fwd PSH flags Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)

Bwd PSH Flags Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)

Fwd URG Flags Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)

Bwd URG Flags Number of times the URG flag was set in packets travelling in the

backward direction (0 for UDP)

Fwd Header Length Total bytes used for headers in the forward direction

Bwd Header Length Total bytes used for headers in the backward direction

FWD Packets/s Number of forward packets per second

Bwd Packets/s Number of backward packets per second

Packet Length Min Minimum length of a packet

Packet Length Max Maximum length of a packet

Packet Length Mean Mean length of a packet

Packet Length Std Standard deviation length of a packet

Packet Length Variance Variance length of a packet

FIN Flag Count Number of packets with FIN

SYN Flag Count Number of packets with SYN

RST Flag Count Number of packets with RST

PSH Flag Count Number of packets with PUSH

ACK Flag Count Number of packets with ACK

URG Flag Count Number of packets with URG

CWR Flag Count Number of packets with CWR

ECE Flag Count Number of packets with ECE

down/Up Ratio Download and upload ratio

Average Packet Size Average size of packet

Fwd Segment Size Avg Average size observed in the forward direction

Bwd Segment Size Avg Average number of bytes bulk rate in the backward direction

Fwd Bytes/Bulk Avg Average number of bytes bulk rate in the forward direction

Fwd Packet/Bulk Avg Average number of packets bulk rate in the forward direction

Fwd Bulk Rate Avg Average number of bulk rate in the forward direction

Bwd Bytes/Bulk Avg Average number of bytes bulk rate in the backward direction

Bwd Packet/Bulk Avg Average number of packets bulk rate in the backward direction

Bwd Bulk Rate Avg Average number of bulk rate in the backward direction

Subflow Fwd Packets The average number of packets in a sub flow in the forward direction

Subflow Fwd Bytes The average number of bytes in a sub flow in the forward direction

Subflow Bwd Packets The average number of packets in a sub flow in the backward direction

Subflow Bwd Bytes The average number of bytes in a sub flow in the backward direction

Fwd Init Win bytes The total number of bytes sent in initial window in the forward direction

Bwd Init Win bytes The total number of bytes sent in initial window in the backward direction

Fwd Act Data Pkts Count of packets with at least 1 byte of TCP data payload in the forward direction

Fwd Seg Size Min Minimum segment size observed in the forward direction

Active Min Minimum time a flow was active before becoming idle

Active Mean Mean time a flow was active before becoming idle

Active Max Maximum time a flow was active before becoming idle

Active Std Standard deviation time a flow was active before becoming idle

Idle Min Minimum time a flow was idle before becoming active

Idle Mean Mean time a flow was idle before becoming active

Idle Max Maximum time a flow was idle before becoming active

Idle Std Standard deviation time a flow was idle before becoming active

3.2　校园网真实数据集

第 4 章　基于图结构的 RNN 及其网络流量异常检测算法

4.1　引言

4.2　神经网络

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。

4.2.1　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）。

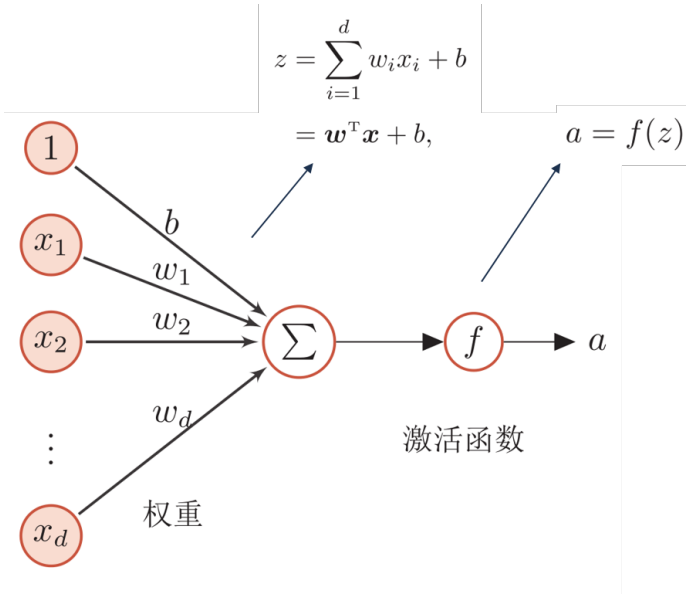

图 4.1　人工神经元模型

- 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。
- 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。

4.2.2　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

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	$f(x)=max(0,x)$	$f(x)=\begin{cases}0 & x<0\\ 1 & x>=0\end{cases}$
Softplus	$f(x)=log(1+e^x)$	$f'(x)=\frac{1}{1+e^x}$
Softmax	$S_i=\frac{e_i}{\sum_j e_j}$	

的网络结构。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

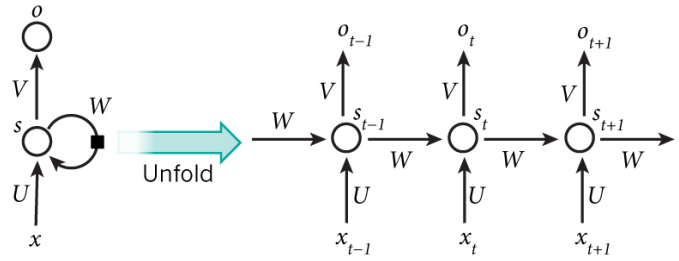

图 4.2　循环神经网络

展开成一个完整的神经网络。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。

4.2.3　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。

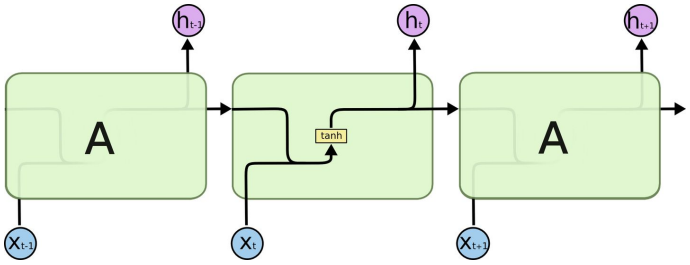

图 4.3　普通 RNN 结构

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

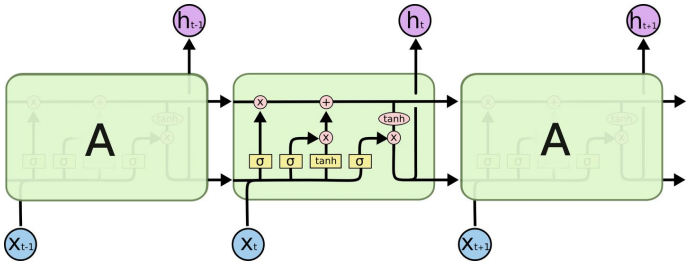

图 4.4　LSTM 结构

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。

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

4.2.4　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，

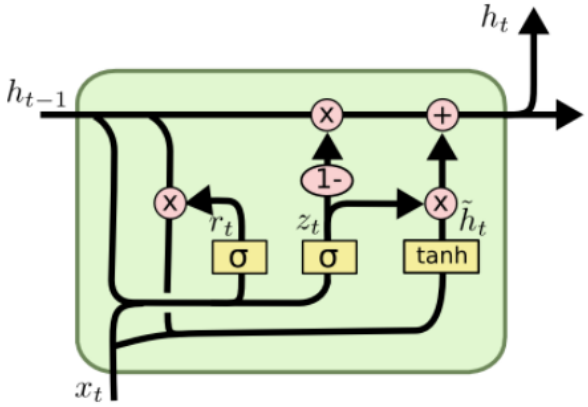

图 4.5　GRU 结构

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

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

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

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

4.3　基于图结构的 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。

表 4.2　部分评估结果

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**

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：

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

4.4　实验方案设计及实验流程

4.5　算法性能评估

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。

4.5.1　基于开源数据集的检测结果

4.5.2　基于真实数据的检测结果

第 5 章　流量检测系统的设计与实现

5.1　引言

5.2　使用大数据平台应对大规模流量的必要性

5.3　基于流式处理的实时检测模块

5.4　系统效率分析

第 6 章　引用文献的标注

模板支持 BibTeX 和 BibLaTeX 两种方式处理参考文献。下文主要介绍 BibTeX 配合 natbib 宏包的主要使用方法。

6.1　顺序编码制

在顺序编码制下，默认的 `\cite` 命令同 `\citep` 一样，序号置于方括号中，引文页码会放在括号外。统一处引用的连续序号会自动用短横线连接。

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）。

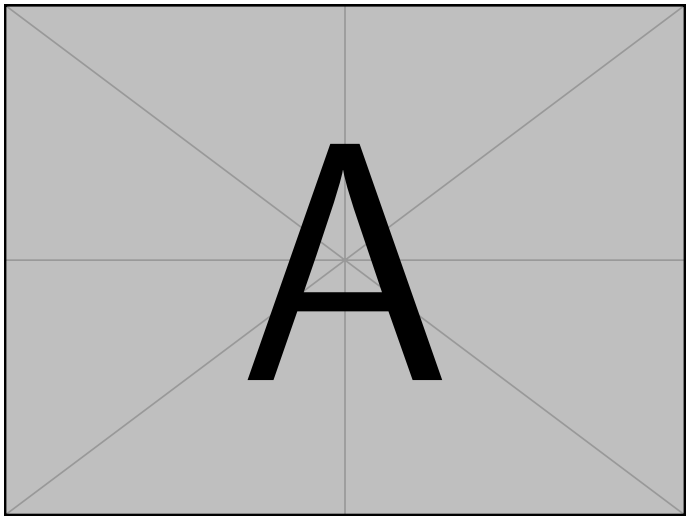

图 A.1　附录中的图片示例

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：_____