独 创 性 声 明

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

研究生签名：杨旭　　时间：2016 年 5 月 25 日

关于论文使用授权的说明

本人完全了解宁夏大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同意宁夏大学可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

(保密的学位论文在解密后应遵守此协议)

研究生签名：杨旭　　时间：2016 年 5 月 25 日

导师签名：[illegible]　　时间：2016 年 5 月 25 日

摘要

网络异常流量分类和检测技术是网络运维管理中重要的技术，因此受到网络安全研究者的广泛关注，并提出了可行的异常流量分类和检测方法。近年来不断恶化的网络安全事件促使着网络异常流量分类和检测技术的不断进步，新的技术不断被提出。

但是随着更大的规模和更复杂拓扑结构的网络不断被建设和使用，传统的基于端口或流特征统计的网络异常流量检测方法无法满足超大流量的数据流的冲击，在时间复杂度上无法满足目前实时检测的需求。网络异常流量检测中主要的时间消耗集中在网络流数据的预处理和规则集的建立中，因此解决大规模网络量级中异常流量分类和检测技术瓶颈的重点就在数据的预处理和规则集的建立中。本文在研究了信息的粒度表示、网络异常流量特征参数提取和大数据技术的基础上，提出了一种基于行为分析的网络异常流量分类和检测方法，该方法在对网络流量进行行为分析的基础上，结合机器学习算法和大数据处理工具，能在保证实时检测的基础上，有效降低检测算法在数据预处理和规则集建立过程中的时间消耗。仿真实验表明，该方法除了在网络异常流量的分类和检测中表现良好之外，还具备聚集未知攻击的能力，可以有效保障网络的平稳正常运行。

关键词：行为分析，异常流量检测，大数据，机器学习

Abstract

Network anomaly traffic classification and detection technology is an important technology in network operation and maintenance management. Therefore, it is widely concerned by network security researchers, and proposes a feasible method for classification and detection of abnormal traffic. In recent years, the continuous deterioration of network security incidents prompted the development of network traffic classification and detection technology, the new technology has been proposed.

With larger scale and more complex topological structure of the network has been continuously construction and use, however, the traditional port based or flow statistical characteristics of network traffic anomaly detection methods can not meet the large flow of data flow impact, in time complex degree, unable to meet the demand of real-time detection. Network traffic anomaly detection in the main time consumption concentrated in the network flow data preprocessing and rules set to establish, thus solving the focus of magnitude in a large-scale network traffic anomaly classification and detection technology bottleneck in data preprocessing and rules set the establishment of. The on the granularity of the information said, network traffic anomaly characteristics parameter extraction and big data technology based on, put forward a kind of based on the data of network traffic anomaly classification and detection method. The method combines the advantages of big data technology and machine learning algorithms, in order to ensure the real time detection based on, effectively reduce the detection algorithm in data preprocessing and rules set established in the course of time consumption. Simulation results show that this method can effectively guarantee the stable and normal operation of the network in addition to the good performance of the classification and detection of abnormal traffic flow and the ability to aggregate unknown attacks.

Key words: behavior analysis, abnormal traffic detection, big data, machine learning

目录

第一章 绪论

1.1 研究背景意义

互联网自诞生起就一直处于不断快速发展的状态中，同时也为人们的工作和生活方式带来了变革，现在各行各业都已经离不开互联网所来带的便利了。新的高速宽带网络不断建设，更复杂的网络拓扑结构不断被规划，更多的企业和个人都有愿望随时随地接入互联网获取最新资讯。随着这样的状态不断延续，企业或个人的网络出口带宽不断拓展、网络流量剧增的同时也带来了新的问题：新的网络安全问题和信息泄露问题频频出现，不断地给企业和个人带来了难以估量的经济损失，成为继续推行网络服务深入的头等障碍。根据国家信息安全漏洞共享平台获得的数据显示：2015 年一年时间内共发现安全漏洞 8043 个，其中高危漏洞达到了 2888 个，占到了整体的 35.91%，这些漏洞很容易被不法分子利用并发动攻击，从而使相关服务无法正常运行甚至会引发严重的经济损失、产生恶劣的社会影响。例如，2015 年 3 月，攻击者利用百度广告联盟的漏洞劫持了其 JS 脚本并将其替换成了恶意代码，其后利用了访问中国网站的国外用户对著名的软件代码托管平台 GitHub 进行了大规模的分布式拒绝服务（DDoS: Distributed Denial of Service）攻击并造成了 GitHub 的瘫痪。由于 GitHub 在软件领域的特殊性，此次攻击事件造成了广泛的关注，百度随后启用了 HTTPS（Hyper Text Transfer Protocol over Secure Socket Layer）来替代更易受到攻击的 HTTP，随后很多知名网站也采取了同样的措施预防类似事件的发生。像这样具有较大影响的网络安全事件几乎每周都有发生。

网络异常检测技术是入侵检测领域的研究热点，网络异常检测问题被定义为[1]：当网络发生异常情况时，其在异常行为下所便显出的行为特征将于网络处于正常状态下所呈现的行为特征不符，网络异常检测的任务就是要寻找这些异常征兆的可能原因[2]。

因此，新的入侵检测技术主见成为研究的热点，研究人员希望在入侵检测技术中找到更好的应对大流量网络环境中异常流量的检测和分类问题。入侵检测系统是当面网络安全防御体系的一个重要组成部分[3]，其基本原理就是汇总收集主机或者网络上的一些关键信息，通过检测其中是否有违反安全策略或者攻击的事件，并且当检测出此类事件时进行警告。目前，主要的入侵检测技术有两种：误用检测和异常检测[4]。误用理论是建立在对已知的攻击行为的建模或者特征描述的基础上的。误用检测是将一种已知的攻击特征或者系统漏洞进行编码存库，然后交给入侵检测系统(intrusion detection system,IDS)对事件进行比对，当匹配成功的时候，则认为有异常行为发生。误用理论的优点是由于利用已有的异常行为知识库进行匹配的，所以其针对性强，误报率低，但是缺点是对未知的异常行为检测率低，并且漏报率较高。异常检测基于建立系统正常工作时的工作模型。当产生入侵行为的时候系统的正常运行模式会发生改变，从而检测出入侵行为。异常检测的优点是有位置检测能力强，但是缺点就是误报率过高。

目前研究的重点主要集中在异常检测技术的研究。该方法由 Denning[5]率先提出，该方法通过检测系统的审计日志数据就能判断出违反规定的危险行为。该方法很快在网络异常检测领域得

到了运用。其中，有指导异常检测(supervised anomaly detection)和无指导异常检测(unsupervised anomaly detection)[6]是网络异常检测领域中常用的两个方法。有指导异常检测要求系统提供完全正常的样本集进行建模，然后通过将检测集与正常样本的建模进行比较得到结论。通常所说的概率统计分析、数据挖掘方法都属于有指导异常检测的范畴。有指导的异常检测理论在处理异常流量的分类问题上有独特的优势，能够较为准确的识别网络攻击流量，但是需要提供要求系统提供一个完全正常的样本在正常的网络条件下基本上是无法满足的，所以，该方法没有普遍适用性；而对于无指导异常检测来说，并不需要提供完全正常的样本集，仅仅需要保证正常流量占大多数，这样的条件更符合实际网络流量的基本情况。因此无指导的异常检测理论适用性较强，能适用于正常的网络中，同时也是现阶段研究的重点。

进行网络异常检测研究的意义十分明显：保障网络的稳定和服务的正常运行、保障网络财产安全、减轻网络管理人员的工作压力。

(1) 保障网络的稳定和服务的正常运行：一个合格的业务承载网络需要 7*24 小时的不间断向用户提供服务，如果因为遭受网络攻击使得正常用户的适用受到影响甚至是无法适用服务，这样的话会严重降低该网络的服务质量(QoS: quality of Service)。而一个良好的异常流量检测和分类平台能有效过滤异常流量信息，从而保证正常的流量不受网络攻击的干扰，因此研究异常流量检测技术对提高网络的服务质量有着极大帮助。

(2) 保障网络财产安全：网络攻击市场与大规模的网页挂马、信息窃取和敲诈勒索挂钩，因此网络攻击常常伴随难以估量的财产损失。所以，入侵检测技术的运用会在一定程度上保障敏感信息的安全，从而降低因为遭受网络攻击所带来的财产损失。

(3) 减轻网络管理人员的压力：一个大型网络的运行维护对网络管理人员的压力是巨大的，如果一项技术能及时发现网络中存在的不安全因素并提醒网络管理人员采取措施应对，就会降低网络管理人员的工作难度。

1.2 国内外研究现状

目前对于网络异常检测相关领域的技术研究工作主要集中在针对网络流量数据处理的问题上。基于不同的网络流量处理的方法，现将该领域的研究方向主要分成四类，即有基于对网络数据的特征进行统计分析的方法、基于对网络数据进行通信层面的信号处理方法、基于对网络数据进行数据的分析挖掘的方法和基于人工智能并通过训练机器学习的方法。

1.2.1 国外研究现状

在基于对网络数据的特征进行统计分析的方法中，Marina Thittan 等[7] 通过对网络流量在管理信息库中的数据项进行分析，并通过标记其中数据项突变的部分进行网络异常情况的判断与标记，提出了一种基于 MIB 变量突变情况来判断网络异常流量的方法；H.Wang 等[8]使用了一种累积和的方法检测 TCP 报文中的控制字段，从而检测网络流量中的 SYN 洪水攻击，他们称这种方法为 CUSUM 方法。可以看出，在基于统计分析的方法中，研究者主要采用的思路还是传统的对

网络数据包的字段的某些属性或者某些字段进行原始的统计分析操作，并依据这种结果对网络的情况进行预测。

在对网络数据进行通信层面的信号处理方法中：首先，美国 S.S.Kim 等[9]通过选取 IP 数据包中一些与目的 IP 有关系的属性，通过对这些属性数据进行小波变换，随后再次进行统计分析，得出了在目的 IP 数据层面上的网络异常流量的特征，并依照这些特征进行异常流量的检测；L.Li 等[10]提出了一种检测网络中分布式拒绝服务攻击的方法，该方法同样通过对网络流量进行小波分析，提取出网络流量中的“能量分布”，并通过这个属性进行 DDoS 攻击；A.Dainotti 等[11]针对小波分析方法中冗余较高的缺点，设计了一种使用连续的小波分析方法，该方法通过连续的小波变换改进了传统方法中使用单一的小波变换导致的冗余过高的问题；P.Barford 等[12]对 IP 数据流和简单网络管理协议中的数据流进行小波分析，证明了这种方法能够很好的表示网络流量的一些细节信息；C.M.Cheng 等[13]通过对网络中的异常传输层数据 TCP 数据进行频谱分析，实验证明该方法能够较好地检测出网络中的拒绝服务攻击。可以看出，在对网络数据进行信号处理的方法中，基于小波分析的方法是经常使用的一种方法。

在对网络数据进行分析挖掘的方法中 L.Kingsly 等[14,15]根据聚类算法不需要给出经过标记的训练集这一特性，创新性的使用了聚类算法对为标记的训练集进行数据挖掘分析，并取得了良好的效果；Y.Yasami 等[15]创新性地将经典的聚类算法 K-Means 算法和经典决策树算法 ID3 算法相结合通过使用决策树算法对聚类结果进行处理并割裂和合并分簇，得到了纯净的分簇来检测网络异常流量；Noble.C 等[16]通过使用网络流数据构建成图数据，并通过检测图中的异常子图的方式判断异常流量；W.Eberle 等[17,18]将深度搜索算法和最短描述长度引入到基于图数据挖掘的异常检测中。综上，在对网络数据进行数据挖掘的方法中，基于聚类算法由于它独有的优势，同样被经常使用。

在基于人工智能构建的机器学些方法中，C.S.Hood 等[19]首先利用了贝叶斯网络的方法，通过管理信息库中信息构建了贝叶斯网络，据此判断网络的异常行为；J.Kline 等[20]在贝叶斯网络的基础上增加了时间序列，对比传统的单一的贝叶斯网络，这种方法提高了贝叶斯网络的准确性；J.Ndong 等[21]通过将信号处理的方法对原始网络流量数据进行处理，然后将处理后的数据带入了一个隐马尔科夫链模型中，使用这种方法实现了对网络异常流流量的检测。

1.2.2 国内研究现状

程光等 [22]通过分析 ICMP 协议中报文的数据，并使用了统计学的理论对网络的行为进行行为判定。梁晟等[23]提出了基于假设检验理论的异常检测方法。

李洋等[6]提出了一种直推式方法的网络异常检测方法，该方法基于改进的置信度机器学习算法(TCM-KNN，transductive confidence machines for K-nearest neighbors)，能够保证检测率的情况下，降低了系统的误报率，并且在一定的噪声干扰下也能够保证其性能。

钱叶魁等[24]提出了一种基于多尺度主成分分析的全网络异常检测方法，提出了基于 MSPCA 的全网络异常检测方法，有效改进了由于单独使用网络流量时间相关性或者空间相关性所导致的检测效率低下的问题。

诸葛建伟等[3]提出了基于 D_S 证据理论的网络异常检测技术，融合多种流量特征对当前网络流量状态进行综合评判，一定程度上降低了误报率和漏报率，并在基于 KDD 99 数据集的检测中达到了 69%的检测率。

穆祥昆等[25] 提出了基于活跃熵的网络异常流量检测方法，该方法利用了 NetFlow 流数据，通过计算活跃熵的方法判断网络流量的异常，并且引入尺度应对不同的流量量级，有效降低了误报率。

颜若愚等[26]提出了一种使用交叉熵检测和分类网络异常流量的方法，该方法同样使用了 NetFlow 流数据，通过使用交叉熵判断流量特征的变化，同时引入了指数加权滑动平均方法检测异常流量，该方法达到了较高的异常流量检出性。

1.3 论文主要内容

本文针对传统方法在应对大规模网络异常流量检测中的不足之处，运用基于大数据的网络异常流量分类和检测技术，主要内容包括：

(1) 论证了网络异常行为与网络流量行为特征参数之间的关系；在分析当前主要的网络流量特征参数提取方法并分析其不足之处的基础上指出了基于粗粒度的网络特征参数提取方法的优势所在；在此基础上提出网络流量特征参数提取方法的具体实施步骤。

(2) 分别分析了 K-Means 数据挖掘算法和 C5.0 决策树算法的原理及实现，并分别介绍基于 K-Means 的网络异常流量分类模型和基于 C5.0 的网络异常流量检测模型的优缺点。

(3) 提出了一种基于流量行为特征的网络异常流量分类和检测方法。通过详细解释该方法的具体实施步骤以及其中包含的算法的具体过程论证该方法的有效性，并通过该方法在具体的 DDoS 攻击中的使用进一步对方法进行验证。

(4) 对论文中提出的算法进行实验验证，通过对实验结果的分析和对比验证算法的有效性。

1.4 论文组织结构

根据上一节所提到的研究内容，将本文分成六个章节。具体安排如下：

(1) 第一章 绪论。主要介绍本文的研究背景及意义、现阶段国内外的研究现状，提出本文的主要研究问题。并梳理文章的研究内容和论文组织结构。

(2) 第二章 网络流量行为特征参数提取方法。首先介绍行为分析与流量特征参数之间的关系；接着分析了现阶段网络中主要攻击的特征；随后介绍了大数据的数据处理方法和基于协议的网络流量分解方法；最后介绍了基于粗粒度的网络流量行为特征参数提取方法。

(3) 第三章 网络异常流量分类和检测模型。分别介绍了基于 K-Means 聚类算法和 C5.0 决策树算法，并分别介绍了基于两种算法的网络异常流量分类和检测模型和优缺点。

(4) 第四章 基于流量行为特征的网络异常流量分类和检测模型。详细介绍了该方法的流程，并分别介绍该方法中所使用到的问题的算法的详解。

(5) 第五章 实验及结果分析。首先介绍试验所使用的软硬件环境和数据集；接着介绍实验的过程、结果及对实验结果的分析。

(6) 第六章 总结与展望。对本文所使用的方法和过程做总结并提出后续的改进过程。

第二章 网络流量行为特征参数提取方法

网络流量数据的量级伴随着网络规模的不断发展而不断增长，并且呈现出愈加快速的变化趋势。人们对网络的依赖性不断增加使得对网络本身和加载在网络上的承载业务的维护工作异常重要。

当稳定运行的网络发生异常行为时，通常会由于该异常自身的特点产生相对应的异常流量，并且这些流量会与正常的流量融合，形成一个一种复杂的流量背景。这种情况带来的结果就是网络中正常的流量行为特征被打乱，也就是说网络中的异常行为会引起网络流量特征参数的改变。网络中异常行为时引起变化的原因，而网络流量行为特征参数的改变是该原因所造成的结果。因此，通常的做法是通过网络流量行为特征参数这个能直观呈现在眼前的属性来判断网络中此时正发生着的异常行为。图 2-1 解释了这种关系：

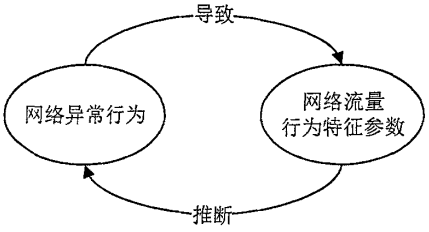


图 2-1 异常行为和网络流量行为特征之间的关系

早期的流量行为特征分析方法普遍采用数据包中的信息进行统计分析，这种方法一般通过分析当前网络中抓取的网络数据包特定字段的信息或者日至服务器中特定字段的日志信息来进行一场流量的检测。这种方法对于异常流量的检测精度比较高，但是随着网络规模的不断增长，这样的方法已经无法满足实时检测的要求。所以为了保证对网络异常流量方法实时性的要求，首先需要研究的就是网络流量行为特征参数的提取方法。

在对行为特征参数提取方法进行研究之前，需要了解该方法具体需要解决的问题：

(1) 解决在大规模数据量运算前提下实时性的需求；

(2) 提取的流量特征行为参数要能比较真实、全面的反映目前网络的实际情况。

2.1 网络中常见的流量异常行为

网络异常流量指得是由于网络扫描、网络攻击或者由于某种原因打破了网络正常运行的格局所导致的网络流数据正常属性的改变[26]。具体表现在网络协议的分布、连接数目、连接的时长、服务类型、源和目的地址等的分布规律打破了正常的限制[5]。由于在实际的网络环境中正常的网络用户和服务所产生的网络流量相对固定，网络管理者通常根据这种固定特征选择适当的网络设备和网络出口带宽，因此当网络中发生异常行为时，所产生的异常流量势必导致正常的网络服务受到影响，从而降低整体的服务质量。所以，在相对固定的网络中及时发现和清理网络异常流量有助于降低网络的安全风险，提高网络的服务质量，并且为网络的后期规划提供依据。

通过对网络行为进行测量和分析,去了解网络运行环境的网络应用和服务的实际工作 状况,从而得到网络的特征[32]。这样的特征可以作为当网络发生异常的时候的一个参考。我们从上网流量、喜好特征、网络异常 3 个方面来分析网络用户的行为特征。

首先,在上网流量方面,在正常的使用中一般会有比较规律的网络流量。尤其是汇总到一个区域网络的总结点时,这样的特征更为明显。同时,可以从上网流量的特征中找出更多的特征值:上网时间、时长、流量大小、网络流向。如果对这样的信息进行采集,可以帮助我们分析整个网络流量的总体状况。

其次,在喜好特征方面,它表现在网络用户对于不同的业务种类的使用情况。在这样的业务类型中,我们可以找到业务使用的时长、次数、用途等特征值。

最后,在网络异常方面,它具体体现在对于网络异常情况的统计中。在网络用户当中通常有以下两种情况:异常产生者,例如垃圾邮件的发送者或者非授权用户的恶意访问等;异常受害者,例如用户资料被盗取等。通过对这样的统计特征的分析,我们可以了解网络的安全状况。下面针对常见的网络异常行为进行详细分析。

2.1.1 网络扫描

网络扫描可以说是网络攻击发起前的准备工作,其具体功能就是利用手头的工具和已掌握的基本信息对特定的网络和区域进行探测,从而掌握探测区域的网络安全漏洞以便后续的网络攻击工作能顺利进行。网络扫描可以一步一步获取到目标主机的网络信息,如 IP 地址、地理位置、内部网络拓扑、同一物理位置包含的其他的站点信息,甚至可以找到扫描目标所在主机的文件信息、电子邮件地址等内容。

网络扫描按照其实施目的的不同可以分为主机扫描和端口扫描两种。主机扫描的目的是探测目标网络区域的活跃主机的信息。活跃主机指的是已经连接在网络中,并且处于运行状态的主机。通常情况下,网络管理人员会通过 ICMP Ping 命令进行网络活跃主机的判断。但是对于准备实施攻击的渗透人员来说,这样的方式显然不够。常见的主机扫描方法有:

Metasploit 的主机发现模块:Metasploit 是一个开源的网络安全漏洞检测工具,集中集成了主机发现的模块,主要有 arp_sweep、udp_sweep 等。其中 arp_sweep 主要依赖 ARP 请求枚举本地网络中所有的活跃主机;udp_sweep 模块利用 UDP 协议的特点发送 UDP 数据包到特定主机检查主机是否活跃,并且可以发现主机上的 UDP 服务。

使用 Nmap 进行主机探测:Nmap 是目前最流行的网络扫描工具,它的特点是不仅可以探测单个目标主机的情况,更可以针对一个 IP 网络端发起网络探测。这样的功能使得 Nmap 在网络探测方面的效率非常高,可以很快探测出网络的活跃主机、开放的服务、甚至是网络中的防火墙。另外,进行主机扫描的同时,还会知道主机操作系统的类型,这对攻击者制定攻击计划是非常有好处的。

除了主机扫描以外,端口扫描技术也是网络扫描的重要分支。端口扫描的目的是判断特定的主机是否打开了特定的端口,并根据端口的特性判断该主机启用的服务类型从而针对特定的服务类型制定攻击计划。因此,端口扫描是进行网络攻击前主要的信息探测方式之一。目前常见的端口扫描技术有:TCP Connect TCP SYN、TCP ACK、TCP FIN 几种。可以看出,这些技术都是使

用了 TCP 协议的一些特性，TCP Connect 是通过建立一次真实的 TCP 连接来判断端口是否被打开，而其他的几种技术是通过发送包含特定标识符的数据包，通过返回的消息判断当前端口的状态，因此前者比较准确但是速度较慢，后几种技术比较方便快捷且比较隐秘，不容易被发现。

类似于主机扫描，Metasploit 和 Nmap 工具里也集成了端口扫描的工具，能够方便地对目标主机端口进行全方位的检测。Nmap 不仅能判断出端口是 open 状态还是 close 状态，甚至当扫描消息可能被防火墙屏蔽也能被侦测到。

综上所述，常见的网络扫描方法都是根据网络协议的特点，选取适合的网络协议对目标主机进行情报收集工作。因此，在防护恶意端口扫描的过程中，就应当关注特定网络扫描的行为特征，并能根据网络数据包和分析总结出来的行为特征参数及时发现恶意事件。

2.1.2 网络攻击

进行完初期的网络扫描工作并发掘有价值的情报之后，攻击者下一步的目标就是筹划网络攻击。目前使用的最多的攻击方式就是分布式拒绝服务攻击(DDoS: Distributed Denial of Service)，因此，下面就 DDoS 攻击做详细分析。从 DDoS 的攻击方式、特征和行为 3 个方面分析其行为特征。

首先，DDoS 攻击通常情况下会有三种表现形式，即利用 TCP 引发的洪水攻击、利用 UDP 协议引发的洪水攻击和 CC(ChallengeCollapsar)攻击。这些攻击模式通常都是通过大量的僵尸主机通过 TCP 或者 UDP 协议的缺陷，发动的耗尽资源性攻击。通过不断地提交无效的操作大量消耗系统的资源，从而达到使服务器拒绝服务的目的[33]。

其次，分布式拒绝服务攻击会造成两个后果，即首先由于攻击者控制庞大的僵尸网络发送巨量的数据包充斥整个网络，会造成网络的拥堵。这样的流量攻击会很快耗尽网络中的出口带宽，从而造成拒绝服务；其次，大量的非法连接请求会占用服务器大量的端口资源、处理资源等，使得正常用户因为分配不到资源而造成拒绝服务。

最后，通过分析 DDOS 的攻击方式和特点，在对分布式拒绝服务攻击的防御措施上，可以提取出诸如传输层端口号，目的 IP 地址等信息。通过对网络中这些信息进行协议分析，就可以建立关于正常网络流量的模型，从而能够识别分布式拒绝服务攻击。

2.1.3 蠕虫病毒

除了特定目标的攻击手段外，通常还有一种可以自动在网络中传播并进行恶意破坏的攻击方式，这就是蠕虫病毒。蠕虫病毒是一种利用互联网或电子邮件系统进行复制、传播，已达到破坏计算机系统、恶意窃取信息等非法的目的。

从攻击方式、特征和行为 3 个方面分析蠕虫攻击的行为特征。

首先，计算机蠕虫可以独立运行，并能把自身的一个包含所有功能的版本传播到另外的计算机上[35]。所以蠕虫可以不借助任何宿主独立地在网络上传播。通常，蠕虫是利用互联网上的漏洞，获得相应的系统权限，对系统进行破坏。并在其传播的过程中产生大量的网络流量导致网络拥堵。

其次，我们可以从其定义和攻击方式来得出其特征：蠕虫会造成网络拥堵降低服务质量。由于蠕虫是一个独立地个体，它在搜索网络和传播的过程中会占用大量的系统资源，并且会向网络中发送大量的数据包，从而会影响系统的性能、耗尽网络带宽，从而造成网络拥堵和服务质量降低[36]。

最后，正是由于蠕虫在传播过程中的特点，服务器或信息系统在感染蠕虫后通常会向周围的网络发送大量的 FIN、ACK、RST 包和 ICMP Ping 包进行网络的搜索并且启用大量的进程完成蠕虫的传播。这是蠕虫最明显的网络行为。通过对服务器上的日志文件和网络抓包的分析，解析出在短时间内是否有大量不正常的 RST 包和 ICMP Ping 包，我们就可以初步推测出系统的安全状况，从而为网络管理员提供较为精确的预警。

2.2 大数据处理原理及工具

区别于传统的以结构化的数据结构存储于数据库中的数据，大数据以一种其数量巨大的非结构化的数据的特征为人们所熟知。在这种大量的非结构化数据中，我们并不一定能保证数据的完整性和正确性。而大数据的魅力正是在如此庞杂但又十分全面的数据中，通过一定的技术手段分析出我们所关心的数据，并通过这样的数据产生价值。由于采用了诸如 Hadoop 这样的分布式架构，大数据的分析效率也比传统的数据仓库分析高很多。因此，大数据能够在更短的时间内为未来的预测提供可靠地数据来源。正是由于这样的特点，大数据才能在新技术迭起的今天成为一个热门的话题。

Map/Reduce 是目前较为流行的分布式并行计算模型，其采用了分治的思想，将一个问题分成若干的块，每一块由一个计算节点分别实现，充分利用了计算集群的优势。并且 Map/Reduce 并行计算模型将一个问题分为两个过程：Map 过程和 Reduce 过程：Map 过程负责将数据块解析成为由(key, value)组成的键值对的组合；Reduce 过程负责处理已经由 Map/Reduce 归纳好的数据，并且将其按照开发者的意图做进一步处理并且输出[37]。整个过程中数据块的划分，任务的分配，数据冗余和容错机制，统统都由 Hadoop 集群负责完成。图 2-2 为 Map/Reduce 的处理过程：

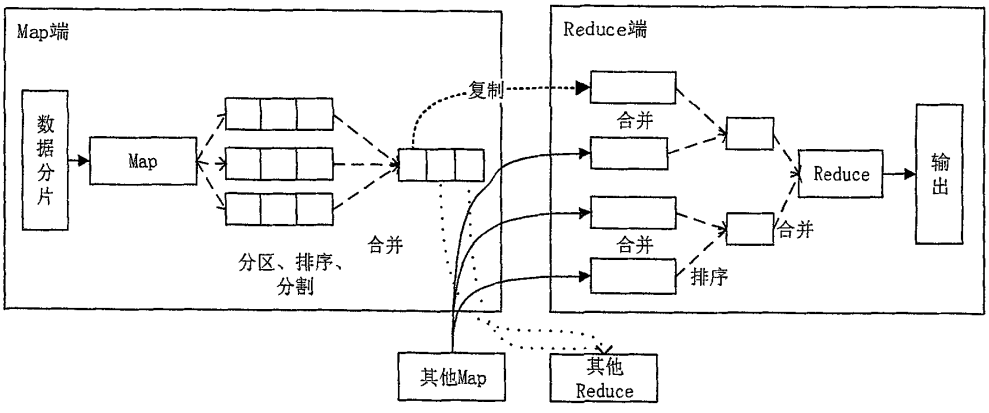


图 2-2 Map/Reduce 任务处理过程

2.3 基于协议的网络流量分解方法

在进行网络流量特征参数提取的过程中，有一个问题需要考虑：常规的方法固然可以表示一般网络中的情况，但是某些异常流量由于其独特的攻击特点，其流量特征往往容易淹没在庞大的流量中，在时间序列中表现不明显。出现这样的问题主要是由于目前的网络攻击主要是以分布式攻击为主，并且往往以来一个特定的网络协议，因此其攻击特征在本协议内的流特征序列内可能表现的比较明显。

为解决上述问题，同时也为了能更清晰地反映现有网络的准确状况，本节提出了一种基于协议的网络流量分解方法，根据协议类型的不同以及传输控制协议(Translation Control Protocol, TCP)标识位的不同取值，将现有的网络流数据分解为不同的子流，这样针对不同的子流进行特征参数的提取有助于找到更多的细节信息，同时对后续的分类和检测方法来说，减小了一个数据维度，从而能减小算法时间消耗，满足实时性的需要。

通信网络中的流量数据主要包括 TCP 流、UDP(User Datagram Protocol)流、ICMP 流以及其他的路由协议流，例如思科的 OSPF(Open Shortest Path First)流和 IGMP(Internet Group Management Protocol)流。其中，TCP 协议和 UDP 协议是传输层协议，网络间的数据都是通过这两种协议进行传输，因此这两种协议的流数据占到了整体网络流数据的很大比重。ICMP 协议主要提供查询和差错控制的功能，由于其协议本身的特点导致了 ICMP 协议很容易被网络攻击者利用并发动攻击，所以在进行网络流量分解时将 ICMP 流单独分解为一个类型。由于网络中路由协议的流数据占比较小，所以在进行网络流量分解的过程中忽略掉路由协议的流数据，仅分析 TCP、UDP、ICMP 三种流。图 2-3 列举出了 TCP、ICMP 和 UDP 协议的报文结构:

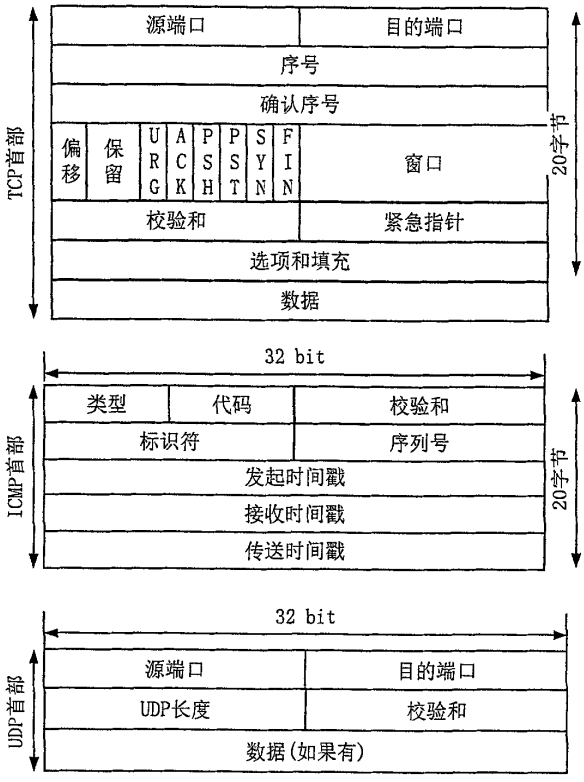


图 2-3 TCP、ICMP 和 UDP 协议的报文结构

由于 TCP 流所占的比重较高，因此基于 TCP 协议的攻击类型也较多。TCP 协议中不同的标识位代表了不同的类型的流量，具体表 2-1 所示：

表 2-1 TCP 各个标识位

标识位	解释
URG	紧急指针
ACK	确认字段
PSH	推送数据
PST	连接时复位
SYN	建立连接时同步
FIN	终止连接

网络中常见的 DDoS 常常是利用 TCP 协议中的漏洞进行攻击，因此又将 TCP 协议根据其标识位分解为 6 个 TCP 子流,具体的分流过程如图 2-4 所示：

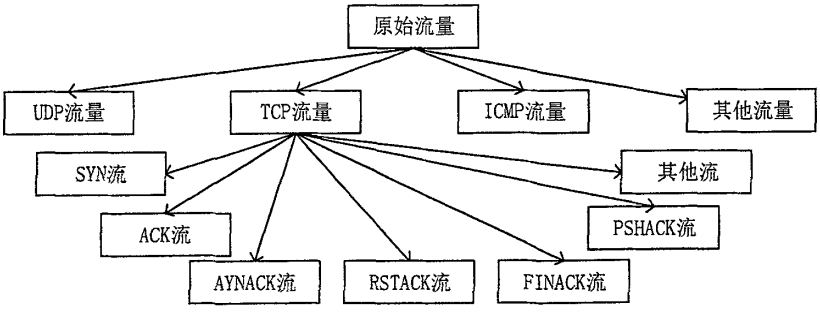


图 2-4 流量分解过程

综上所述，最终的基于协议的网络流量分解方法流程如下：

1. 读入二进制流数据文件，流数据文件来源为 tcpdump 抓取的网络数据包。Tcpdump 是一种 linux 系统中下强大的网络数据采集和分析工具，通过设置抓取的网络接口、抓取的时间或抓取的包的数目等规则生成实时的网络流数据。
2. 读入一条流数据，截取特定字段组成新的流数据，并根据 IP 头中的协议标识位判断该条流数据是 TCP 流、UDP 流、ICMP 流中的一种，随后将该条数据写入特定的文件中。
3. 针对步骤 2 中生成的 TCP 流的文件，进一步判断 TCP 的标识位，根据标识位的不同，进一步将 TCP 流文件分成 6 个 TCP 子流文件，完成基于协议的网络流量分解步骤。

2.4 基于粗粒度的网络流量行为特征参数提取方法

在基于协议的网络流量分解方法的流程中提到，截取特定的字段组成新的流数据有助于简明清晰地表示当前网络流数据的准确状态。那么在网络流数据的众多属性当中，哪些属性能够准确地表示出现有网络的情况呢？本节提出的基于粗粒度的网络流量行为特征参数提取方法，通过对已标记的网络流数据进行粗粒度的分析，找出在众多属性中对于判断网络流量正常与否影响最大的几个属性，并将这些属性设置成为关键属性，也就是网络流量行为特征参数的内容。

粒度(granularity)在传统行业中通常用来表示颗粒的大小。通常意义上球体颗粒使用其直径作为衡量标准，立方体颗粒使用立方体的边长作为衡量标准，不规则颗粒物通常将其等价为某一与其有相同行为的规则物体作进行判断。由此可知，粒度是一种反映颗粒在“尺寸”上的量化分析的标准。粒度被引入信息领域，指的是信息单元的相对大小或粗糙程度。在粗糙集理论中，把人们对于一种知识不同的分类方法称为不同的知识粒度，或者称之为信息粒度。所以，信息粒度可以表示人们对于一种知识的分辨能力，它随着知识的划分程度而增加或减少。

在网络流量的行为特征参数提取方法上，基于细粒度的提取方法主要通过详细解析网络流数据包中的各个字段并逐一提取出每个字段的内容进行分析。例如，将数据包中的源 IP 地址、目标 IP 地址、源端口号、目标端口号、协议号、标识位等等信息逐一分析，这就是一种细粒度的处理方法。而基于粗粒度的提取方法更偏向于关注各种统计数据，例如一种属性的固定值在固定时间段内出现的频率、该属性具有多少种固定的取值等等。

通常情况下，信息粒度的选取直接影响到方法的精度问题和后续处理方法的资源消耗问题。信息的粒度越细，精度越高，同时资源的消耗也就越大；信息的粒度越粗，精度越低，同时资源消耗也会相应减少。但是目前海量的网络数据的背景下，基于细粒度的流量特征行为参数难以满足在这种数据量级下实时性的需求。如何在能保证实时性前提下，使用粗粒度方法尽可能精确的反映目前网络的实际情况，就成了当前研究的重点。

信息熵是一种经典的知识粗粒度表示方法。熵的概念最初用在热学，是一种表示热学概念中分子不规则运动程度的物理量。1948 年，香农经熵的概念引入信息论中并且提出了信息熵的概念，用来衡量信息的不确定性[8]。信息熵的优点在于其不关心离散量的本身，只关心该离散量出现的概率分布。所以，当离散量的概率分布有一个比较明显的改变的时候，信息熵能够以比较小的代价得知这种变化。下面就信息熵的概念介绍如下：

熵是随机变量不确定性的度量，现在设 X 为一个离散型的随机变量[9]，其概率密度函数为 $p(x)$。

定义 1：一个离散型随机变量 X 的熵H(X)定义为：

$$\mathrm{H(X)} = -\sum_{x\in X} p(x)\log p(x) \tag{2-1}$$

其中，对数log所用的底数为 2，并且无特殊说明，一下公式中所有的对数底数都为 2。熵H(X)的单位为比特，表示信息量的多少。而且因此，当$x \to 0$时，约定$0\log 0 = 0$，表示当前熵加上零概率不改变原有的熵值。

同时，X 的熵又可以被解释为随机变量$log(1/p(X))$的数学期望：

$$\mathrm{H(X)} = E_p \, log\frac{1}{p(X)} \tag{2-2}$$

互信息是一个随机变量包含另一个随机变量信息量的度量。互信息也是在给定另一个随机变量知识的条件下，原随机变量不确定度的缩减量。

定义 2：考虑两个随机变量 X 和 Y，它们的联合概率密度函数为$p(x,y)$，其边际概率密度函数分别是$p(x)$和$p(y)$。互信息$I(x:y)$为联合分布与乘积分布$p(x)p(y)$之间的相对熵，即：

$$\mathrm{I(x:y)} = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{2-3}$$

由上述推理可以看出，互信息就是在给定 Y 知识的条件下 X 的不确定度的缩减量。一个属性与结果之间的互信息值的大小反映了给定该属性之后结果不确定的缩减程度，因此，可以使用互信息值大小的排列顺序确定具体的网络流行为特征参数，并将这些参数运用到后续的分类检测步骤中。

因此，基于粗粒度的网络流量行为特征参数提取方法的流程如下：

1 文件输入，其中输入文件中某一条记录的格式为：该条记录中各属性在固定窗口内的信息熵值加上本条记录的标识位共同组成。

2 将输入的文件提交到 Hadoop 分布式文件系统中。

3 通过公式 2-3 中的计算方法，使用 Map/Reduce 分别计算所有记录中各个垂直属性与标识位属性之间的互信息值。

4 根据互信息权重的大小进行排序，并选出前 N 位作为网络流量的行为特征参数，并保存。

2.5 本章小结

本章通过分析了当前网络中导致异常流量发生的主要行为，明确了研究的内容；随后介绍了大数据的原理、方法和主要工具，为大数据量级的网络流数据的处理做足了前期准备。在明确了这些原理的基础上，本章提出了基于协议的网络流量分解方法和基于粗粒度的网络流量行为特征参数提取方法，前者明确了对网络流量按照协议进行细分可以避免大量流量淹没异常流量的缺陷，后者在众多的网络属性中筛选出对网络异常流量行为影响较大的几个特征参数，这些参数在能够准确反映网络目前状态的同时，降低了后续算法的处理维度，满足了实时性的需求。

第三章 网络异常流量分类和检测模型

在对网络异常流量的分类和检测的前人研究成果进行研究与总结的基础上，总结出了常用的网络异常流量分类和检测模型。聚类算法通常被用在网络异常流量的分类工作中，其中以 K-Means 算法的使用最为广泛；而决策树算法通常被使用在网络异常流量的检测工作中。因此，本章中主要针对这两种模型进行分析。

3.1 K-Means 聚类算法

3.1.1 K-Means 聚类算法分析

K-means 聚类算法是一种典型的无监督机器学习算法，由于它在针对大量数据时简单易用的特性，因而被大量使用在具有大规模数据量的场景中。而且由于无监督机器学习的特性，K-means 算法不需要使用已经标记的数据集进行建模， 因此被大量使用在网络流量的分类和检测中。

K-means 算法有时也被称为 K 均值算法，K-means 算法的核心原理是使用平方差作为聚类度量，将集合划分为 K 个簇，使得每一个簇的各个元素之间尽可能相似，但是簇与簇之间的尽可能有较大的差异，最后得到每一个簇的聚类中心。

设现有集合D = $\{x_1, x_2, x_3, \cdots, x_n\}$，其中$x_i$为一个$d$维向量，在给定聚类数目$k(k \leq n)$的前提下，将集合 D 分成$k$个类，算法步骤如下：

(1) 随机给定$\{m_1, m_2, m_3, \cdots, m_k\}$作为算法的初始聚类中心，其中$m_i \in$ D；

(2) 利用就近原则，将 D 中的所有元素划分为k个簇$D_i(1 \leq i \leq k)$，具体方法是使用欧式距离作为远近的衡量标准，计算x_i到每一个m_k之间的欧氏距离，即

$$arg \min_{i \in n} \min_{j \in k} |x_i - m_j|^2 \tag{3-1}$$

(3) 根据步骤(2)中划分的k个簇，分别计算每一个簇的均值，并以该均值为新的簇中心，也就是新的聚类中心点：

$$m_i' = \frac{1}{n_i} \sum_{j \in D_i} x_j \tag{3-2}$$

上式中n_i为新划分的样本D_i中包含的元素个数。

(4) 计算平方差，并根据平方误差衡量算法迭代的终止条件，若平方误差持续减小，那么算法继续迭代，反之则停止迭代并返回已经得到的最新的聚类中心。计算平方误差的公式如下：

$$E = \sum_{i=1}^{k} \sum_{x_j = D_i} (x_j - m_i')^2 \tag{3-3}$$

通常情况下，为了较小计算负担，也通过对比两次迭代所产生的聚类中心的差异作为算法终止的条件，这样做也能保证聚类算法的准确性，同时省去了计算平方误差的计算过程。

下面分析为什么聚类算法会使用这样一种计算方式来达到算法的目的：

(1) 首先，经过初始的选定k个初始聚类中心之后，进行所有的样本属性与k个初始聚类中心计算欧式距离确定该条记录是否需要移动的过程。对于这一步，如果该条记录脱离其原始点并被划分到新的聚类中心去，那么该点新本聚类中心的距离肯定是最小的；若该点没有被指派到新的聚类点，那么说明该点还从属于原来的聚类中心，其欧式距离没有发生变化，即有如下关系：

$$|x - m_i| \leq |x - m_j|, 1 \leq j \leq k \tag{3-4}$$

(2) 对于不断迭代求得最小平方误差的过程，其实就是求得平方误差函数E的极小值的过程，即有如下推导：

$$\begin{cases} \frac{\partial E}{\partial m_1} = -2\sum_{x_j \in D_1} x_j - m_1 \\ \frac{\partial E}{\partial m_2} = -2\sum_{x_j \in D_2} x_j - m_2 \\ \cdots \\ \frac{\partial E}{\partial m_k} = -2\sum_{x_j \in D_k} x_j - m_k \end{cases} \tag{3-5}$$

那么，当$\frac{\partial E}{\partial m_i}$为零，即$\sum_{x_j \in D_i} x_j - m_i = 0$时，$E$取得其极小值，所以有：

$$m_i = \frac{1}{n_i}\sum_{x_j \in D_i} x_j \tag{3-6}$$

因此，k-means 算法的目的就是通过算法迭代的方式，求得 k 个集合的均值点。

3.1.2 基于 K-Means 的异常流量分类模型

由于 K-Means 算法在无监督学习领域的优势，那么该算法在网络异常流量的分类领域中被大量使用。基于 K-Means 算法的网络异常流量分类算法一般步骤是：首先，进行网络流量的采集；随后确定k的值，并开始运行 K-Means 算法；接着根据聚类结果的样本中心进行测试集的分类并输出结果。具体的流程图如图　所示：

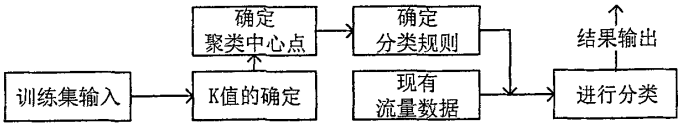


图 3-1 K-Means 算法进行网络流量分类流程

(1) 训练集的采集：网络流量采集是进行网络异常流量分类的前提，网络异常流量的采集工作要求收集到的流量能完美反映当前网络的整体状况。目前网络流量采集的方法基本是基于时间序列进行的，即按照特定的时间间隔对通信网络上的流量进行采集并进行汇总分析。这种方法的好处是可以降低采集设备的工作强度，提高采集速度并能有较高的效率；但是缺点也是十分明显的：由于目前分布式攻击的特点，基于时间序列的网络流数据采集方法很容易漏掉一些网络攻击的行为特征，从而导致异常流量分类结果不理想。

(2) k 值和聚类中心点的确定：k 值和聚类中心点的确定是 k-means 算法最重要的环节，一个好的聚类中心点的集合不仅可以简化算法的运行效率，而且得到的结果也更加准确。首先，在确定 k 的大小的时候一般是按照算法数据集的特点手动设定；确定好 k 值之后就要从数据集中选取 k 个聚类中心点，通过分批次选择距离较远的点的方法：即先随机选择一个点作为第一个聚类中心点，随后依次通过选择距离该点最远的点作为下一个聚类中心点，以此类推最后选出 k 个值。

(3) 运行 k-means 算法并计算最终的聚类中心点：当 k 值和初始聚类中心点确定好之后，将初始数据集和初始聚类中心点的集合作为输入条件，开始运行 k-means 算法。通过数次迭代，知道达到规定的最大迭代次数或者两次的初始聚类中心点的差异达到最小时，停止算法并输出 k 个新的聚类中心。

(4) 对训练集进行分类：将步骤(3)中输出的 k 个新的聚类中心点作为分类标准，设置差异度开始对新的训练集进行分类并输出结果，完成整个分类过程。

由此可见，基于 k-means 的网络异常流量分类方法在基本原理和操作方法上都比较简单，可以有效地对未标记的网络流数据进行分类。但是 k-means 算法在运用到网络异常流量分类时也有其不足之处：

1) 算法比较容易陷入局部最优。因为 k-means 属于梯度下降的算法，所以当算法的初始值选取出现问题的话，就容易导致算法在局部上得到完美的结果但是全局效果不是很理想。因此，解决该问题的根源是找出最优的初始聚类中心的个数和值。
2) 算法结果中簇的纯净度不高。在训练样本差异度较大的情况下，k-means 算法的簇纯净度可以保证；但是当训练样本之间差异度比较小的时候，就比较容易造成一个簇中比较容易参杂很多噪音，最终导致分类时结果的准确率降低。

综上所述，解决上述问题的方法，一个是通过对训练样本进行特征参数的分析确定初始的 k 值；另一个是通过引入新的特征参数或者方法进行簇的提纯工作。

3.2 C5.0 决策树算法

3.2.1 算法介绍

决策树(Decision Tree)是一种树形结构，在其树形结构中，每一个非叶子结点代表的是针对一个属性点上的判断，该结点的孩子结点代表在上述属性点上分类的结果，最后每一个叶子结点代表一个单独的类别。决策树的构造方法决定了决策树进行决策的过程，即从根节点开始，测试选中的属性点并按照分类输出孩子结点，最后通过不断地迭代方式构造出完整的决策树。

不同于其他的分类算法，决策树算法构造的标准不是依赖于类似于贝叶斯算法的领域知识这样较为复杂的知识，而是事物本身的属性度量作为划分标准，将输入划分为不同的类。而前面提到的决策树，就是在进行属性选择过程中加上度量值之后生成的拓扑。

因此，进行决策树构造的关键是如何选择每一步中的决策属性和分裂条件。在决策树中按照一个特定的属性进行划分构造，使划分结果纯度更好的这个特定属性称为分裂属性。分裂属性在具体的划分中有以下几种情况：

1) 若分裂属性为一个随时间变化的连续值，那么该分裂属性在进行操作的过程中应该判断出其分裂点，按照分裂点将该属性分裂为大于和小于该分裂点的两个分支。

2) 若分裂属性为一个离散值并且决策树构造要求只能有两个孩子结点，那么选择其中的一个离散量，将该分裂属性分裂为属于该离散量和不属于该离散量两种。

3). 若分裂属性为一个离散值并且决策树构造没有要求孩子结点的个数，那么按照离散量的多少将该分裂属性进行全部划分。

在确定好分裂属性之后，就需要确定具体的度量标准。目前较为成熟的计算度量标准的算法就是 ID3 算法以及后续的改进算法，包括 C4.5 算法和 C5.0 算法。

3.2.1.1 信息增益

通过第二章的信息熵的介绍可以知道，熵是表示信息多少的概念。而 ID3 算法就是以信息增益这一概念作为度量标准进行属性的划分，选择进行分裂之后属性的信息增益最大的那个属性进行分裂。下面介绍信息增益的概念。

现在设 D 为一个训练元组，则 D 的信息熵可以表示为：

$$info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{3-7}$$

参照信息熵的定义可知，p_i为训练元组 D 中第i个元组出现的概率，D 的信息熵可以表示出 D 中实际蕴含的信息量。

如果按照训练元组中实际属性 A 来进行划分，那么属性 A 对于训练元组 D 的期望信息熵为

$$info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} info(D_j) \tag{3-8}$$

信息增益即为公式 与公式 的差值：

$$gain(A) = info(D) - info_A(D) \tag{3-9}$$

以上就是决策树算法的分裂属性的选择方法。当算法运行时，需要分裂的时候，就分别计算剩余属性的信息增益的值并取最大值对应的属性进行分裂操作。下面通过一个例子演示 ID3 算法决策树构造流程：

我们以一个很典型被引用过多次的训练数据集 D 为例，来说明 ID3 算法如何计算信息增益并选择决策结点，训练集表 3-1 所示所示。

表 3-1 训练集数据列表

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	hot	High	FALSE	No
2	Sunny	hot	High	TRUE	No
3	Overcast	hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	No
7	Overcast	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Overcast	Mild	High	TRUE	Yes

13	Overcast	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

这是一个决策树的经典案例，通过天气(Outlook)、温度(Temperature)、湿度(Humidity)、刮风程度(Windy)这 4 个天气原因属性来决策是否打高尔夫球。从表格中可以看出，给出的属性均为离散量，所以问题就转化为通过若干离散的属性集合判断一个二分类的问题。

按照 ID3 算法构造决策树，首先应该计算决策树型的信息熵，即计算属性 PLAY 的信息熵。样本中 PLAY 属性有两种取值，分别为 Yes(9 次)和 No(5 次)。根据公式 有：

$$info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) = -\frac{9}{14} \times \log_2 \frac{9}{14} - \frac{5}{14} \times \log_2 \frac{5}{14} = 0.94$$

下面进行第一次分裂，通过分别计算 4 个属性的信息增益实现。首先计算各属性对于训练集 D 的期望信息熵：

对于属性天气(Outlook，有晴天(Sunny)、多云(Overcast)、下雨(Rainy)三种取值；属性温度(Temperature)有热(Hot)、中等(Mild)、冷(Cool)三种取值；属性湿度(Humidity)有正常(Normal)、高(high)两种取值；属性是否刮风(Windy)有是(True)、否(False)两种取值。例如，天气属性与结果的对应关系如表 3-2 所示：

表 3-2 不同属性与结果之间关系

序号	OUTLOOK	PLAY
1	Sunny	No
2	Sunny	No
8	Sunny	No
9	Sunny	Yes
11	Sunny	Yes
4	Rainy	Yes
5	Rainy	Yes
6	Rainy	No
10	Rainy	Yes
14	Rainy	No
3	Overcast	Yes
7	Overcast	Yes
12	Overcast	Yes
13	Overcast	Yes

因此，按照上述表格计算属性 Outlook 对于数据集的期望信息熵为：

$$info_{Outlook}(D) = \sum_{j=1}^{3} \frac{|D_j|}{|D|} info(D_j) = \frac{5}{14} \times \left[-\frac{2}{5} \times \log_2 \frac{2}{5} - \frac{3}{5} \times \log_2 \frac{3}{5}\right] + \frac{5}{14} \times \left[-\frac{3}{5} \times \log_2 \frac{3}{5} - \frac{2}{5} \times \log_2 \frac{2}{5}\right] + \frac{4}{14} \times \left[-\frac{4}{4} \times \log_2 \frac{4}{4} - \frac{0}{4} \times \log_2 \frac{0}{4}\right] = 0.694$$

$$info_{Temperature}(D) = \sum_{j=1}^{3} \frac{|D_j|}{|D|} info(D_j) = 0.911$$

$$info_{Humidity}(D) = \sum_{j=1}^{2} \frac{|D_j|}{|D|} info(D_j) = 0.789$$

$$info_{Windy}(D) = \sum_{j=1}^{2} \frac{|D_j|}{|D|} info(D_j) = 0.892$$

下一步，根据公式 3 可以分别计算出各个属性的信息增益，即：

$$gain(Outlook) = info(D) - info_{Outlook}(D) = 0.94 - 0.694 = 0.246$$

$$gain(Temperature) = info(D) - info_{Temperature}(D) = 0.029$$

$$gain(Humidity) = info(D) - info_{Humidity}(D) = 0.151$$

$$gain(Windy) = info(D) - info_{Windy}(D) = 0.048$$

经过比较可以看出，信息增益最大的属性是天气属性(Outlook)，因此第一个分裂属性即为 Outlook 属性。随后针对分裂后的子集进一步进行计算选取新的分裂属性并继续分裂，就能构建出一颗完整的决策树。

3.2.1.2 信息增益率

通过上述例子可以看出，ID3 算法是一种比较好理解的决策树算法。但是 ID3 算法使用的信息增益作为判断分裂属性本身存在着一定的问题，即信息增益总是偏向于拥有较大数量的属性。当一种属性拥有越多不同类型的取值时，它就越有可能被当做分裂属性，而这样产生的划分往往是没有意义的。所以，C4.5 算法在 ID3 算法的基础上做出了改进，主要就是使用了信息增益率的概念替代 ID3 算法中的信息增益。

信息增益率是使用了分裂信息值将训练集中各属性的信息增益进行规范化，即有如下公式：

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|} \tag{3-10}$$

$SplitInfo_A(D)$的值表示将 D 根据属性 A 划分出 V 个输出，并计算出各个划分信息熵的和。那么，信息增益率的定义即为信息增益比上分裂信息的值，即为：

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{3-11}$$

将信息增益率最大的属性选作决策树的分裂属性。

另外，由于决策树在创建时受到噪声和离群点的干扰，部分分枝实际反映的是错误的信息，就需要将这些错误的分枝进行枝剪，除去这些干扰信息之后才能得到最终的决策树。一般进行枝剪操作的方法有两种：先剪枝和后剪枝。先剪枝的方法是通过提前终止某个节点的分裂来停止决策树的构造来实现的，先剪枝的缺点是有可能过早地停止了决策树的生长从而导致决策树的视野效果变差；后剪枝的方法是更常用的方法，它首先将整个决策树构建完成，随后确定需要修剪的树枝并通过使用叶子结点来替换整个分枝实现。

C4.5 采用了一种悲观剪枝法进行决策树的修剪，其思路是：假设对于一个叶子结点，包含了N个样本，其中错误的数目为E个，对于训练机来说该结点的出错率就为N/E，但是在新数据的环境下，需要添加一个惩罚因子对结果进行校正，所以该结点的错误率为$(E+0.5)/N$。拓展到一个分枝而言，其叶子节点的数目为L，因此一个分枝的错误率为$(\sum E_i + 0.5 * L)/\sum N_i$。进行枝剪后，

变成叶子结点之后的错误就为J+0.5。因此，判断该分枝是否需要修剪的依据就是J+0.5 是否在$(\sum E_i + 0.5 * L)/\sum N_i$的标准误差内。

已知一个分枝的样本值为 1，正确分类一个样本值为 0，该树错误分类的概率为e，那么这样的一个分布就为伯努利分布，所以误判次数的均值和标准差分别为：

$$E(subtree\_err\_count) = N \times e \tag{3-12}$$

$$var(subtree\_err\_count) = \sqrt{N \times e \times (1-e)} \tag{3-13}$$

把子树替换成叶子节点后，该叶子的误判次数也是一个伯努利分布，其概率误判率e为(E+0.5)/ N，因此叶子节点的误判次数均值为:

$$E(leaf\_err\_count) = N \times e \tag{3-14}$$

使用训练数据，子树总是比替换为一个叶节点后产生的误差小，但是使用校正后有误差计算方法却并非如此，当子树的误判个数大过对应叶节点的误判个数一个标准差之后，就决定剪枝：

$$E(subtree\_err\_count) + var(subtree_{err_{count}}) > E(leaf\_err\_count) \tag{3-15}$$

C5.0 算法是 c4.5 算法的适用版本，由于 ID3 以及后来的 c4.5 算法在计算过程中需要多次进行扫描、排序等工作，算法效率较低，并且缺乏对大数据集的支持，只能进行常驻内存的数据集的计算。因此在 c4.5 的基础上开发了 c5.0 算法。

C5.0 算法在上述缺陷的基础上，对算法的运行效率和内存做了大幅度优化，使之能应用于大数据集环境中；并且在面对数据集中遗漏数据和输入字段错误的问题上也有很大改进。最重要的是 C5.0 算法采用一种全新的 Boosting 方式提高了模型的准确度，占用内存资源更少，在软件上的计算速度更快。

3.2.2 基于 C5.0 的网络异常流量检测模型

由于 c5.0 算法对于大数据集的支持以及决策树算法在构建分类检测模型方面的优势，使用基于 c5.0 算法的网络异常流量检测模型进行网络异常流量的检测是一个比较好的选择。该模型的基本流程是根据提供的已知的网络异常流量的样本集通过决策树算法生成特定的网络异常流量规则集，通过这些规则集生成网络防护设备常用的 snort 规则库，进行网络异常流量的检测。以下是基于 c5.0 算法的网络异常流量检测模型实现流程：

(1) 网络流数据的读取：读取的数据中包含有流数据正常或异常的标记。

(2) 进行关键特征参数的提取：通常的网络流数据中包含有过多的行为特征参数，而且网络流数据的数据量级一般较大，如果对所有的网络流特征参数进行建模并运算，势必会造成算法的时间消耗过大。因此采用基于粗粒度的网络流量特征参数提取方法进行关键特征参数的提取并进行标记。

(3) 以关键特征参数作为输入，数据集的标记属性作为输出，运行 C5.0 数据模型，输出规则集。

(4) 利用输出的规则集进行 snort 规则库的编写，并将 snort 规则库输入到网络边界设备，进行实时的检测操作。

基于 C5.0 的网络异常流量检测模型的优势是可以根据已经标记的数据集生成较为准确的 snort 规则，并且由于 C5.0 在 C4.5 算法上进行的改进，因此 C5.0 算法对于大数据集的支持很好，

效率可以保障。但是，该方法的劣势也非常明显：该方法需要人工对数据集进行标记，而数据集的标记工作直接影响到算法的最终结果，所以标记的准确性十分重要。但是在实际网络环境中，这样大规模的数据集标记工作是十分困难的。

综上所述，如果能有一个有效地方法能对数据集进行无监督的学习，并且能准确分离网络流量数据中的异常流量信息，那么基于 C5.0 的网络异常流量检测方法模型就能生成较为准确地规则库信息。因此，对网络流数据的标记是 C5.0 能否实用的关键。

3.3 本章小结

本章中在分别介绍 K-Means 聚类算法和 C5.0 决策树算法的基础上，分析了基于 K-Means 算法的网络异常流量分类模型和基于 C5.0 的网络异常流量检测模型。可以看出，如果能有一个有效地方法解决 K-Means 算法的固有缺陷，使得 K-Means 算法得到的聚类结果相对准确，那么 K-Means 的结果就能为 C5.0 算法提供样本标记，这样就可以结合两者的优点设计出网络异常流量分类和检测的方法。

第四章 基于流量行为特征的网络异常流量分类和检测模型

上一章分别介绍了基于 K-Means 算法的网络异常流量分类模型和基于 C5.0 的网络异常流量检测模型。基于 K-Means 算法的网络异常流量分类算法具有簇纯净度不高，且容易陷入局部最优解等缺点，需要进行改进；而基于 C5.0 的网络异常流量检测模型的训练集需要有已标记的异常数据，由于受到输入数据的局限，只能进行已知异常流量的探测，并且标记异常流量工作困难，导致实用性都不高。

并且，当网络规模不断扩大，网络的拓扑结构不断复杂化，一旦针对某一点发动大规模的攻击，网络中瞬间产生的数据流量数值可能会非常恐怖！2014 年针对我国 域名系统的流量规模达 1Gbit/s 以上的拒绝服务攻击事件日均约 187 起，约为 2013 年的 3 倍，攻击目标上至国家顶级域名系统，下至 CDN 服务商的域名解析系统[38]。2014 年 12 月 10 日起，更是发生了全球性的 DNS 流量异常，导致全球大部分的根域名服务器流量暴增已致无法进行正常的域名请求操作。

因此，现今的通信网络中呈现出的这种用户群异常庞大、网络流量异常密集的现象日益严重。尤其是目前智能移动终端的不断接入，更是加剧了检测异常流量的困难，具体表现在：

1. 难以对超大规模的网络数据包做实时分析。针对网络出口中可能超过 100G 每天的流量数据，使用传统的包过滤技术对设备的要求十分高，甚至单独的设备已经无法实时处理这样一个量级的数据，处理时间的延长也导致了针对异常流量的处理工作的滞后。
2. 传统的攻击溯源操作变得非常困难。大量移动智能终端的加入，使得网络的复杂度上升到了一个新的高度，同时也可能使得网络攻击流量很可能会被大量的正常流量淹没，从而无法进行有效地追踪。

针对上述的这些难题，在本章中设计了基于行为特征的网络异常流量分类和检测方法。该方法中首先使用了粗粒度的网络流量分析方法对现有流量数据进行建模，确定异常发生的时间节点，随后使用了 K-Means 算法寻找异常发生的时间节点，通过对簇进行提纯、去离群点操作之后，对 k-means 生成的簇进行 C5.0 进行决策并生成规则集。

该方法解决了 K-means 算法中聚类数无法确定和初始聚类中心选择的问题，同时也解决了 C5.0 算法需要已知训练集标记的问题。由于采用了粗粒度的分析方法，所以算法在应对大规模网络流量时能够有很好的时间复杂度。接下来，本章就该方法的主要流程和算法详细分析两个部分进行阐述。

4.1 异常流量分类和检测方法流程分析

首先，进行该方法的流程分析之前，需要清楚的两个依据：

(1) 在一个正常的网络拓扑中，由于具有固定的服务和用户数目，其流量特征会有一个大体的范围。因此，进行粗粒度的网络流量特征曲线生成过程中窗口的选择就决定着网络流量分析方法的精度问题。窗口越大，生成的曲线精度越低；窗口越小，生成的曲线精度越高。

(2) 依据 K-Means 算法的原理可知，K 的取值越大，最后生成的每一个簇中包含的元素就越少，簇的纯净度也就越高，当 K 的取值与样本数相等的情况下，每个簇只包含一个元素，此时每一个簇的纯净度达到最高，为 100%。因此，在窗口确定的问题中应当考虑到后续 K-Means 算法的特性，适当的 K 值可能会得到纯净的簇，有利于 C5.0 算法进行规则集构建。

基于流量行为特征的网络异常流量分类和检测算法核心思路如下：

1. 从需要分析的网络设备设置端口镜像，通过窗口镜像的方式抓取网络流数据。
2. 根据网络流数据的量级确定检测窗口的大小，并利用选定的窗口进行粗粒度的网络流量特征参数处理工作，标记网络流量异常行为的发生点，并确定针对该异常的初始聚类中心的值。
3. 根据确定的 K 值和初始聚类中心的值，对异常流量的信息熵序列进行聚类，得到 k 个聚类中心。
4. 进行簇内纯净度检查和簇间相似度计算，分离出噪声点并合并具有较高相似度的簇。
5. 通过经过簇号标记的数据集，计算关键流量行为特征参数。
6. 以关键特征行为参数为输入，K-Means 聚类算法的 K 个聚类值为输入，采用 C5.0 生成规则集，完成对异常流量数据的分类和检测工作，算法结束。

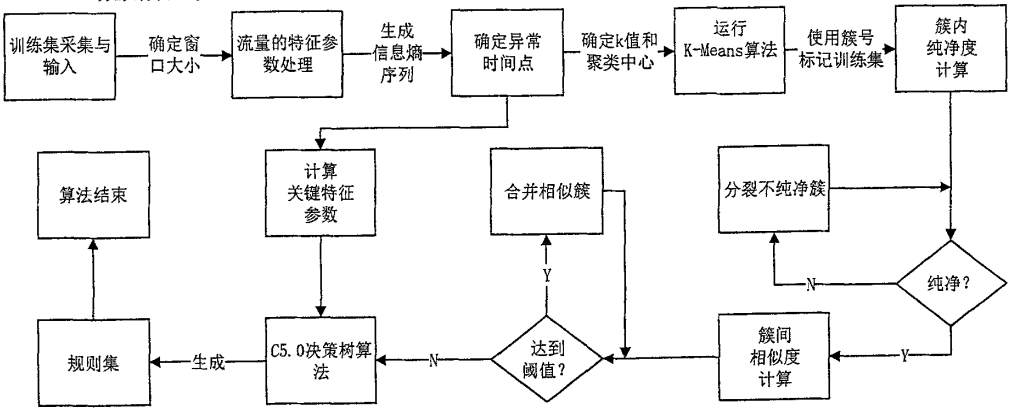


图 4-1 基于流量行为特征的网络异常流量分类和检测算法模型

4.2 算法详细分析

4.2.1 算法窗口大小选择依据

在完成训练集输入之后需要针对网络流数据的特征选择适合现有网络状态下的算法窗口确定。随着窗口的缩小，单个窗口内形成的信息熵值能够更加灵敏的反映出网络流量的一些特征。这一点很重要，虽然目前网络环境中较多情况下出现的是 DDoS 攻击，但是如果窗口过大，很可能丢失流量的细节信息，从而无法检测出基于 U2R 和 R2L 的相关细节。因此，对算法窗口大小选择的理论做出如下证明：

现假设网络流数据的一个属性 M 在集合 D 中存在 n 种不同的元素$\{d_1,d_2,\cdots,d_n\}$，因此，M 在 D 上的信息熵为：

$$H(X) = -\sum_{i=1}^{n}\left(\frac{d_i}{D}\right)\log\left(\frac{d_i}{D}\right) \tag{4-1}$$

而当减小检测窗口的大小的时候，例如将检测窗口调整为$D/2$时，熵值在空间上的变化更加灵敏。现给予证明。设有公式 4-2 和公式 4-3：

$$H_2(X) = \sum_{i=1}^{n/2}\left(\frac{d_i}{D/2}\right) log\left(\frac{d_i}{D/2}\right) \tag{4-2}$$

$$H_3(X) = \sum_{i=1}^{n/2}\left(\frac{d_i}{D/2}\right) log\left(\frac{d_i}{D/2}\right) \tag{4-3}$$

那么，$H(X)$的值可作出如下推导：

$$\begin{aligned} H(X) &= -\sum_{i=1}^{n}\left(\frac{d_i}{D}\right) log\left(\frac{d_i}{D}\right) \\ &= -\sum_{i=1}^{n}\frac{1}{2}\left(\frac{d_i}{D/2}\right)\left(log\, d_i - log\frac{D}{2} - 1\right) \\ &= -\frac{1}{2}\sum_{i=1}^{n}\left(\frac{d_i}{D/2}\right)\log\left(\frac{d_i}{D/2}\right) + \sum_{i=1}^{n}\frac{d_i}{D} \\ &= \frac{1}{2}\left(H_2(X) + H_3(X)\right) + \sum_{i=1}^{n}\frac{d_i}{D} \\ &= \frac{1}{2}\left(H_2(X) + H_3(X)\right) + 1 \end{aligned}$$

因此，在其他条件不变的情况下，减小检测窗口的大小会使检测精度发生明显的提高，但同时会造成检测算法的时效性降低，并且会带来更高的误报率。

4.2.2 关键流量行为特征参数的提取算法

关键流量特征行为参数的提取可以对数据集进行降维，从而简化 C5.0 算法的时间复杂度。而关键流量特征行为参数的选择依据是计算各属性与结果属性之间的互信息值，因此互信息权重计算是体现本算法精度的关键所在，同时也是时间消耗最大的部分。下面就互信息权重的理论和算法做简单介绍：

将网络流数据构成的集合记做$Flow$，由N条记录组成，每条记录都包含由d+1 个属性值构成的序列$V_i(0< i <d+1)$，其中$\forall x_i \in Flow(0<i<N)$表示一个$d$+1 维向量，$x_i^j(0<j<d)$表示向量$x_i$中的每一条属性，$x_i^{d+1}$为该条数据的标识位$Flag$，表示了网络流数据的正常与异常。

对于$\forall x_i$，$x_i^j(0<j<d)$与x_i^{d+1}之间的互信息$I(x_i^j;\ x_i^{d+1})$表示在给定了属性$V_j(0<j<d)$之后，V_{d+1}不确定度的减小量。因此，$I(x_i^j;\ x_i^{d+1})$的值代表了属性序列中的任意属性在判断网络流量中权值的大小。因此，就整个$Flow$而言，$V_i(0< i <d)$与V_{d+1}的互信息$I(V_i; V_{d+1})$为：

$$\begin{aligned} I(V_i; V_{d+1}) &= H(V_i) - H(V_i|V_{d+1}) \\ &= -\sum_{j\in N_1} p(V_i = j) log p(V_i = j) \\ &\quad + \sum_{j\in N_2} p(V_{d+1} = j) H(V_i|V_{d+1} = j) \end{aligned} \tag{4-4}$$

其中：

$$H(V_i|V_{d+1}=j)=\sum_{m\in N_1}p(V_i=m|V_{d+1}=j)logp(V_i=m|V_{d+1}=j) \quad (4\text{-}5)$$

基于上述理论，设计了基于 Map/Reduce 的计算互信息权重值的算法 CIEWBMR(count information entropy based on Map/Reduce),包括 Map 方法，Reduce 方法，Driver 方法。

算法的计算流程如下：

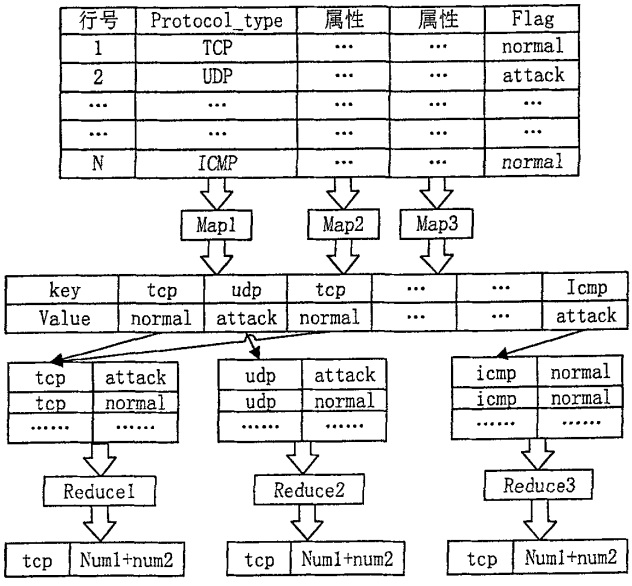


图 4-2 CIEWBMR 算法计算流程

输入：经过整理的网络流数据

输出：<key,value> //key 表示V_i中不同的取值，value 表示 key 值的信息熵。

(1) Map 方法从网络流数据中逐行读取，以截取字符串的形式分离出特定的属性值V_i，并且以V_i为 key，标识位为 value，生成<V_i，V_{d+1}>键值对。

(2) Reduce 方法取得 1 中的键值对，并统计每一个 key 在不同$Flag$上的数量，计算每一个键在不同标识位上的数量，并将各部分的数量以 $s_1+s_2+\ldots+s_n$ 的方式封装于 value 中，并输出<V_i,$s_1+s_2+\ldots+s_n$>键值对。

(3) 重复 1，2 步骤，输出所有V_i在不同$Flag$下的数量。

(4) 输入<V_i,$s_1+s_2+\ldots+s_n$>键值对，统计每个V_i和$Flag$的互信息值，输出每一个V_i的信息熵和互信息权重值。

4.2.3 簇内纯净度计算

经过 K-Means 算法计算之后的分簇在数值的统计分析特征上有一定的相似性。具体表现为簇间的各个元组的标准差在一定的范围之内。在经过 C5.0 决策树算法进行划分之后，那么如何判定一个簇是否纯净？也就是说，如何判定一个簇中是否只包含一种网络流量特征参数？这就需要进入簇纯净度的定义。

设需要判定的簇为C_k，$D_m=\{D_{m1}, D_{m2}, \cdots, D_{mn}\}$为其中的一条数据。$n$表示该条数据的$n$个属性。簇纯净度的定义如下：

定义 4.1 将簇C_k中的所有元素按照元素的时间或空间序列的连续性分割成若干个子簇，若对于各个子簇而言其标准差非零的属性n的取值相同或者属性n在以δ为窗口构建的熵值序列上无明显变化，那么该簇就是纯净的。

判定簇纯净度的目的是为了在一个簇中分离出包含有不同的流量特征参数的子簇，从而使得每一个单独的簇都是纯净的。

算法流程如下：

输入：K-Means 算法输出的簇C_k

输出：纯净的子簇C_{k1}，C_{k2}，C_{k3}，…，C_{km}。

(1) 将簇C_k按照序列号划分成若干子簇C_{k1}，C_{k2}，C_{k3}，…，C_{km}；

(2) 选择δ值的大小以适应子簇大小；

(3) 对于每一个子簇C_{k1}，C_{k2}，C_{k3}，…，C_{km}，选择标准差非零的特征属性 n，以δ值为窗口计算属性 n 的信息熵；

(4) 对比在属性 n 下各子簇的信息熵值，若熵值相同或在阈值范围内，则该簇是纯净的，若信息熵值超出阈值范围，则对簇进行分裂，并且设置 k=k+1；

(5) 对分裂之后的簇使用 k 值重新进行标记；

(6) 不断变化C_k中k的取值，逐次执行上述操作，最后使每一个簇都是纯净的，结束算法。

经过簇内纯净度的计算和分离之后，保证了每一个簇都已经是纯净的子簇。也就是说纯净子簇之内包含的元素具有相同的特征向量，在该簇所对应的网络流量数据包内有着相同的行为特征，这样就可以判定这些行为特征所对应的具体网络行为的唯一性。

综上所述，经过上述步骤之后，保证了针对具体行为建立特征向量的准确性，排出了由于 K-Means 算法中可能出现的离群点的干扰。

4.2.4 簇间相似度计算

基于流量行为特征的网络异常流量分类和检测算法中由于每一个异常的时间点内都会生成一个初始的聚类中心以便将本异常时间内的元组进行聚类操作。但是假设两次异常事件是由同一种网络攻击方式造成的，那么这两种攻击方式就会表现出相似的流量行为特征参数。单纯的进行聚类操作可能会造成聚类结果的两个簇之间有较大的相似度。因此，需要对聚类生成的簇进行簇间的相似度计算并对相似的簇进行融合的操作。

定义 1：对于一个聚类中心的簇C_j的所有点X_i，若任意一点到其聚类中心的距离大于常数 c，那么定义该点为离群点，其中，常数 c 为：

$$\frac{a*\sum_{d=1}^{dim}\sum_{i=1}^{n_j}(X_{id}-C_{jd})^2}{n_j} \tag{4-6}$$

其中 a 为可调的系数；dim 为空间的维度，具体取值为聚类中选取的属性个数，n_j为该簇中包含的样本的数目。

定义 2：设簇C_i，C_j为经过聚类生成的两个簇，$D_i=\{D_{i1}, D_{i2}, \cdots, D_{im}\}$和$D_j=\{D_{j1}, D_{j2}, \cdots, D_{jn}\}$分别为$C_i$，$C_j$中的一个元组。簇$C_i$，$C_j$的相似度为下式：

$$sim = \frac{n_{ij}+n_{ji}}{n_i+n_j} \tag{4-7}$$

上式中，n_i和n_j分别代表簇C_i，C_j中包含的样本数目，其中n_{ij}表示簇C_i中距离簇C_j的聚类中心小于C_j的半径的数目；同样，n_{ji}表示对应的意义。所以，簇间相似度即代表所判定的两个簇之间相互包含对方数据的百分比。

簇间相似度算法流程如下：

输入：经过簇内纯净度检查的所有簇C_m

输出：簇内纯净、簇间无相似的元素的簇集合C_n

(1) 从C_m中挑选C_i、C_j，计算簇C_i的簇半径为R_i，簇C_j的簇半径为R_j；

(2) 根据公式 4-7 计算簇C_i，C_j间的相似度；

(3) 相似度若大于阈值，则C_i，C_j合并成为一个簇C_i并删除簇号C_j；

(4) 重新计算C_i的聚类中心点和簇半径；

(5) 重复步骤(3)和步骤(4)，直到所有的簇间相似度都小于阈值，输出C_n。

经过簇间相似度的计算以及相似簇的合并之后，保证了所有簇间的相似度都小于一个固定的阈值。也就是说不同的簇中所包含的流量行为特征向量能唯一表征一种流量特征行为。

4.3 基于流特征参数的 DDoS 攻击分类和检测方法

根据本章提出的基于流量行为特征参数的网络异常流量分类和检测方法，本节通过分析一个具体的网络异常问题：DDoS 攻击，详述该方法的流程。下面，详细说明各步骤在具体检测 DDoS 攻击中的应用。

(1) 抓取网络流数据。

从具体网络设备中取得的数据为 NetFlow 数据或者 Tcpdump 格式的流量数据。但是这些数据都是以二进制存储的，因此，需要参照 NetFlow 或者 Tcpdump 的数据结构，对流量特征值进行提取。针对 DDoS 提取到的是数据包五元组数据(源 IP、目的 IP、源端口、目的端口、协议)以及流数据的大小、源和目的自治域号等参数。

(2) 确定异常点的位置以及确定初始聚类中心

当网络中遭受 DDoS 时，在遭受攻击的时间段内，网络中充斥大量特定大小的流数据、并且源 IP 和目的 IP 地址的数量急剧增多，在信息熵理论中，数据在一个时间点上越集中，该点的信息熵的值就越大；数据在该点内越分散信息熵的值越小。因此，可以根据信息熵值的急剧变化对攻击动作进行标记。

所以，选择目的 IP 地址熵值的急剧变化作为发生 DDoS 的标志，所以首先建立基于特定检测窗口目的 IP 地址的信息熵的值。接下来就根据时间序列进行推进并检测是否为异常行为发生

的点，若检测到发生目的 IP 地址的信息熵序列有较多下降的点，将该点标记为 DDoS 行为发生的点，否则就继续向下推进，直至训练集检查结束。算法进入到下一步。流程如下：

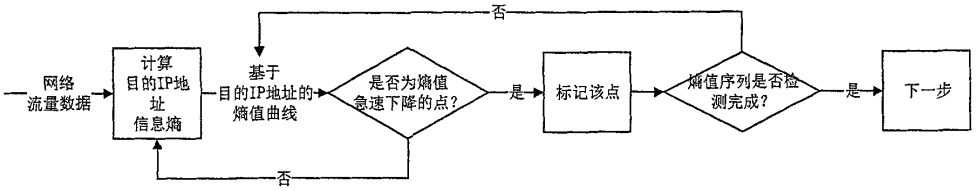


图 4-3 网络流量数据异常点标记流程

为了进行不同类别的 DDoS 攻击的分类，就需要在每一个发生 DDoS 攻击的时间序列内找出一个样本值，并以此样本值作为初始的聚类中心

当训练集标记完成之后，分别在正常的熵值序列内和异常发生的熵值序列内寻找聚类中心点，最后得到一个正常序列的初始聚类中心和多个异常的聚类中心以及 K 值的大小。在进行 DDoS 攻击的分类和检测中，进行 K-Means 聚类的目的是为了检测不同的 DDoS 攻击类型。

(3) 进行聚类操作，并进行簇内纯净度和簇间相似度检查

根据第二步中确定的聚类中心对关于异常点时间序列内目的 IP 的熵值集合进行 K-Means 聚类操作，并最终使得算法收敛，得到 K 个关于目的 IP 地址的簇。

首先对簇进行纯净度检查，使用簇内纯净度检查算法流程过滤离群点；接着进行簇间相似度检查，使用簇间相似度检查算法，合并相似度较高的簇。

随后，使用簇号进行输入数据集的标记，得到带有簇号标记的流量特征行为参数。

(4) 关键流量特征行为参数的提取。

以步骤(3)中的数据集为输入，运行 CIEWBMR 算法，并取算法结果的前 5 位作为关键流量特征行为参数。

(5) C5.0 算法生成规则集

经过前面步骤的操作，得到了 C5.0 算法的输入：5 个关键流量特征行为参数的信息熵值、带有簇号标记的数据集。至此，判定网络异常流量行为的规则集产生，算法结束。

4.4 本章总结

本章详细介绍了基于流量行为特征的网络异常流量分类和检测模型，通过对模型的框架、实施流程、其中的详细算法进行介绍，建立了进行网络异常流量分类和检测具体实施过程。通过对基于流特征参数的 DDoS 攻击的分类和检测方法进行具体研究，补充了基于流量行为特征的网络异常流量分类和检测模型中的一些细节问题。

第五章 网络异常流量分类和检测方法验证与分析

针对提出的基于流量行为特征的网络异常流量分类和检测算法，在本章设计了一系列的实验环境验证算法的可用性和准确性。

5.1 实验基本情况介绍

5.1.1 实验数据集

本实验主要选取了 KDD 99 数据集作为分析对象进行算法和方法的验证。KDD 是数据挖掘与知识发现(Data Mining and Knowledge Discovery)的简称。国际知识发现和数据挖掘竞赛每年都会提出一个在该年度的一个热门问题，并且收集数据组成数据集。这些数据都是真实的，仅仅经过最初的数据汇总工作形成的。例如，2015 年的 KDD CUP 题目就是预测在慕课课堂中的辍学比例(Predicting dropouts in MOOC)。其数据集就来自于清华大学创建的慕课平台：学堂在线。数据集包含了每个学生在学堂在线中的信息，并且要求参赛者根据这些消息预测其在未来的 10 天内会不会放弃一个课程。

KDD 99 数据集是 KDD CUP 1999 年的比赛用数据集。本次比赛的任务是建立一个网络入侵检测的模型，区分数据集中“好”的连接和“坏”的连接，将这些连接分为正常连接和异常连接，其中异常连接包含正在遭受攻击的连接和进行入侵探测的连接。整个数据集的收集过程是在美国国防部高级规划署(DARPA)1998 年的一个入侵检测评估项目上的基础上进行的。为了这个项目，军方在麻省理工学院的林肯实验室中搭建了一个模拟网络环境用来模拟军队的局域网络。在这个网络的基础上，林肯实验室搜集了这个模拟网络中九周的网络连接数据和审计数据。其中参杂了各种类型的网络流量和各种不同类型的攻击手段。

KDD 数据当中的网络连接记录被定义为在一定时间内的 TCP 数据包序列，并且在该时间段内，数据在预定义的协议下从源地址到目的地址的传递。并且，训练数据集的网络连接记录的末位还带有网络连接状态的标志位，分为正常(normal)和异常(attack)。

KDD 数据集包含了一个约 5,000,000 条的训练数据集和大约 2,000,000 条的测试数据集。数据集中包含了四种类型的属性特征：TCP 连接的基本特征、TCP 连接的内容特征、基于时间的网络流量统计特征和基于主机的网络流量统计特征。同时，数据集中的异常类型也分为 4 种：拒绝服务攻击(Dos)、来自远程主机的未授权访问(R2L)、未授权的本地超级用户特权访问(U2R)和端口见识或扫描(PROBING)。KDD 数据集的样本类别所占比例如表 5-1 所示：

表 5-1 KDD 数据集样本比例分配

	Normal	Dos	U2R	R2L	PROBE
训练集	0.1969	0.7924	0.0001	0.0022	0.0083
测试集	0.1975	0.7490	0.0007	0.0528	0.0136

5.1.2 实验环境

进行实验之前，首先介绍本实验所使用的软硬件环境以及各个环境之间的配置。实验环境包括大数据处理模块和数据建模模块两个部分的内容。

(1) 实验硬件环境

首先，在大数据处理模块的硬件需求部分，使用了两台 Linux 系统的主机进行部署。其中一台 Centos 7 系统的主机作为 Hadoop 分布式计算集群的名称结点和数据接点，另一台 Centos 主机作为数据结点、第二名称结点和 MySql 数据库服务结点。部署机器。

其次，数据建模部分使用了 Windows 10 专业版作硬件支持，并且安装数据挖掘软件，进行 K-Means 聚类数据挖掘和 C5.0 决策树算法规则集计算使用。整个实验环境的拓扑如图 5-1 所示。

(2) 实验软件环境

在两台 Linux 主机上首先部署了 Hadoop 大数据处理平台：采用了 Hadoop2.5.2 版本进行搭建。Hadoop 从 0.23.X 和 2.X 版本以后启用了新的 Yarn MapReduce 架构，主要解决了 0.20.X 和 1.X 版本中单一的 JobTracker 由于任务繁重、资源急速消耗而带来的集群数目无法突破的问题。图 5-2 中提供了新的 Yarn 架构的架构图。

跟以前的 Hadoop 版本对比可以发现，JobTracker 和 TaskTracker 被 ResourceManager、ApplicationMaster 和 NodeManager 三个部分替换。形成了一个新的架构。ResourceManager 主要负责进行资源调度并实时启动新的作业的 ApplicationMaster 并同时监视其状态。ApplicationMaster 主要负责一个作业生命周期的一切事物，每一个新申请的作业都会有一个新的 ApplicationMaster 对该作业进行管理。所以 ApplicationMaster 基本完成了旧框架中 JobTracker 中的任务。框架中的 NodeManager 主要负责对 Container 的状态进行维护，并向 ResourceManager 保持心跳机制以确定该结点的可用性。

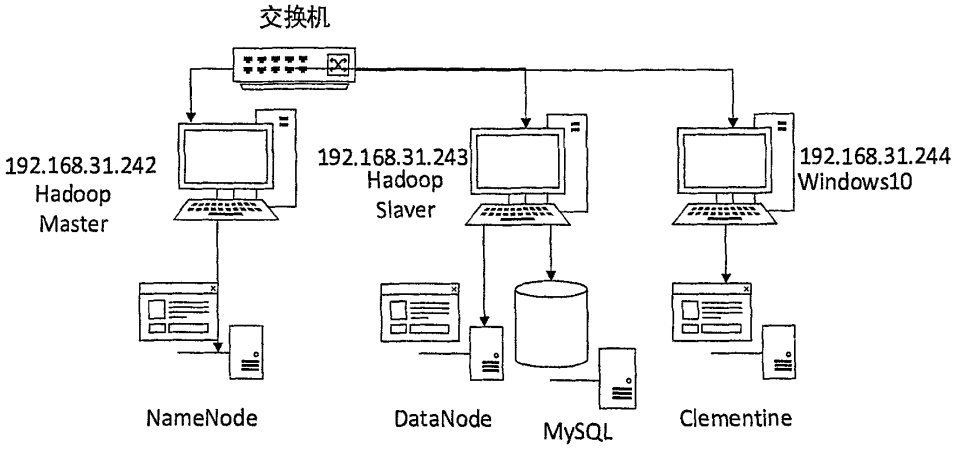


图 5-1 实验系统的物理拓扑图

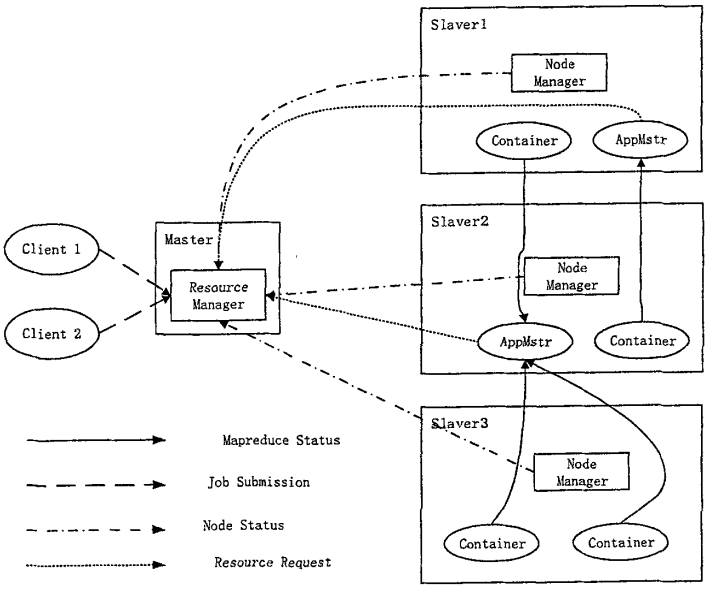


图 5-2 Hadoop Mapreduce Yarn 架构

那么，整个 Yarn 架构的工作流程就可以总结为一下步骤：

(1) 向 ResourceManager 上面提交编辑好的作业；

(2) ResourceManager 为 ApplicationMaster 管理的作业申请 CPU、内存、存储等资源，并尝试与具有心跳的 NodeManager 建立通信，启动 ApplicationMaster；

(3) ApplicationMaster 启动成功，并与 NodeManager 通信，启动对应的 MapReduce 任务；

(4) 当一个 ApplicationMaster 上所有任务完成之后，ResourceManager 注销该 ApplicationMaster，并释放资源。

其次，在 Windows 主机上配置了 Clementine 12.0。Clementine 是一个商业软件，是为了进行稳定的商业数据挖掘而研发的软件。其中包含了很多成熟的建模环境和各式各样的数据预处理部分：K-Means 数据挖掘模块、神经网络、贝叶斯网络、C&R 树等等模型和数据的过滤、分类、填充、导出、分区等等的操作。

软硬件环境搭建完成之后，各部分进行协同工作就能完成基于流量行为特征的网络异常流量分类和检测模型的整个过程。具体流程如图 5-3 所示：

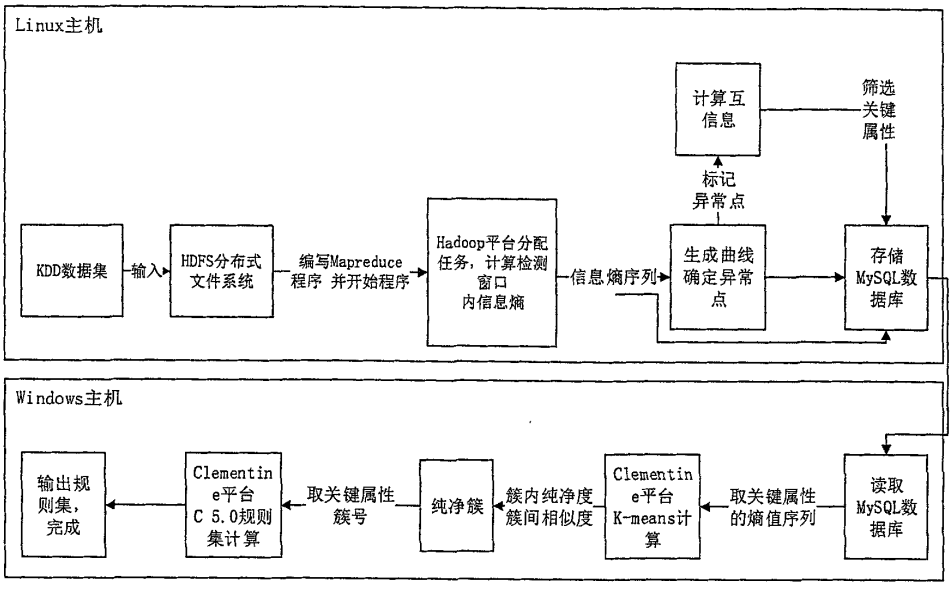


图 5-3 分类检测模型具体实施图

5.2 实验过程

5.2.1 源数据处理

由于方法中使用了已经经过整理的数据集，因此在模型的详细流程中并未涉及到源数据的处理过程。但是直接从网络中使用 Tcpdump 命令采集到的网络数据包文件由于本身是二进制，无法直接导入 Hadoop 的 HDFS 直接进行运算。首先需要对原始的二进制数据进行翻译整理。

Linux 中进行网络数据包抓取的命令是 Tcpdump，tcpdump 命令的格式如下：

Tcpdump [protocol] [-nn] [-i 接口] [-w 储存文档名] [-c 次数]

图 5-4 显示了使用 tcpdump tcp -i eth1 -w /root/test.cap 命令抓取到的数据包经过算法处理之后的结果：

```
public void testDataInputStream(String filepath)throws IOException{
    File file=new File(filepath);
    DataInputStream din=new DataInputStream(new FileInputStream(file));
    StringBuilder hexData=new StringBuilder();
    StringBuilder asciiData=new StringBuilder();
    byte temp=0;
    for(int i=1;i<=file.length();i++){
        temp=din.readByte();
        String str=Integer.toHexString(temp);
        if(str.length()==8){
            str=str.substring(6);
        }
        if(str.length()==1){
            str="0"+str;
        }
        System.out.print(str.toUpperCase()+" ");
        if(i%16==0){
            System.out.print("\n");
        }
    }
}
public static void main(String[] args)throws IOException{
    String filepath="/root/test1.cap";
    Test2 test=new Test2();
    test.testDataInputStream(filepath);

}
}
```

14 14 4B 80 3E 29 98 90 96 C6 E5 F2 08 00 45 00

00 28 4E 31 40 00 40 06 6E 3B CA C9 8C F2 6A 78

BC 2F D3 29 00 50 3D AF B9 33 0C A4 9A 3A 50 11

FF FF C0 34 00 00 00 00 00 00 00 00

图 5-4 源数据处理过程

经过上述过程处理，并结合 TCP 协议本身的字段对应，可以很轻易读出该数据包的信息。例如，次数据包的源 IP 地址为：106.120.188.47（14:14:4b:80:3e:29），目的 IP 地址为：202.201.140.242（98.90.96.c6:e5:f2）。

5.2.2 Hadoop 平台计算过程

整个网络流量数据信息熵值的生成过程以及关键网络流量特征参数提取过程都是在 hadoop 分布式计算平台中完成的，MapReduce 任务复杂将数据持续读入，并进行逐行处理。整个程序分分为 Map 过程、Reduce 过程以及 Driver 过程三个部分。整个 MapReduce 代码是在 Eclipse 中进行编辑的，采用了 Java 语言进行编写。图 5-5(a)和 5-5(b)分别给出了 Mapper 部分和 Reducer 部分的代码。

```
public class ListallMapper extends MapReduceBase
implements Mapper<LongWritable,Text,Text,DoubleWritable> {
   private DoubleWritable one=new DoubleWritable(1);


   public void map(LongWritable key,Text value,OutputCollector<Text,DoubleWritable>output,Reporter reporter)
         throws IOException{

      String line=value.toString();    //dumu yihangshuju :line
      String[] str = line.split(",");  //jiang zheyihang shuju anzhao "," huafen bing baocun zai shuzu zhong


//    if(!str[41].equals("normal.")){  //panduan shi bushi normal
//       output.collect(new Text(str[1]),one); //jiequ guding de zieuan
//    }
      output.collect(new Text(str[32]+" "+str[41]),one);
   }

}
```

图 5-5(a) Mapper 部分的代码

```
public class ListallReducer extends MapReduceBase
implements Reducer<Text,DoubleWritable,Text,DoubleWritable> {


   public static double Logjisuan(double value,double base){
      return Math.log(value)/Math.log(base);
   }


   public void reduce(Text key,Iterator<DoubleWritable> value,
         OutputCollector<Text,DoubleWritable>output, Reporter reporter) throws IOException {

      double sum=0;
      double gailv=0;
      double HE=494021;          //changliang  normal 97278  huozhe  other 396743 or 494021
      while(value.hasNext()){    //dang xiangtong de key haiyou de shihou jinxuxunhuan

         sum+=value.next().get(); //tongji zheyige key gongyou duoshao
      }
//    System.out.println(sum);    zheyiju shi yige ceshi
      gailv=sum/HE;               //jisuan meiyige shuxing de gailv

      double shang=-gailv*(Math.log(gailv)/Math.log(2));
      output.collect(key,new DoubleWritable(sum));
      }
   }
```

图 5-5(b) Reducer 部分的代码

得到所有属性的信息熵值序列之后，利用其中 2 个信息熵值序列进行异常行为的判别，并对判别结果进行标记。随后进行关键流量特征属性的选定，也就是计算各属性与判别结果之间的互信息值的大小，选取前 5 位作为关键流量特征属性，最后将关键属性所对应的信息熵值记录在 MySQL 数据库中。

5.2.3 Clementine 12.0 建模过程

经过 Hadoop 数据处理平台进行信息熵化的数据已经可以进行数据挖掘的操作。此时，选用 Clementine 12.0 进行模型构建和挖掘流程，挖掘和建模的过程可以分为三部：首先进行 K-Means 算法进行数据聚类并以聚类结果标记数据集；其次，对聚类产生的簇进行纯净度检查和相似度检查，合并相似的簇并剔除离群点；最后，将结果输入 C5.0 模型进行规则集生成。图 5-6 给出了 Clementine 12.0 中构建出的整体处理流程。

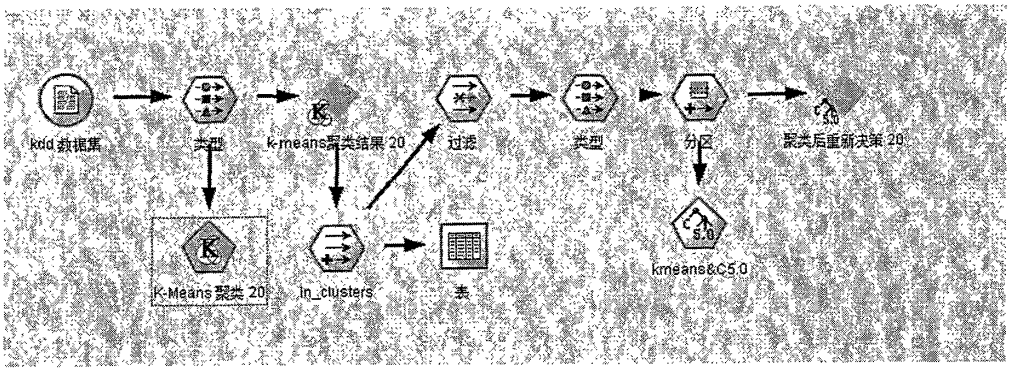


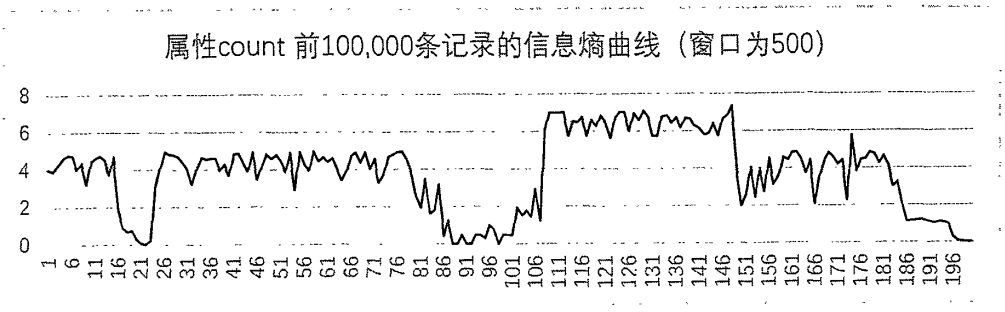
图 5-6 Clementine 整体处理流程

其中，K-Means 聚类 20 模块是进行 K-Means 聚类算法初始值设定的模块，基于之前进行异常标记的结果，K-Means 聚类算法的初始值设定为 20，算法的终止条件设置为经过 20 次聚类或者两次结果之间的差异小于 0.00001；kmeanss&C5.0 模块是进行 C5.0 决策树算法预设值的配置的模块，采用专家模式进行计算，并将决策树分枝修剪的修建严重性设置为 75，每个子分支的最小记录数设置为 2，采用推进技术进行结果的验证(Boosting 技术)并且验证次数设置为 10 次。

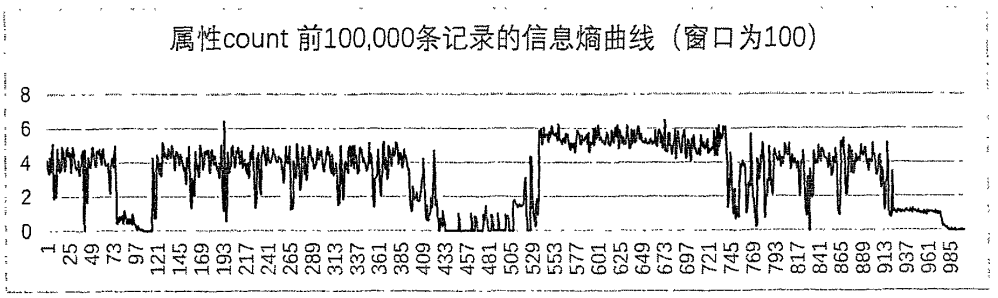
5.3 实验结果分析

5.3.1 对网络异常行为标记的结果

在本文第四章中分析了算法窗口的选择对于检测结果的影响，证明了检测窗口越小，结果越清晰，但是随之而来的是算法的空间复杂度和计算时间延长。图 5-7(a)、(b)中分别给出了在检测窗口为 500 和 100 时，100,000 条记录中属性 count 所生成的信息熵曲线的的对比图。



5-7（a）检测窗口为 500 时属性 count 的熵值曲线



5-7(b) 检测窗口为 100 时属性 count 的熵值曲线

从上图中可以看出，在相同数据量级的前提下，检测窗口的选择会严重影响结果的精度。经过对比，选择了检测窗口的大小为 100。在此基础上，生成的属性 count、src_byte 和 srv_count 的结果如图 5-8 所示

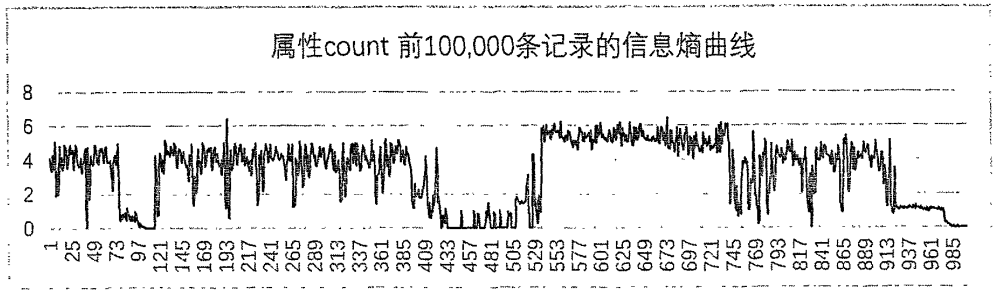


图 5-8(a) 属性 count 的信息熵曲线

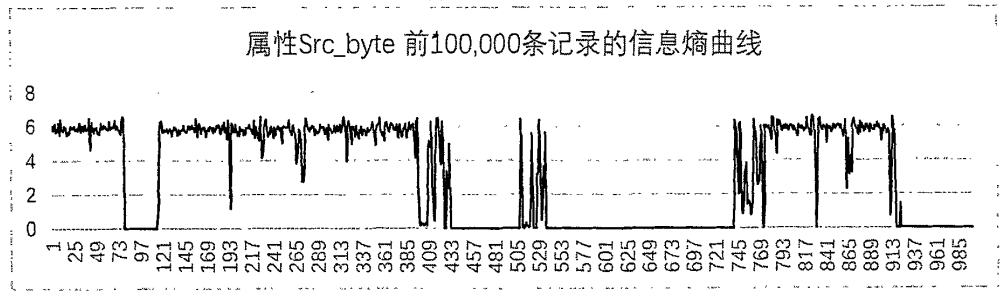


图 5-8(b) 属性 src_byte 的信息熵曲线

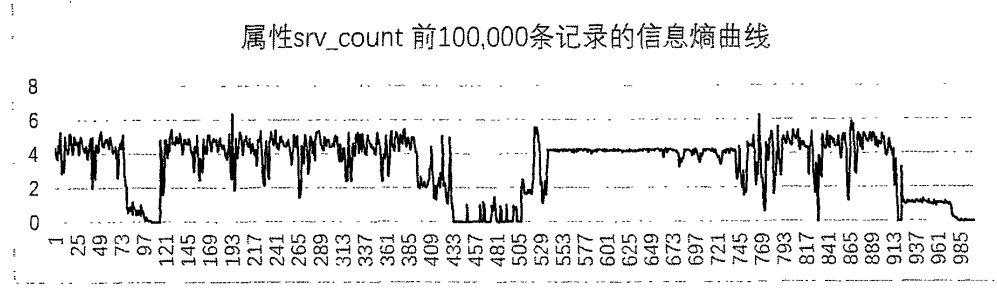


图 5-8(c) 属性 srv_count 的信息熵曲线

选择属性 Src_byte 对初始样本进行标记，之后计算样本属性与标记结果之间的互信息，确定关键流量行为特征参数。得到表 5-2 中的数据：

表 5-2 互信息权值的大小

属性	信息熵	互信息权值
count	4.660021	3.086141
Srv_count	3.799109	2.297533
Src_byte	3.480408	1.556668
dst_host_srv_count	2.779929	1.522957
service	1.884822	1.271968
……	……	……

可以看出，互信息权值较高的几个属性分别为过去两秒内，与当前连接具有相同的目标主机的连接数(count)，过去两秒内，与当前连接具有相同服务的连接数(Srv_count)，从源主机到目标主机的数据的字节数(Src_byte)，前 100 个连接中，与当前连接具有相同目标主机相同服务的连接数(dst_host_srv_count) 等。

对异常行为标记的准确性是决定网络异常流量分类和检测方法准确性的关键。因此在这里有必要使用测试集对本结果进行验证。这里选择了一种对关键信息熵进行加权平均之后的结果进行验证：

$$realtime\_line = \sum_{i=1}^{n} H_i(X)/w_i \qquad \text{(公式 5-1)}$$

因此，选择$w_1 = 3.086$ 、$w_2 = 2.298$ 、$w_3 = 1.557$，对测试集的关键属性进行加权平均，得到的结果如图 5-9 所示。

经过统计，在检测阈值为 0.8 下，测试集的异常流量窗口的个数为 2636 个，占到了总体比例的 84.76%，基本符合测试集中异常流量的比值。因此，可以看出该方法对网络异常行为的标记结果是可行的。

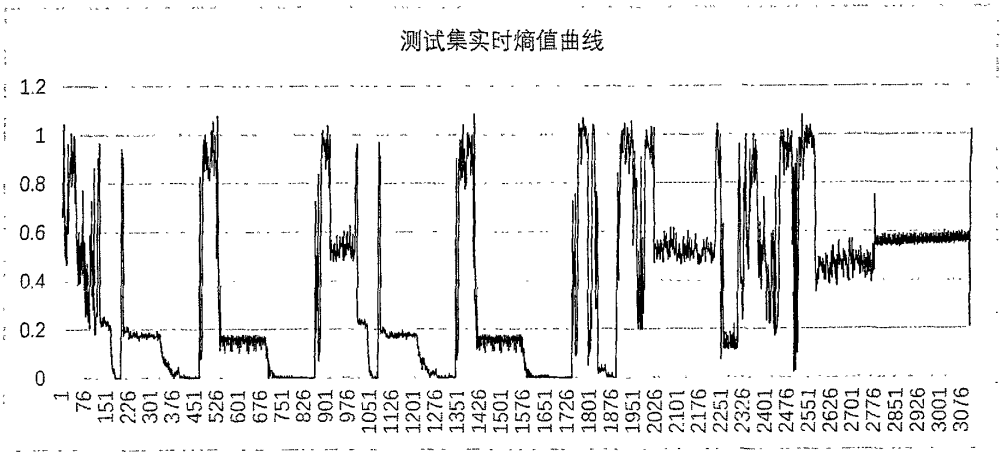


图 5-9 测试集经过加权调整之后的信息熵曲线

5.3.2 规则集的生成

经过异常行为标记的训练集进行 K-Means 聚类操作之后，要对聚类结果进行簇内纯净度检查和簇间相似度计算，经过簇的提纯与合并之后，得到最后的规则集结果。

由于包含 5 条属性的数据集计算结果太过庞大，无法进行结果的验证，这里选取了 3 种属性 1000,000 条记录的数据集进行验证。3 种属性分别为当前连接具有相同的目标主机的连接数(count)；过去两秒内，与当前连接具有相同服务的连接数(Srv_count)；从源主机到目标主机的数据的字节数(Src_byte)。初始的 K 值选取了 7 个。

经过第一次簇内纯净度的检查之后，将聚类 2 重新分成两个子簇，经过随后的簇间相似度计算之后，最终的结果为 6 个簇。图 5-10 给出了未经过簇内纯净度检查和簇间相似度计算的簇生成的规则集与经过检查之后生成的规则集之间的差异。

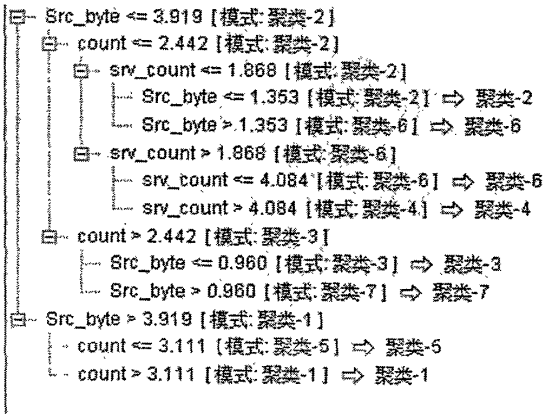


图 5-10(a) 未经过提纯的 C5.0 决策树

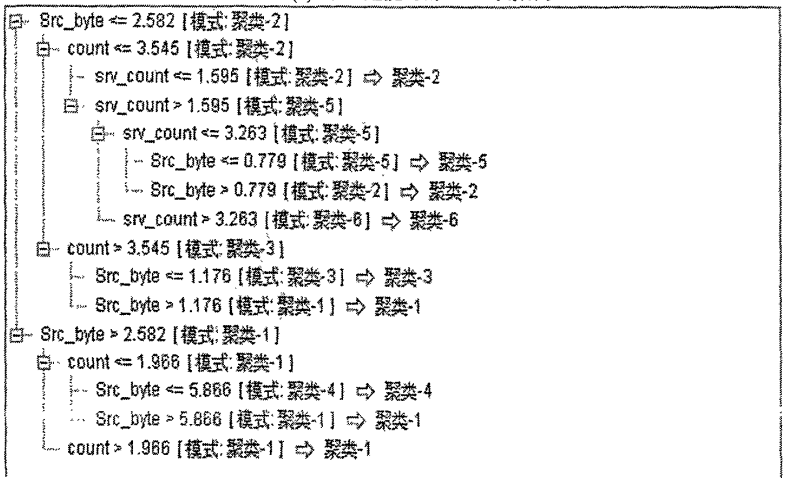


图 5-10(b) 经过提纯的 C5.0 决策树

最后，选取最后一次结果，重新生成网络异常流量分类和检测的规则集。图 5-11 给出了采用 3 个关键属性，100,000 条训练集产生的规则集：

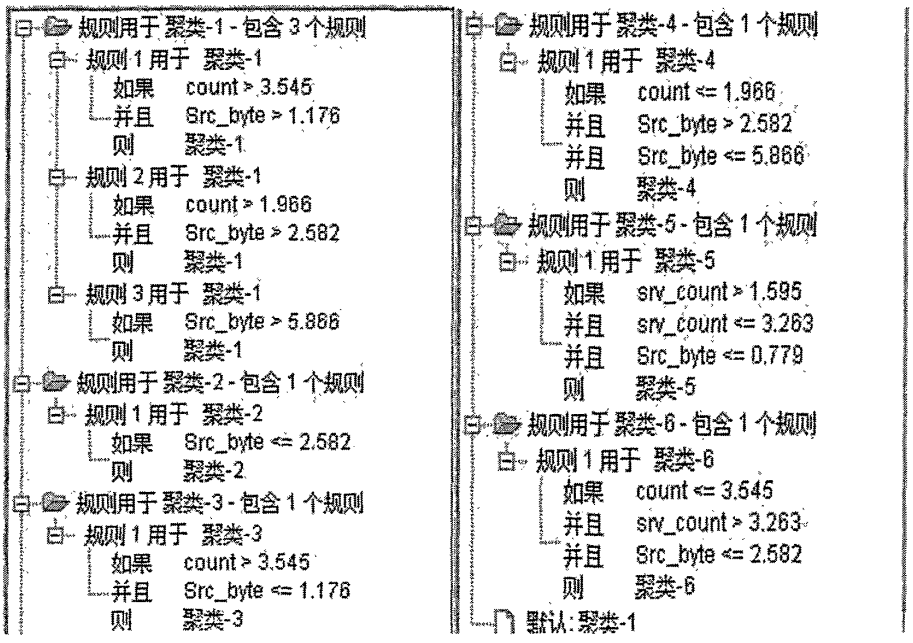


图 5-11 最终规则集的生成

第六章 总结与展望

6.1 本文总结

网络异常流量分类和检测技术是一个非常有实际意义的研究内容，经过实际业务网络的数据集进行训练所得到的网络一场流量分类和检测规则集能十分快速准确地对网络中的异常流量进行检测并分类。本文针对现有的 K-Means 算法的网络异常流量分类模型和 C5.0 的网络异常流量检测模型的缺点，研究出了基于行为分析的网络异常流量分类和检测方法。主要的研究成果包括：

(1) 提出了一种网络流量特征参数提取方法。采用粗粒度的处理方法对现有网络数据流进行处理，并使用大数据处理平台搭建刘特征参数计算平台。该方法即满足大数据量级的环境下网络数据包处理的实时性要求，又能使用流特征参数准确表示现有网络。

(2) 提出了一种结合 K-Means 算法和 C5.0 算法的网络异常流量处理模型。该方法对传统的 K-Means 方法进行改进：首先改进了 K 值选择的问题和初始聚类中心的选择问题；其次，对于算法生成的簇，使用簇内纯净度和簇间相似度计算，保证了簇内元素的纯净性。最后通过 C5.0 算法生成基于聚类结果的规则集。

(3) 在理论分析的基础上，本文设计了实验仿真平台，对结果进行验证。该平台使用 Linux 和 Windows 双平台上搭建了 Hadoop 并行计算平台和 Clementine 数据分析软件。并在 Hadoop 平台上实现了本文的核心算法的编写。

6.2 结果展望

本文的工作还有不足之处，在对结果进行分析的基础上，提出改进的方向：

(1) 本文的仿真系统采用双平台进行搭建，数据的交换采用了 MySQL 远程数据库的形式，这样的方法可能无法拖慢结果的生成时间。因此在后续的试验中考虑使用 Java 语言重新编写 K-Means 算法和 C5.0 算法，并全部放在 Linux 系统中统一实现。

(2) 本文对 K-Means 算法生成的聚类结果进行去除离群点和合并相似簇中，采用了手工重新输入的方式，这样的方法虽然易于操作和程序实现，但是破坏了平台的整体性。因此在后续的操作中应该实现程序的接口，通过接口直接对数据进行读取。这样既满足了平台模块化的需求，同时又能满足平台的整体要求。

(3) 由于时间关系，对方法进行验证时没有找到大量的数据进行验证，可能在针对特有的异常流量中无法进行异常流量模型的生成。同时检测窗口为 100 也损失了一些精度。未来考虑进一步简化方法中包含的 I/O 操作，同时减小窗口大小，这样可能对于分散的、涉及数据包较少的异常行为进行检测。

致谢

在研究生学习阶段，我接触到了许多计算机网络中新的领域，在新知识和新技能的学习中，首先感谢我的导师高玉琢导师，高老师在传授知识中那种敏锐的洞察力和轻松幽默的氛围使我在接触新知识时十分轻松；最让我受益良多的是高老师的处事哲学，高老师一直强调的教书育人，首先要教会的是做人的道理，并且经常用自己为人处事的例子做比喻，让我非常受启发。我觉得高老师这种为人处事的作风值得我一直学习。

同时我想感谢杨军老师、王恒老师、余兆明老师、余秀雅老师、滑斌老师和刘艳青老师，各位老师在学习中和生活中一直很关心我，有时候老师们有时间还会跟我聊一会儿关于我的事情，让我感觉到我十分温暖。

感谢我的同门师兄师姐和师弟师妹，师兄师姐帮助我尽快适应研究生的工作、学习和生活的节奏。同时在生活中大家也互相帮助，共同探讨一些感兴趣的东西。他们是张丽琴师姐、薛砚丹师姐、马国宝师兄、李慧杰师妹、张莹师妹和韩志俊师弟。

感谢同寝室的陈璞、王章训和葛阿雷，是他们陪我一起度过了一个充实的研究生的时光，生活中大家一直相互帮助，并且还经常一起参加体育锻炼。

感谢同为 2013 级的高娟、刘鑫达、刘东平等同学，他们不计个人得失热情帮助其他人的精神使我获益良多，同时也感谢其他的学长学姐们，他们热情的为人处世的作风让我难忘。

最后，感谢我的父亲和家人，他们在我身上倾注了太多的心血，我无以为报，只能在以后的日子中更加孝顺他们；感谢亲爱的薛晓婷同学，你给了我很多鼓励，让我能走到今天。

衷心祝福所有关心和帮助过我的人！

参考文献

[1] Gavalas D, Greenwood D, Ghanbari M 等. Advanced network monitoring applications based on mobile/intelligent agent technology[J]. Computer Communications, 2000, 23 (8): 720-730.

[2] 宿娇娜, 李程, 李巍 等. 基于改进 NB 分类方法的网络异常检测模型[J]. 计算机工程, 2008, (05): 148-149+152.

[3] 诸葛建伟, 王大为, 陈昱 等. 基于 D-S 证据理论的网络异常检测方法[J]. 软件学报, 2006, (03): 463-471.

[4] Bykova M, Ostermann S, Tjaden B. Detecting network intrusions via a statistical analysis of network packet characteristics[C], 309-314.

[5] Denning DE. An Intrusion-Detection Model[J]. IEEE Transactions on Software Engineering, 1987, 13 (2): 222-232.

[6] 李洋, 方滨兴, 郭莉 等. 基于直推式方法的网络异常检测方法[J]. 软件学报, 2007, (10): 2595-2604.

[7] Thottan M, Ji C. Anomaly detection in IP networks[J]. IEEE Transactions on Signal Processing, 2003, 51.

[8] Haining W, Danlu Z, Kang GS. Detecting SYN flooding attacks[C]: IEEE, 1530-1539.

[9] Seong Soo Kim Seong Soo K, Reddy ALN. Statistical Techniques for Detecting Traffic Anomalies Through Packet Header Data[J]. IEEE/ACM Transactions on Networking, 2008, 16 (3): 562.

[10]Li L, Lee G. DDoS Attack Detection and Wavelets[J]. Telecommunication Systems, 2005, 28 (3): 435-451.

[11]Dainotti A, Dainotti A, Pescape A 等. NIS04-1: Wavelet-based Detection of DoS Attacks[C]: IEEE, 1-6.

[12]Barford P, Kline J, Plonka D 等. A signal analysis of network traffic anomalies[C]: ACM, 71-82.

[13]Chen-Mou C, Kung HT, Koan-Sin T. Use of spectral analysis in defense against DoS attacks[C], 2143-2148 vol.2143.

[14]Leung K, Leckie C. Unsupervised anomaly detection in network intrusion detection using clusters[C], 333-342.

[15]Yasami Y, Mozaffari SP. A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods[J]. The Journal of Supercomputing, 2010, 53 (1): 231-245.

[16]Noble C, Cook D. Graph-based anomaly detection[C]: ACM, 631-636.

[17]Eberle W, Holder L. Insider Threat Detection Using Graph-Based Approaches[C], 237-241.

[18]Eberle W, Holder L. Anomaly detection in data represented as graphs[J]. INTELLIGENT DATA ANALYSIS, 2007, 11 (6): 663-689.

[19]Hood CS, Chuanyi J. Proactive network fault detection[C], 1147-1155 vol.1143.

[20]Kline J, Kline J, Sangnam N 等. Traffic Anomaly Detection at Fine Time Scales with Bayes Nets[C]: IEEE, 37-46.

[21]Ndong J, Salamatian K. A Robust Anomaly Detection Technique Using Combined Statistical Methods[C]: IEEE, 101-108.

[22]程光, 龚俭, 丁伟. 基于抽样测量的高速网络实时异常检测模型[J]. 软件学报, 2003, (03): 594-599.

[23]梁昇, 肖宗水, 许艳美. 基于统计的网络流量异常检测模型[J]. 计算机工程, 2005, (24): 123-125.

[24]钱叶魁, 陈鸣, 叶立新 等. 基于多尺度主成分分析的全网络异常检测方法[J]. 软件学报, 2012, (02): 361-377.

[25]穆祥昆, 王劲松, 薛羽丰 等. 基于活跃熵的网络异常流量检测方法[J]. 通信学报, 2013, (S2): 51-57.

[26]颜若愚, 郑庆华. 使用交叉熵检测和分类网络异常流量[J]. 西安交通大学学报, 2010, (06): 10-15.

[27]Barford P, Plonka D. Characteristics of network traffic flow anomalies[C]: ACM, 69-73.

[28]Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. ACM, 2005: 217-228.

[29]Xu K, Zhang Z-L, Bhattacharyya S. Internet traffic behavior profiling for network security monitoring[J]. IEEE/ACM Transactions on Networking (TON), 2008, 16 (6): 1241-1252.

[30]杨静婷. 网络态势感知中的数据仓库设计与关键技术实现[D]: 电子科技大学, 2009.

[31]周正宇. 基于数据仓库的网络态势感知系统研究与设计[J]. 计算机与现代化, 2011, (09): 130-133.

[32]蔡鹏程，李磊. 网络行为分析及研究[J].科技信息，2006,11:171-173.

[33]张永铮，肖军，云晓春，王凤宇. DDOS 攻击监测和控制方法[J].软件学报 Journal of Softwore,2012,23(8):2058-2072.

[34]周颖杰，焦程波，陈慧楠，马力，胡光岷.基于流量行为特征的 DoS&DDoS 攻击监测与异常流识别.计算机应用，2013,33(10):2838-2841.

[35]Eugene H. Spafford, The Internet worm program: an analysis[J].ACM Computer Communication Review, 1989, 19 (1): 17-57.

[36]文伟平，卿斯汉，蒋建春，王业君.网络蠕虫研究与进展[J]. 软件学报 Journal of Softwore,2004,15(8):1208-1219

[37]曹卫东, 白亮, 聂笑盈. 基于 Map/Reduce 的民航高价值旅客发现方法[J]. 计算机工程与设计, 2015, (04): 1078-1083.

[38]国家计算机网络应急技术处理协调中心.CNCERT/CC2014 年中国互联网网络安全报告[EB/OL].北京:2015 年 6 月

个人简历及论文发表情况

个人简历

杨旭，男，汉族，出生年月 1991 年 12 月，山西运城人。本科阶段入学时间为 2009 年，就读于西安邮电大学计算机科学与技术专业。2013 年 9 月进入宁夏大学，攻读计算机应用技术工学硕士学位。

论文发表情况

杨旭，高玉琢. 高校校园网络安全问题及策略研究[J].电脑知识与技术, 2016, (06): 37-38.

杨旭，高玉琢. 信息熵在网络异常检测中的应用[J].网络安全技术与应用, 2016, (04): 30