摘　要

随着攻击手段的复杂化和多样化，传统的入侵检测方法已不能满足安全需求，智能入侵检测已逐渐成为入侵检测乃至整个网络安全领域的研究重点之一。本文采用神经网络模型来实现系统的智能化检测。同时为了加快神经网络的学习收敛速度，引入了能够对大量数据进行有效的属性约简的粗糙集理论，并将这两种技术融合，从而设计实现了一个新的智能入侵检测系统 R_NNIDS。实验测试表明，本系统能够完成基本的入侵检测，有一定的实用性；能够完成从数据发现新的规则，有一定的智能性。

关键词：入侵检测，神经网络，粗糙集，属性约简

ABSTRACT

With the attack means complicated and diversified, the traditional intrusion detection means can't meet the need of network security, and the intelligent intrusion detection system has already become one of keys in the realm for the research of intrusion detection and even in the whole field of network security. This paper uses neural network model to realize the intelligent detection in this system. At the same time for improving the speed of study of network, the rough set that can reduce the attribute from a great deal of data is used. Finally, a new intelligent intrusion detection system-R_NNIDS is designed based on these theories. The result of experiment testing this model shows that this system can complete the basal mission, so it have certain practicability; and this system can find new rules from pockets, so it have certain intelligence.

Shang Libiao (Computer Applied Technology)
Directed by prof. Zhu Youchan

KEY WORDS: intrusion detection system, neural network, rough sets, attribute reduce

声　明

本人郑重声明：此处所提交的硕士学位论文《智能入侵检测系统的研究及其应用》，是本人在华北电力大学攻读硕士学位期间，在导师指导下进行的研究工作和取得的研究成果。据本人所知，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得华北电力大学或其他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。

学位论文作者签名：商李彪　日　期：2004.12.30

关于学位论文使用授权的说明

本人完全了解华北电力大学有关保留、使用学位论文的规定，即：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的,复制赠送和交换学位论文；⑤同意学校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。

(涉密的学位论文在解密后遵守此规定)

作者签名：商李彪　　导师签名：毕如玉

日　　期：2004.12.30　　日　　期：2004.12.30

第一章　绪论

1.1 论文的选题背景及意义

1.1.1 网络安全

计算机网络尤其是因特网的遍及全球，为各种用户提供了多样化的网络与信息服务。利用Internet实现全球范围的电子邮件、电子传输、信息查询、语音与图像通信服务功能，对推动世界经济、社会、科学、文化的发展产生了不可估量的作用，它对人类社会的进步做出了巨大贡献的同时也产生了严重的网络安全问题。尤其是近年来对计算机及网络基础设施的攻击行为，已越来越严重并引起社会广泛关注，无论政府、商务，还是金融、媒体的网站都在不同程度上受到入侵与破坏，特别是各种政府机构的网站，更是成为黑客攻击的主要目标。计算机网络越来越受到广泛关注，成为计算机当今最热门的研究领域之一。

计算机网络安全的威胁来自于多个方面，主要包括如下几种类型：

1. 物理威胁：偷窃、废物搜寻、间谍行为、身份识别错误。

2. 线缆连接：窃听、拨号进入、冒名顶替。

3. 身份鉴别：口令圈套、口令破解、算法考虑不周、编辑口令。

4. 编程：病毒代码如Internet蠕虫、代码炸弹（一旦到了设定的时间，它就被触发并产生破坏）、特洛伊木马（病毒、代码炸弹、蠕虫和诸如此的恶意代码的通称和系统漏洞（亦称为陷阱，通常由系统开发者有意设置的，能在用户失去了对系统的所有访问权后仍进入系统）等。

目前，全世界每年由于信息系统的脆弱性而导致的经济损失逐年上升，安全问题日益严重。网络安全已成为国家与国防安全的重要组成部分，同时也是国家网络经济发展的关键。

1.1.2 入侵检测系统

传统上，我们一般采用防火墙作为网络安全的第一道防线。然而随着攻击者知识的日趋成熟，攻击工具与手法的日趋复杂多样，单纯的防火墙策略已经无法满足需要，网络的防卫必须采用一种纵深的、多样的手段。与此同时，当今的网络环境也变得越来越复杂。各式各样的复杂的设备，需要不断升级、补漏的系统使得网络管理员的工作不断加重，不经意的疏忽便有可能造成安全的重大隐患。在这种环境下，入侵检测系统成为了安全市场上新的热点，不仅愈来愈多的受到人们的关注，

而且已经开始在各种不同的环境中发挥其关键作用。

入侵检测（Intrusion Detection），顾名思义，便是对入侵行为的发觉。它通过对计算机网络或计算机系统中的若干关键点收集信息并对其进行分析，从中发现网络或系统中是否有违反安全策略的行为和被攻击的迹象。入侵检测是一种主动的安全技术，其作用在于：(1)识别入侵者；(2)识别入侵行为；(3)为对抗入侵及时提供重要信息，阻止入侵事件的发生和事态的扩大等。因而，研究入侵检测方法和技术，并根据这些方法和技术建立相应的入侵检测系统对保证网络安全是非常必要的。

目前国内外商品化的产品包括：国外的Cisco公司的NetRanger（网络巡逻兵）、Network Associates公司的CyberCop（计算机巡警）、Internet SecuritySystem公司的RealSecure（实时安全）、以及国内的金诺网安KIDS、启明星辰的天阗黑客入侵检测与预警系统、中科网威“天眼”网络入侵侦测系统等。这些产品对网络中存在的无论是内部攻击、外部攻击或者是误操作等等非法行为都可以进行一定程度上的防范。但是随着网络攻击技术的不断复杂化， 入侵检测产品在规则库的更新方面的要求越来越频繁。为了适应这种变化，我们希望系统的规则库能够进行自我更新，从而实现智能化。但是目前市场上的产品都不具有这种智能更新的特点。总体上说，入侵检测系统智能化研究仍处于理论研究阶段，还有许多的实际问题需要解决。本课题就是在这样的背景下提出的。

1.2 本文的主要工作

1. 研究了入侵检测的基本技术，简述了入侵检测系统的现状和发展方向。
2. 研究了神经网络模型，以及神经网络在入侵检测系统中的应用。
3. 提出了一种基于粗糙集约简算法的智能入侵检测系统的总体模型。
4. 使用 VC++和 Matlab 完成了对智能入侵检测系统中数据采集模块、预处理模块、二进制转换模块、综合分类器模块和响应模块的设计开发工作。
5. 完成了对智能入侵检测系统的性能测试，提出了其优缺点。

1.3 小结

本章首先介绍了论文的选题背景和意义，最后给出了论文所做的主要工作。

第二章　入侵检测系统概述

2.1 什么是入侵检测系统

2.1.1 入侵检测的重要性及其发展历史

随着互联网技术的飞速发展，网络的结构变得越来越复杂，网络安全也变得日益重要和复杂。一个健全的网络信息系统安全方案应该包括安全效用检验、安全审计、安全技术、安全教育与培训、安全机构与程序和安全规则等内容，是一个复杂的系统工程。安全技术是其中的一个重要环节，目前经常使用的安全技术主要有防火墙、防病毒软件、用户认证、加密、入侵检测技术等。多年来，人们在维护信息系统安全时常用的安全技术往往是防火墙。然而，近年来随着各种网络安全事件的发生，以及各种黑客技术如雨后春笋般地在Internet上出现并散布开来，使得人们越来越清醒地认识到仅仅依靠防火墙来维护系统安全是远远不够的。

入侵检测是一种主动的网络安全防御措施，它不仅可以通过监测网络实现对内部攻击、外部攻击和误操作的实时保护，有效地弥补防火墙的不足，而且还能结合其它网络安全产品，对网络安全进行全方位的保护，具有主动性和实时性的特点，是防火墙重要的和有益的补充。

对入侵检测的研究最早可追溯到20世纪80年代，但受到重视和快速发展还是在Internet兴起之后。按时间顺序，入侵检测技术的研究和发展历史概况如下[12]：

1980年，美国人James Aderson首先提出了入侵检测的概念，他将入侵划分为外部闯入，内部授权用户的越权使用和滥用三种类型，并提出用审计追踪来监视入侵威胁。

1986年，为检测用户对数据库的异常访问，在IBM主机上用Cobol语言开发的Discovery系统称为最早的基于主机的入侵检测系统（Intrusion Detection System，IDS）雏形之一。

1987年，美国乔治敦大学的Denning提出了一个经典的入侵检测模型，首次将入侵检测的概念作为一种计算机系统的安全防御措施提出。

1988年，美国SRI公司计算机科学研究室（SRI/CSL）的Teresa Lunt等人改进了Denning提出的入侵检测模型，并创建了入侵检测专家系统（Intrusion Detection Expert System，IDES），提出了与平台无关的实时检测思想。同年，美国军方和政府为Unisys大型主机开发了Haystack系统。

1989年，洛斯阿勒莫斯（Los Alamos）美国国家实验室开发了W&S（Wisdom and

Sense)， Planning Research公司开发了ISOA（Information Security Officers Assistant）。

1990年，加州大学戴维斯分校Heberlein等人提出新概念：基于网络的入侵检测—NSM（Network Security Monitor）。从此，入侵检测被分为两个基本类型：基于主机的系统和基于网络的系统。

1991年，NADIR（Network Anomaly Detection and Intrusion Report）与DIDS（Distribute Intrusion Detection System）提出了收集和合并处理来自多个主机的审计信息来检测针对一系列主机的协同攻击。

1994年，美国人Mark Crosbie和Gene Spafford建议在IDS中使用自治代理（Autonomous Agents）来提高IDS的可伸缩性、可维护性、效率和容错性。

1995年，IDES的完善版本NIDES（Next-Generation Intrusion Detection System）实现了可以检测多个主机上的入侵。

1996年，GRIDS（Graph-based Intrusion Detection System）的设计和实现使得对大规模自动协同攻击的检测更为便利。同年，美国人Forrest将免疫原理运用到分布式入侵检测领域。此后，在IDS中还出现了遗传算法、遗传编程的运用。

1998年，美国人Ross Anderson 和Abida Khattak将信息检索技术引进到了入侵检测领域。

同年，美国人W. Lee提出和实现了在CIDF（Common Intrusion Detection Framework）上实现多级IDS，并运用数据挖掘技术对审计数据进行处理。

最近，美国人Cheung、Steven等人又提出了入侵容忍（Intrusion tolerance）的概念，在IDS中引入了容错技术。

然而，入侵检测技术发展到今天，面对层出不穷、变化多端的攻击仍然显得十分不成熟。

2.1.2 入侵检测的概念、通用模型及框架

“入侵”是个广义的概念，不仅包括发起攻击的人取得超出合法范围的系统控制权，也包括收集漏洞信息， 造成拒绝服务访问（DoS）等对计算机造成危害的行为。入侵行为不仅可以来自外部，同时也可来自内部用户的未授权活动。从入侵策略的角度可将入侵检测的内容分为：试图闯入；成功闯入；冒充其它用户；违反安全策略；合法用户的泄露；独占资源以及恶意使用（标点符号）。而入侵检测是对入侵行为的发觉，它通过从计算机网络或计算机系统的关键点收集信息并进行分析，从中发现网络或系统中是否有违反安全策略的行为和被攻击的迹象。进行入侵检测的软件与硬件的组合便是入侵检测系统（IDS）。

1987 年，美国乔治敦大学的Denning提出了一个抽象的通用入侵检测模型[1]，

如图2-1所示。

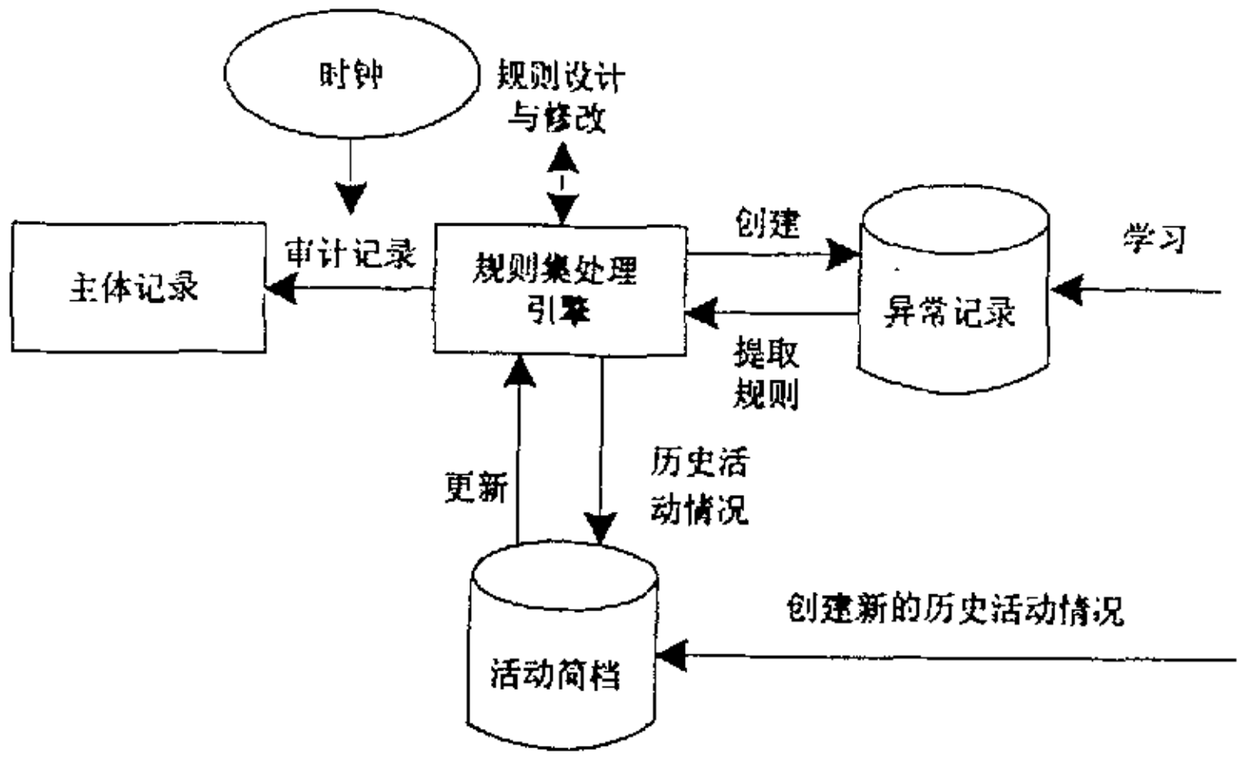

图2-1 Denning模型

该模型主要由六部分构成：主体，对象，审计记录，活动简档，异常记录，活动规则。IDES与它的后继版本NIDES都完全基于Denning的模型，然而并不是所有的IDS都能够完全符合该模型。

这几年，入侵检测系统的市场发展很快，但是由于缺乏相应的通用标准，不同系统之间缺乏互操作性和互用性，大大阻碍了入侵检测系统的发展。为了解决不同IDS之间的互操作和共存问题，1997年3月，美国国防部高级研究计划局（DARPA）开始着手CIDF（Common Intrusion Detection Framework，通用入侵检测框架）标准的制定，试图提供一个允许入侵检测、分析和响应系统共享分布式协作攻击信息的基础结构。加洲大学Davis分校的安全实验室完成了CIDF标准。CIDF阐述的是一个入侵检测系统的通用模型[11]。如图2-2所示

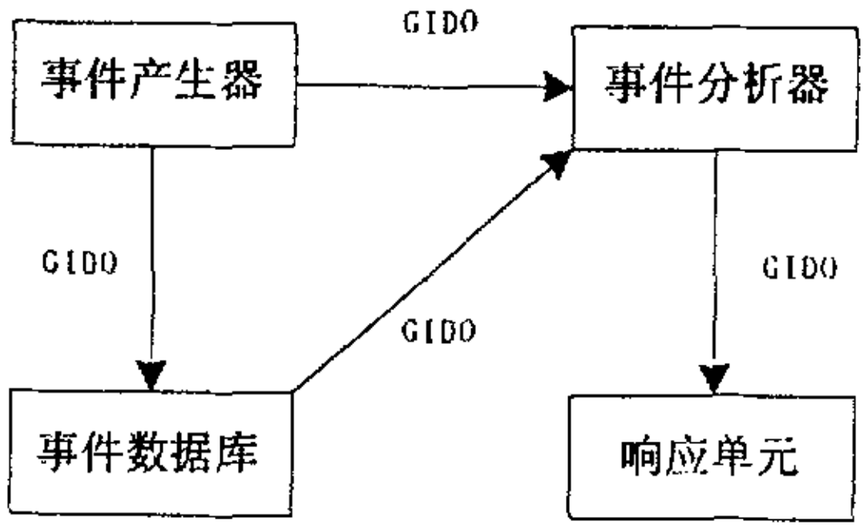

图2-2 CIDF入侵检测框架模型

按功能，它把一个入侵检测系统分为以下组件：

• 事件产生器(Event generators)：从整个计算环境中获得事件，并向系统的其他部分提供此事件。

• 事件分析器(Event analyzers)：分析得到的数据，并产生分析结果。

• 响应单元(Response units)：对分析结果作出反应的功能单元，它可以作出切断连接、改变文件属性等强烈反应，也可以只是简单的报警。

• 事件数据库(Event databases)：是存放各种中间和最终数据的地方的统称，它可以是复杂的数据库，也可以是简单的文本文件。

CIDF将IDS需要分析的数据统称为事件，事件可以是网络中的数据包，也可以是从系统日志等其他途径得到的信息。在这个模型中，前三者以程序的形式出现，而最后一个则往往是文件或数据流的形式。以上四类组件以GIDOs（Generalized Intrusion Detection Objects，通用入侵检测对象）的形式交换数据，而GIDOs通过一种用CISL（Common Intrusion Specification Language，通用入侵规范语言）定义的标准通用格式来表示。

2.1.3 入侵检测系统的分类

入侵检测系统可以从不同的角度进行分类，主要有以下几种分类方法。

1. 根据其采用的分析方法可分为异常检测和误用检测[13]

• 异常检测(Anomaly detection)：假定所有的入侵行为都与正常行为不同，建立正常活动的简档，当主体活动违反其统计规律时，则将其视为可疑行为。该技术的关键是异常阈值和特征的选择。其优点是可以发现新型的入侵行为。缺点是容易产生误报。

• 误用检测(Misuse detection)：假定所有入侵行为和手段（及其变种）都能够表达为一种模式或特征，系统的目标就是检测主体活动是否符合这些模式。关键是如何表达入侵的模式，把真正的入侵行为与正常行为区分开来，因此入侵模式表达的好坏直接影响入侵检测的能力。其优点是误报少，缺点是只能发现攻击库中已知的攻击，且其复杂性将随着攻击数量的增加而增加。

2. 根据系统所检测的对象可分为基于主机的和基于网络的[4]

• 基于主机的IDS(HIDS)：通过监视和分析主机的审计记录检测入侵。优点是可精确判断入侵事件，并及时进行反应。缺点是会占用宝贵的主机资源。另外，能否及时采集到审计也是这种系统的弱点之一，因为入侵者会将主机审计子系统作为攻击目标以避开IDS。典型的系统主要包括上面提到的：Discovery、Haystack、IDES、ISOA、MIDAS以及Los Alamos国家实验室开发的异常检测系统W&S。

• 基于网络的IDS(NIDS)：通过在共享网段上对通信数据进行侦听，分析可疑现象。这类系统不需要主机通过严格的审计，主机资源消耗少，可提供对网络通用的保护而无需顾及异构主机的不同架构。但它只能监视经过本网段的活动，且精确度较差，在交换网络环境下难于配置，防欺骗能力也较差。典型的系统有：为Los Alamos国家实验室的集成计算机网络设计的网络异常检测和入侵检测报告NADIR（是一个自动专家系统）；加利福尼亚大学的NSM系统（它通过广播LAN上的信息流量来检测入侵行为）； 分布式入侵检测系统DIDS等。

• 基于路由器的入侵检测系统：通过对网关中相关信息的提取，提供对整个信息基础设施的保护，确保大型网络计算机之间安全、可靠的连接。一般安装在路由器上，但负载变化对网络性能的影响很大。

以上三种入侵检测系统都具有自己的优点和不足，可互相作为补充。一个完备的入侵检测系统（IDS）一定是基于主机和基于网络两种方式兼备的分布式系统，但现在还没有一种完美的IDS系统模型可以照搬。事实上，现在的商用产品也很少是基于一种入侵检测模型，使用一种技术实现的，一般都是理论模型与技术条件的折衷方案。不同的体系结构、不同的技术途径实现的入侵检测系统都有不同的优缺点，都只能最适用于某种特定的环境。

3. 根据系统的工作方式可分为离线检测和在线检测[12]

• 离线检测：在事后分析审计事件，从中检查入侵活动，是一种非实时工作的系统。

• 在线检测：实时联机的检测系统，它包含对实时网络数据包分析，对实时主机审计分析。

另外，根据系统的对抗措施还可分为主动系统和被动系统；根据系统检测频率可分为实时连续入侵检测系统和周期性入侵检测系统。值得注意的是，以上这几种方法并不相交，一个系统可以属于某个类也可以属于某几类。当然，系统攻击和入侵检测是矛和盾的关系，各种不同机制的入侵检测系统之间并没有绝对的优劣之分。在当前，由于对计算机系统各部分存在漏洞的情况、人类的攻击行为、漏洞与攻击行为之间的关系都没有（也不可能）用数学语言明确的描述，无法建立可靠的数学描述模型，因而无法通过数学和其他逻辑方法从理论上证明某一个入侵检测模型的有效性，而只能对于一个已经建立起来的原型系统，进行攻防比较测试，通过实验的方法在实践中检验系统的有效性。

2.2 入侵检测技术研究[13,14]

从技术上看，入侵可以分为两类：一种是有特征的攻击，它是对已知系统的系统弱点进行常规性的攻击；另一种是异常攻击。与此对应，入侵检测也分为两类：基于误用的（Misuse-based）和基于异常的（Anomaly-based）。

1. 基于误用的检测。首先定义一个入侵特征模式库，包括如网络数据包的某些头信息等，检测时就判别这些特征模式是否在收集的数据包中出现；基于误用的检测优势在于：如果检测器的入侵特征模式库中包含一个已知入侵行为的特征模式，就可以保证系统在受到这种入侵行为攻击时能够准确地检测出来，但是对于某种入侵的变种和新的入侵攻击行为却无能为力。目前的IDS大部分采用这种检测技术。主要包括以下几种：

(1) 模式匹配

这是最为通用的误用检测技术，特点是原理简单、扩展性好、检测效率高、可以实时检测，但是能用于比较简单的攻击方式，且误报率高。著名的Snort系统就采用了这种检测手段。

(2) 专家系统

早期的入侵检测系统多采用这种技术，在这些系统里，入侵行为被编码成专家系统的规则。这些规则既可识别单个审计事件，也可识别表示一个入侵行为的一系列事件。专家系统可以解释系统的审计记录并判断他们是否满足描述的入侵行为的规则。缺点在于：使用专家系统来表示一系列规则不太直观；规则的更新较困难，必须有专业人员做系统规则的更新。

(3) 状态转移分析

它主要使用状态转移表来表示和检测入侵。入侵行为是由攻击者进行的一系列操作，这些操作可以让系统从某些初始状态迁移到一个危害系统安全的状态，每次转移都是由一个断言确定的状态经某个事件触发转移到下一个状态。该方法类似有限状态机，在NSTAT系统中采用了这种技术。

2. 基于异常的检测。首先定义一组系统正常情况的数值，如CPU利用率、内存利用率、文件校验和等，然后将系统运行时的数值与所定义的正常情况进行比较。基于异常的检测无法准确判别出攻击的方法，但是可以判别更广泛、甚至未发现的攻击。采用这种检测技术的IDS有SRI公司研究小组的NIDES等。常用的有：

(1) Denning模型中归纳了四种可以用于异常检测的统计模型：

操作模型(Operational Model)：针对系统中的事件计数度量,超过正常范围触发异常事件报告。

均值与标准偏差模型(Mean and Standard Deviation Model)：以均值之间的标准偏差d 定义信任区间，当系统P用户行为超出信任区间就认为是异常。

多元模型(Multivariate Model)：是均值与标准偏差模型的扩展。它基于对两个或多个系统度量之间的相关分析。

马尔可夫过程模型(Markov Process Model)：将不同类型的审计事件看做是一个状态变量,使用状态转移矩阵表示在系统状态转移的过程中存在的概率特征。检测过程中,使用正常情况下的状态转移矩阵对系统的实际状态变化计算其发生概率，若结果非常小，便认为出现异常。

(2) 统计分析

首先要建立一个统计特征轮廓，它通常由主体（用户、文件、设备等）特征变量的频度、均值、方差、被监控行为的属性变量的统计概率分布以及偏差等统计量来描述。典型的系统主体特征有：系统登录与注销时间、资源被占用的时间以及处理机、内存和外设的使用情况等。优点是可以检测未知的入侵行为，缺点是误报、

漏报率高。

除上述方法外，目前研究人员又提出一些新的入侵检测技术。(1)数据挖掘技术，它能自动分析数据源，得出归纳性的推理，挖掘出潜在的模式。将数据挖掘技术应用到入侵检测中，分别从网络和主机两个方面进行审计数据的挖掘处理，就可以发现误用检测的攻击模式和异常检测的检测模型，从而建立高效的入侵检测模型；(2)免疫系统，利用生物免疫系统和计算机系统的保护机制之间的形似性进行构建，例如分层保护—生物多层保护机制如皮肤、体温等；(3)基于Agent的检测，利用网络中执行特定监视任务的软件实体Agent，并在多个Agent间交流信息达到协同工作；(4)神经网络，它具有自学习、自适应的能力，只要提供系统的审计数据，神经网络就会通过自学习从中提取正常的用户或系统活动的特征模式；而不需要进行大量的统计分析。优点是避开了选择统计特征的困难问题，使如何选取一个主体属性的子集的问题成了一个不相关的事。虽然目前没有成熟的产品，但该方法大有前途。

2.3 入侵检测系统的主要发展方向[7-9]

这些年入侵检测系统得到了飞快的发展，越来越多的公司和科研机构投入到它的研究和开发中去，入侵检测已经成为人们在网络安全领域中关注的热点。但是，它也有很多不足，主要有以下几点：

1. 体系结构存在问题。现在的很多入侵检测系统是从原来的基于网络或基于主机的入侵检测系统经过不断改进而得来的，在体系结构等方面不能满足分布、开放等要求。

2. 误报率和漏报率高。主要体现在以下几个方面：

在高速交换网络中，入侵检测系统不能很好地检测所有的数据包，分析的准确率不高，经常产生漏报。攻击特征库的更新不及时，检测规则的更新总是落后于攻击手段的更新。很多已经公布的攻击并没有总结出相应的检测规则或者已有的检测规则误报率很高。检测分析方法单一，攻击方法越来越复杂，难以发现某一些攻击如拒绝服务攻击（DoS）。

3.不同的入侵检测系统之间不能互操作。在大型网络中，网络不同的部分可能使用了不同的入侵检测系统，但现在的入侵检测系统之间不能交换信息，使得发现攻击时难以找到攻击的源头，甚至给入侵者制造了攻击的漏洞。

4. 入侵检测系统不能和其他网络安全产品互操作。一个安全的网络中应该根据安全政策使用多种安全产品，但目前的大多数入侵检测系统不能很好地和其他安全产品协作。

目前的网络攻击手段向分布式的方向发展，且采用了各种数据处理技术，其破坏性和隐蔽性也越来越强。目前已有的IDS远远不能满足入侵检测的需要，未来入

侵检测系统的研究会朝以下几个方向发展：

1. 分布式、协作式入侵检测技术和通用入侵检测体系结构的研究：包括同一IDS中不同部件的协作，特别是不同平台下部件的合作；IDS与其他网络安全技术相结合，如结合防火墙、互联网工程任务组（IETF）的公钥基础设施（PKIX）、安全电子交易SET (Secure Electronic Transaction) 等新的网络安全技术；还包括进行入侵检测系统的标准化研究，建立新的检测模型，使不同的IDS产品可以协同工作。

2. 入侵检测新技术、新方法的研究：入侵方法越来越多样化与综合化，现有的入侵检测技术已经远不能满足要求。目前，智能化检测的相关技术如神经网络、数据挖掘、模糊技术和免疫原理等等已引起IDS学术界的广泛关注，相信智能入侵检测将是一个有良好应用前景的领域。

3. 高速网络环境下的入侵检测：目前重点研究千兆网下的入侵检测技术，而在高速网络环境下进行入侵检测是一个迫切需要解决的课题。

4. 入侵检测的评测分析方法：用户需对众多的IDS系统进行评价，评价指标包括IDS检测范围、系统资源占用、检全率、检准率和IDS系统自身的可靠性。从而设计通用的入侵检测测试与评估方法与平台，实现对多种IDS系统的检测已成为当前IDS的另一重要研究与发展领域。

5. 应用层入侵检测的研究：许多入侵的语义只有在应用层才能理解，而目前的IDS仅能检测如Web之类的通用协议，而不能处理如Lotus Notes系统、数据库系统等其他的应用系统。

2.4 小结

本章首先介绍了什么是入侵检测系统，包括入侵检测的重要性及其发展历史；抽象的通用入侵检测模型-Denning模型； CIDF标准入侵检测框架模型；入侵系统的分类：异常检测和误用检测，基于主机的检测系统和基于网络的检测系统，离线检测和在线检测。

接下来介绍了两类常用的入侵检测技术：基于特征的检测和基于异常的检测。具体实现中包括专家系统、数据挖掘、状态转移分析、统计分析和神经网络等方法。

最后介绍了入侵检测系统存在的问题包括：体系结构存在问题；误报率和漏报率高；不同的入侵检测系统之间不能互操作；入侵检测系统不能和其他网络安全产品互操作；

以及入侵检测系统主要发展方向：分布式、协作式入侵检测技术和通用入侵检测体系结构的研究；入侵检测新技术、新方法的研究；高速网络环境下的入侵检测；入侵检测的评测分析方法；应用层入侵检测的研究。

第三章　基于神经网络的智能入侵检测系统[26]

3.1 神经网络概述

神经网络(Neural Network，简称NN)是人工智能领域中的一个重要分支，它是由大量的、很简单的处理单元(或称神经元)广泛地互相连接而形成的复杂网络系统。它反映了人脑智能的许多基本特征但并不是人脑神经元联系网的真实写照，而只是对其作某种简化、抽象和模拟。神经网络是由各种神经元按一定的拓扑结构相互连接而成的，它通过对连续的和间断的输入做出状态反馈而完成信息处理工作。神经网络在模仿生物神经计算方面有一定优势，它具有自学习、自组织、自适应、联想、模糊推理等方面的能力。

神经网络模型用于模拟人脑神经活动的过程，其中包括对信息的加工、处理、存储和搜索等过程，它具有以下特点：

1. 神经网络具有分布式存储信息的特点，它存储信息的方式与传统的计算机思维方式是不同的，一个信息不是存储在一个地方，而是分布在不同的位置，网络的某一部分不只存储一个信息，而是分布式存储的，神经网络是用大量神经元之间的联结及对各联结权值的分布来表示特定的信息，因此，当局部网络受损时，这种分布式存储方式仍具有能够恢复原来信息的优点。

2. 神经网络对信息的处理及推理具有并行的特点。每个神经元都可根据接受到的信息作独立的运算和处理，然后将结果传输出去，这体现了一种并行处理，神经网络对于一特定的输入模式，通过前向计算产生一输入模式，各输出节点代表的逻辑概念被同时计算出来，在特定的模式中，通过输出节点的比较和本身喜好的程度而得到特定的解，同时排除其余解，这体现了神经网络并行推理的特点。

3. 神经网络对信息的处理具有自组织、自学习的特点。神经网络中各神经元之间的连接强度用权值来表示，这种权值可事先给定，也可适应周围环境而不断的变化，这种过程称为神经元的学习过程，神经网络具有自学习过程模拟了人的形象思维方法，这与传统符号逻辑完全不同的一种非逻辑非语言的方法。

总之，神经网络是对信息的分布式存储和并行处理为基础，它具有自组织、自学习的功能，在许多方面更接近人对信息的处理方法。

神经网络有许多种类型，主要有前向型、反馈型、随机型和自组织竞争型等。其中前向型神经网络是数据挖掘中广为应用的一种网络，其原理或算法也是其他一些网络的基础。比较成熟的有BP神经网络（Back-Propagation Network，简称BP网络）、径向基函数(RBF)神经网络等。

3.2 神经元模型

在神经网络中，神经元及其突触是神经网络中的基本器件。目前人们提出的神经元模型有很多，其中最早提出且影响最大的，是 1943 年美国心理学家 McCulloch 和数学家 W.Pitts 在分析总结神经元基本特性的基础上首先提出的 M-P 模型，也叫感知器模型。该模型经过不断改进后，形成目前广泛的形式神经元模型。结构如图 3-1 所示：

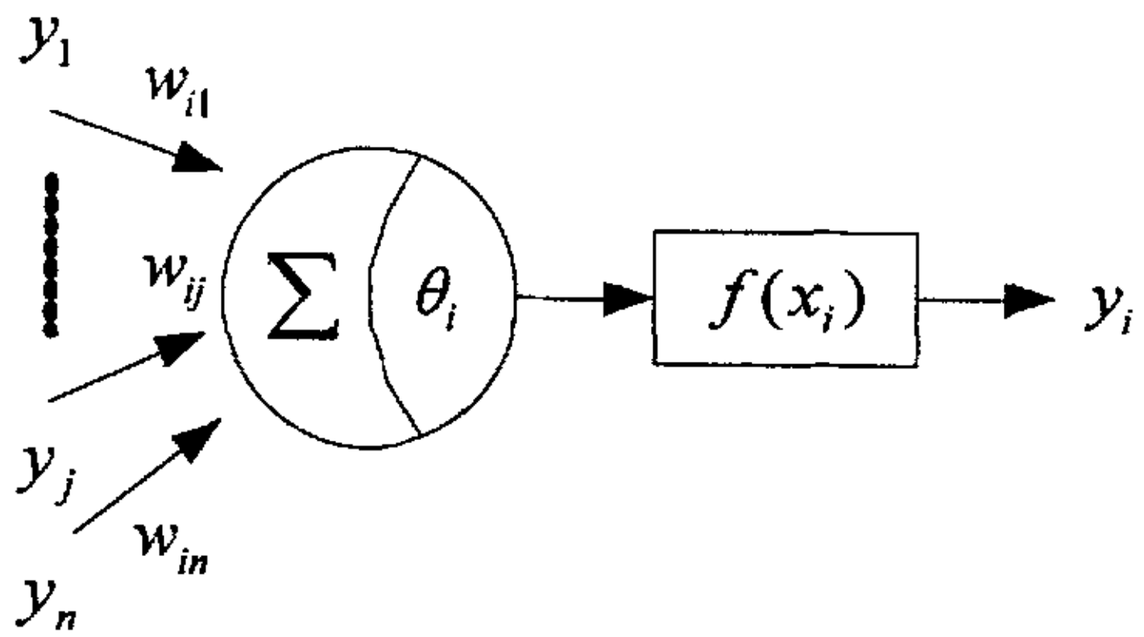

图 3-1a 神经元模型结构

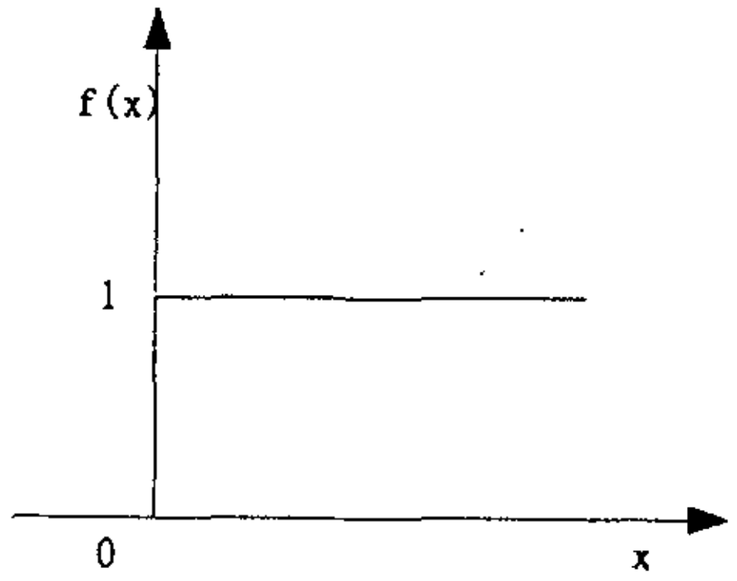

图 3-1b 作用函数

图中：y_i--神经元的 i 的输出，它可与其他多个神经元通过权连接；

y_j--与神经元 i 相连的神经元 j 的输出，也是 i 的输入，$i \neq j$（j=1, 2…n）；

w_{ij}--神经元 j 至 i 的连接权值；

θ_i--神经元 i 的阈值；

$f(x_i)$--非线性函数。

神经元 i 的输出 y_i 可用下式描述：

$$y_i = f(\sum_{j=1}^{n} w_{ij} y_j - \theta_i), \quad i \neq j \tag{3-1}$$

设

$$y_i = f(x_i) \tag{3-2}$$

则

$$x_i = \sum_{j=1}^{n} w_{ij} y_j - \theta_i \tag{3-3}$$

各神经元的输出为“0”或“1”，分别表示“抑制”或“兴奋”状态，则

$$f(x) = \begin{matrix} 1, x \geq 0 \\ 0, x \leq 1 \end{matrix} \tag{3-4}$$

$f(x)$是一个传递函数（Activation Function，又称为作用函数），式（3-4）的传递函数为阶跃函数，如图 3-1（b）所示。

由式（3-1）和式（3-2）可知，当神经元 i 的输入信号加权和超过阈值时，输出为“1”，即“兴奋”状态；反之，输出为“0”，是“抑制”状态。

若把阈值也作为一个权值，则式(3-1)可写为：

$$y_i = f(\sum_{j=0}^{n} w_{ij} y_j) \tag{3-5}$$

式中 $w_0 = -\theta_i$，$y_0 = 1$。

在神经元模型中，传递函数除了式（3-4）的形式之外，还有以下几种，如图 3-2 所示。不同的传递函数可构成不同的神经元模型。

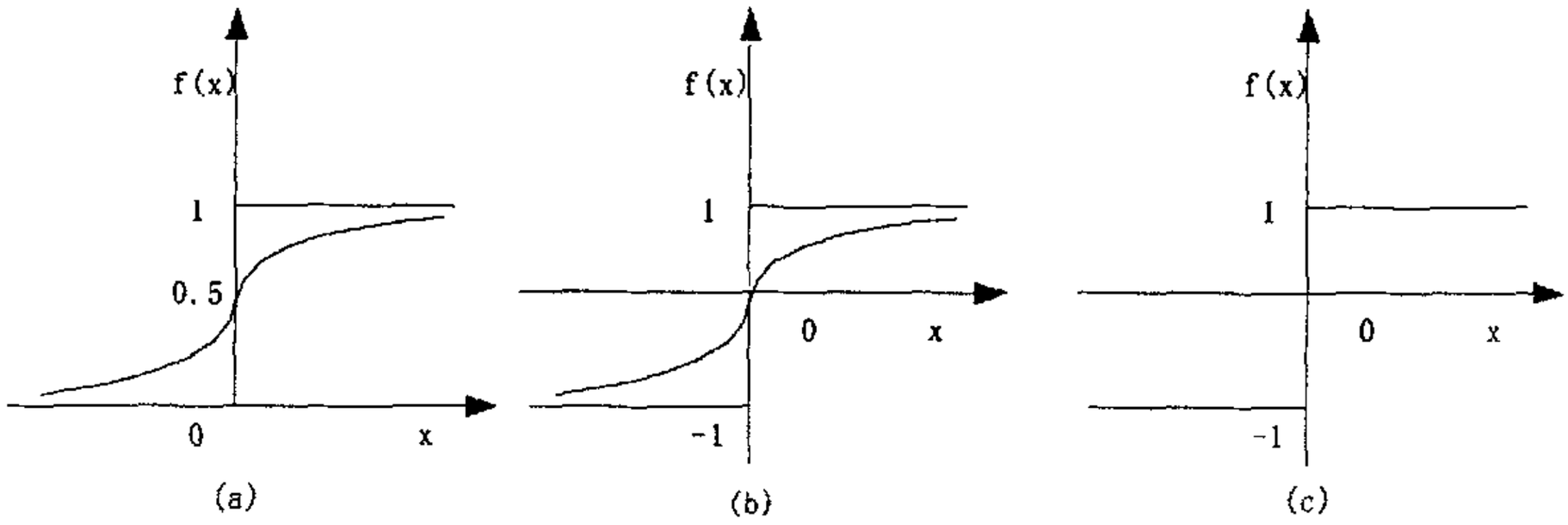

图 3-2 三种非线性传递函数

1. 非对称的 Sigmoid 函数

图 3-2（a）所示的传递函数可用下式表示：

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3-6}$$

Sigmoid 型，也称 S 传递函数，是可微分的。

2. 对称型 Sigmoid 函数

见图 3-2（b），函数是可微的，用下式表示：

$$f(x)=\frac{1-e^{-x}}{1+e^{-x}} \tag{3-7}$$

3. 对称型阶跃函数

图 3-2（c）表示的传递函数，为对称型阶跃函数。

$$f(x)=\begin{matrix}1,x\geq 0\\-1,x\leq 1\end{matrix} \tag{3-8}$$

采用阶跃传递函数的神经元，称为阈值逻辑单元。

3.3 神经网络学习规则

人类高度发展的智能，主要是通过学习获得的，学习是我们获取新知识的过程，因此，要让神经网络具有人脑的特性，必须使神经网络具有学习的功能。神经网络中最早的理论是Hebb规则。Hebb规则代表一种纯前馈、无导师学习，是很多学习规则的基础。

1949年，心理学家Hebb提出了关于神经网络学习机理的“突触修正”的假设。

假设从神经元u_j到神经元u_i的连接强度Δw_{ij}可表示为：

$$\Delta w_{ij}=G[a_i(t),t_i(t)]\times H[y_i(t),w_{ij}]$$

其中$t_i(t)$是神经元的教师信号，函数 G 是神经元u_i的活性度$a_i(t)$和教师信号$t_i(t)$的函数，H 是神经元u_j、输出y_j和连接权值w_{ij}的函数。

输出与活性度之间满足如下关系：　$y_i(t)=f_j[a_i(t)]$

其中$f_j(.)$为非线性函数。当上述教师信号$t_i(t)$没有给出时，函数 H 只与输出成正比，于是有：　$\Delta w_{ij}=\eta a_i y_i$

其中 η 是学习因子。上式表明，对一个神经元，输入和该神经元活性度均较大的情况下，他们之间的连接权重也较大。即，当神经元u_i与神经元u_j同时处于兴奋状态时，两者之间的连接强度应加强。

在 Hebb 规则的基础上，神经网络逐渐发展出两种学习算法用来训练，即指导式（有师）学习算法和非指导式（无师）学习算法。此外，还存在第三种学习算法，即强化学习算法，可把它看做有师学习的一种特例。

1. 有师学习 有师学习算法能够根据期望的和实际的网络输出（对应于给定输入）间的差来调整神经元间连接的强度或权。因此，有师学习需要有个老师或导师来提供期望或目标输出信号。有师学习算法的例子包括δ 规则、广义δ 规则或反向传播算法以

及 LVQ 算法等。

2. 无师学习 无师学习算法不需要知道期望输出。在训练过程中，只要向神经网络提供输入模式，神经网络就能够自动地适应连接权，以便按相似特征把输入模式分组聚集。无师学习算法的例子包括 Kohonen 算法和 Carpenter-Grossberg 自适应谐振理论（ART）等。

3. 强化学习 强化学习是有师学习的特例。它不需要老师给出目标输出。强化学习算法采用一个“评论员”来评价与给定输入相对应的神经网络输出的优度（质量因数）。强化学习算法的一个例子是遗传算法(GAs)。

3.4 前馈人工神经网络与 BP 学习算法

3.4.1 前馈人工神经网络

由BP（Back Propagation，误差反向传播）算法训练的多层前馈人工神经网络(也称为BP人工神经网络)，是人工神经网络分类器中最普遍、最通用的形式。已经证明：由一个单隐含层和非线性兴奋函数组成的多层前馈人工神经网络，是通用的分类器。也就是说，这样的网络能逼近任意复杂的决策边界。图3-3给出一个三层前馈人工神经网络结构示例：

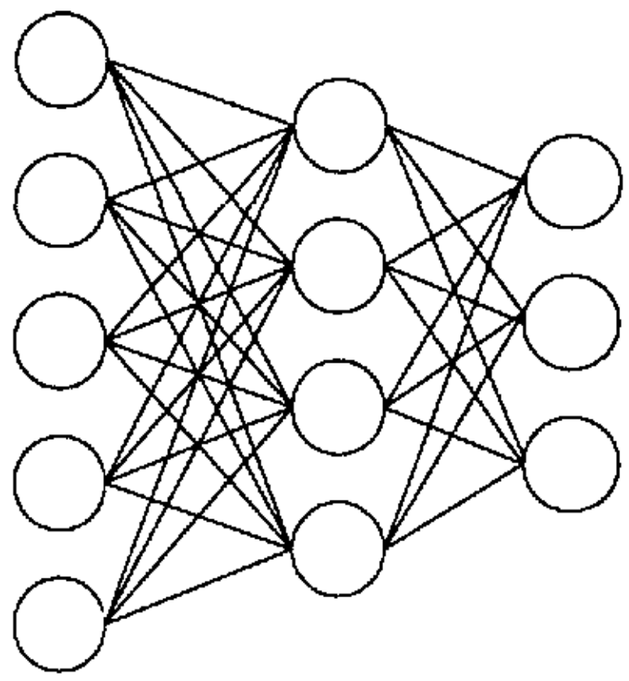

图 3-3 三层前馈人工神经网络实例

将代表待识别模式的输入矢量输入至输入层，并传至后面的隐含层，最后通过连接权输出到输出层。该网络中每个神经元通过求输入权值以及经非线性传递函数传递结果来工作，其数学描述如下：

三层前馈网中，输入向量为$X=\{x_1, x_2, \cdots, x_i, \cdots, x_n\}^T$，如加入$x_0=-1$，可为隐含层神经元引入阈值；隐含层输出向量为$Y=\{y_1, y_2, \ldots, y_j, \ldots, y_m\}^T$，如加入$y_0=-1$，可为输出层神经元引入阈值；输出层输出变量为$O=\{o_1, o_2, \ldots, o_k, \ldots, o_l\}^T$；期望输出向量为$d=\{d_1, d_2, \ldots, d_k, \ldots, d_l\}^T$。输入层到隐含层之间的权值矩阵用V表示，

V=$\{V_1, V_2, \ldots, V_j, \ldots, V_m\}$，其中列向量$V_j$为隐含层第j个神经元对应的权向量；隐含层到输出层之间的权值矩阵用W表示，W=$\{W_1, W_2, \ldots, W_k, \ldots, W_l\}$，其中列向量$W_k$为输出层第k个神经元对应的权向量。

则对于输出层，有

$$o_k = f(net_k) \quad k=1, 2, 3, \cdots, l \tag{3-9}$$

$$net_k = \sum_{j=0}^{m} w_{jk} y_j \quad k=1, 2, 3, \cdots, l \tag{3-10}$$

对于隐含层，有

$$y_j = f(net_j) \quad j=1, 2, \ldots, m \tag{3-11}$$

$$net_j = \sum_{i=0}^{n} v_{ij} x_i \quad j=1, 2, \ldots, m \tag{3-12}$$

以上两式中，$f(.)$均为传递函数。

3.4.2 BP学习算法

以三层前馈网为例介绍BP学习算法。

3.4.2.1 网络误差与权值调整

当网络输出与期望输出不等时，存在误差E，定义如下

$$E=\frac{1}{2}(d-o)^2 = \frac{1}{2}\sum_{k=1}^{l}(d_k - o_k)^2 \tag{3-13}$$

将以上误差定义展开至隐含层，有

$$E = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(net_k)]^2 = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(\sum_{j=0}^{m} w_{jk} y_j)]^2 \tag{3-14}$$

进一步展开至输入层，有

$$E = \frac{1}{2}\sum_{k=1}^{l}\{d_k - f[\sum_{j=0}^{m} w_{jk} f(net_j)]\}^2 = \frac{1}{2}\sum_{k=1}^{l}\{d_k - f[\sum_{j=0}^{m} w_{jk} f(\sum_{i=0}^{n} v_{ij} x_i)]\}^2 \tag{3-15}$$

出上式可以看到，网络输入误差是各层权值w_{jk}，v_{ij}的函数，因此调整权值就可以改变误差E。

显然，调整权值的原则是使误差不断地变小，因此应使权值的调整量与误差的负梯度成正比，即

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial w_{jk}} \quad j=0, 1, \ldots, m; k=0, 1, \ldots, l \tag{3-16a}$$

$$\Delta v_{ij} = -\eta \frac{\partial E}{\partial v_{ij}} = -\eta \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial v_{ij}} \qquad i=0,1,..,n;\ j=0,1,..,m \qquad (3\text{-}16b)$$

式中负号表示梯度下降，常数$\eta \in (0,1)$表示比例系数，在训练中反映了学习速度。可以看出BP算法属于δ学习规则类。

3.4.2.2 BP学习算法权值调整计算公式

式（3-16）仅是对权值调整的思路的数学表达，而不是具体的权值调整计算公式。三层前馈网的BP学习算法权值调整计算公式为：令

$$\delta_k^o = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial o_k} f'(net_k) \qquad (3\text{-}17)$$

$$\delta_j^y = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial y_j} f'(net_j) \qquad (3\text{-}18)$$

利用式（3-13）(3-14)，可得计算公式如下：

$$\Delta w_{jk} = \eta \delta_k^o y_j = \eta (d_k - o_k) o_k (1 - o_k) y_j \qquad (3\text{-}19)$$

$$\Delta v_{ij} = \eta \delta_j^y x_i = \eta (\sum_{k=1}^{l} \delta_k^o w_{jk}) y_j (1 - y_j) x_i \qquad (3\text{-}20)$$

容易看出，BP学习算法中，各层权值调整公式形式上都是一样的，均由3个因素决定，即：学习率η、本层输出的误差信号δ以及本层输入信号Y（或X）。其中输出层误差信号同网络的期望输出与实际输出之差有关，直接反映了输出误差，而隐含层的误差信号与前面各层的误差信号都有关，是从输出层开始逐层反传过来的。

3.4.3 前馈人工神经网络的特点

前馈人工神经网络是目前应用最多的神经网络，这主要归结于基于BP算法的前馈人工网络具有以下一些重要能力。

1. 非线性映射能力

前馈人工神经网络学习和存储大量输入-输出模式映射关系，而无需事先了解描述这种映射关系的数学方程。只要能提供足够多的样本模式对BP网络进行学习训练，它就能完成由n维输入空间到m维输出空间的非线性映射。

2. 泛化能力

前馈人工神经网络训练后将所提供的样本对的非线性映射关系存储在权值矩阵中，在其后的工作阶段，当向网络输入训练时未曾见过的非样本数据时，网络也能完成由输入空间到输出空间的正确映射。这种能力称为前馈人工神经网络的泛化

能力，它是衡量前馈人工神经网络性能优劣的一个重要方面。

3. 容错能力

前馈人工神经网络的魅力还在于，允许输入样本中带有较大的误差甚至个别错误。因为对权值矩阵的调整过程也是从大量的样本对中提取统计特性的过程，反映正确规律的知识来自全体样本，个别样本中的误差不能左右对权值矩阵的调整。

3.5 神经网络与入侵检测[21-23]

人工神经网络（NN）模型力图模仿生物神经系统，通过接受外部输入的刺激，不断获得并积累知识，进而具有一定的判断预测能力。尽管神经网络模型的种类众多，但基本模式都是由大量简单的节点广泛相互连接而构成的一种并行分布处理网络。基于神经信息传输的原理，各个节点通过可变的权值（权值反映了节点之间传递信息时互连的相对强度）彼此相连接，每个节点对 N 个加权的输入求和，当求和值超过某个阈值时， 节点呈“兴奋”状态，有信号输出。节点的特征由其阈值、非线性函数的类型所决定，而整个神经网络则由网络拓扑，节点特征以及对其训练所使用的规则所决定。训练或学习规则就是指通过实例使神经网路自适应或自学习地形成网络中的权函数，以使网络正确理解和解决特定的问题并达到最佳的性能。

可以看到，尽管每个神经元的结构和功能十分简单，但大量神经元构成的网络模型的行为却是多样的。此外 NN 具有相当的健壮性和容错能力，NN 具有大量的运算节点，每个节点又由权系数与大量的其他节点相连接，信息是以分布方式存储于权函数上，当少数节点或节点之间的连接损坏时，不至于对整个网络的性能造成灾难性的影响。NN 还具有自组织性和自学习能力，自组织性和自学习规则使网络能在学习和训练过程中，自适应地发展并总结输入信号的本质特性、规律性等，从而使系统能够发展知识，进而超过原有的知识。

通过对神经网络的介绍，可以看到神经网络在概念和处理方法上都适合入侵检测系统的要求：

1. 神经网络可以通过利用大量实例进行训练的方法学会知识，获得预测的能力，并且这一过程可以是完全抽象的计算，无须强调对数据分布的假设，无须向神经网络解释知识的细节，神经网络可以根据已有的实例自动掌握系统的各个度量之间的内在关系。

2. 可以向神经网络展示新发现的入侵攻击实例，通过再训练使神经网络能够对新的攻击模式产生反应，从而使入侵检测系统具有自适应的能力。

3. 当神经网络学会了系统正常工作模式后，能够对偏离系统正常工作的事件做出反应，进而可以发现一些新的攻击模式。

4. 经过训练后的神经网络将对模式的匹配和判断转换为数值的计算，从而提

高了系统的处理速度，适合于实时处理。

最后，我们根据入侵检测系统的 CIDF 标准建立了基于神经网络的入侵检测系统模型，如图 3-4 所示：

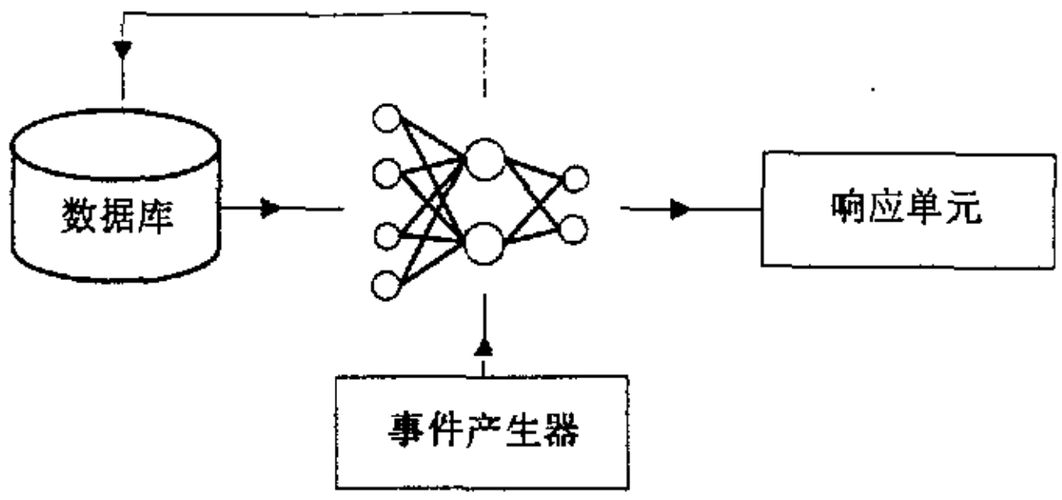

图 3-4 基于神经网络的入侵检测模型

3.6 小结

本章首先介绍了神经网络的基本概念：什么是神经网络；神经网络的基本元素-神经元模型；基本的学习规则等。其中重点介绍了论文以后会用到的基于 BP 学习算法的前馈人工网络。最后讨论了神经网络模型应用于入侵检测系统的可行性，提出了基于神经网络的入侵检测系统模型的基本框架。

第四章　智能入侵检测系统的总体设计

要设计一个入侵检测系统是一项相当复杂的系统工程，网络的拓扑结构的类型、采用数据包捕获、数据检测的方法都会对系统的总体结构产生影响。本章结合R_NNIDS（Neural Network Intrusion Detection System based on Rough Sets）系统对入侵检测系统的总体设计进行阐述，由于以往的IDS系统往往采用C语言进行设计，模块的利用率较低，而且系统的依赖性比较强。在R_NNIDS系统中除神经网络引擎采用Matlab设计以外，其余的部分都采用visual C++来设计，同时参照CIDF标准进行设计，能够提高模块的利用率以及程序的可移植性。

4.1 R_NNIDS系统的可行性

1. 正如CIDF标准所描述的那样，CIDF为系统的每一个模块设计了规定的功能，按其功能所要求，开发出一个入侵检测系统是可能的。

2. 根据其功能要求，可以由网络捕获网络上的数据包，完成事件产生器模块的功能。

3. 所捕获的数据可以由综合分类器模块进行分析，完成事件分析器的功能。

4. 上面所产生的数据写入到特定的事件数据库中。

5. 综合分类器可以检测出偏离正常的异常行为，并总结出新的规则语句，从而具有一定的智能化特点。

6. 计算机网络实验具有较完善的网络环境可以进行实验，可以完成相关的测试。

4.2 R_NNIDS系统的设计目标

根据CIDF标准所描述的入侵检测系统的模块分析，可以分为事件产生器、事件分析器、事件数据库、响应单元划分，R_NNIDS系统的目标如下：

1. 实现一个入侵检测系统，其基本模块与上述CIDF模块基本相对应。

2. 具有智能化特点，能够发现新的入侵行为，产生新的规则。

3. 具有较好的实时性，在网络流量大的高速网络中，系统必须能够实时地捕获到网络数据，并且能够对数据进行实时的处理，从中发现出入侵或入侵企图，并且根据相应的规则做出适当的反应。

4. 安全性与可用性。R_NNIDS系统本身应当尽可能的健壮和完善，具有抵抗入

侵的能力，这样才能为整个网络提供较好的保障。并且能够在设计和实现时，对于较常见的对于入侵检测系统本身的入侵，做到及时的预测和处理。

5. 具有较好的检测准确性，减少误报率。

4.3 R_NNIDS 系统的总体结构设计

根据上述理论及设计目标，将R_NNIDS系统分为数据采集模块、预处理模块、二进制转换模块、综合分类器模块、响应模块等五个模块。各个模块之间的工作流程如图4-1所示。

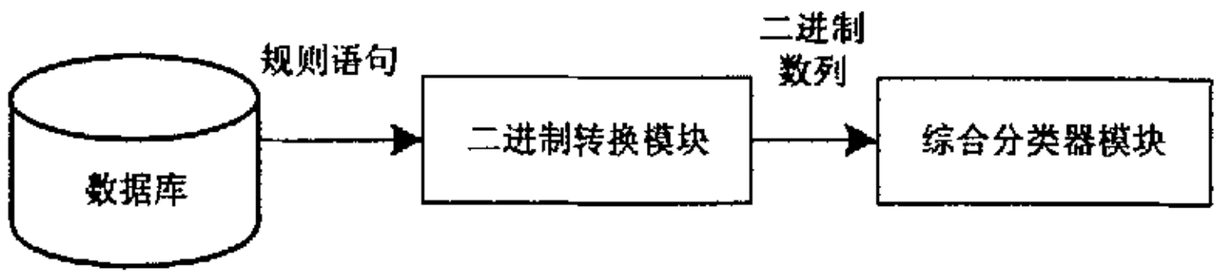

图4-1a 系统中综合分类器模块学习流程

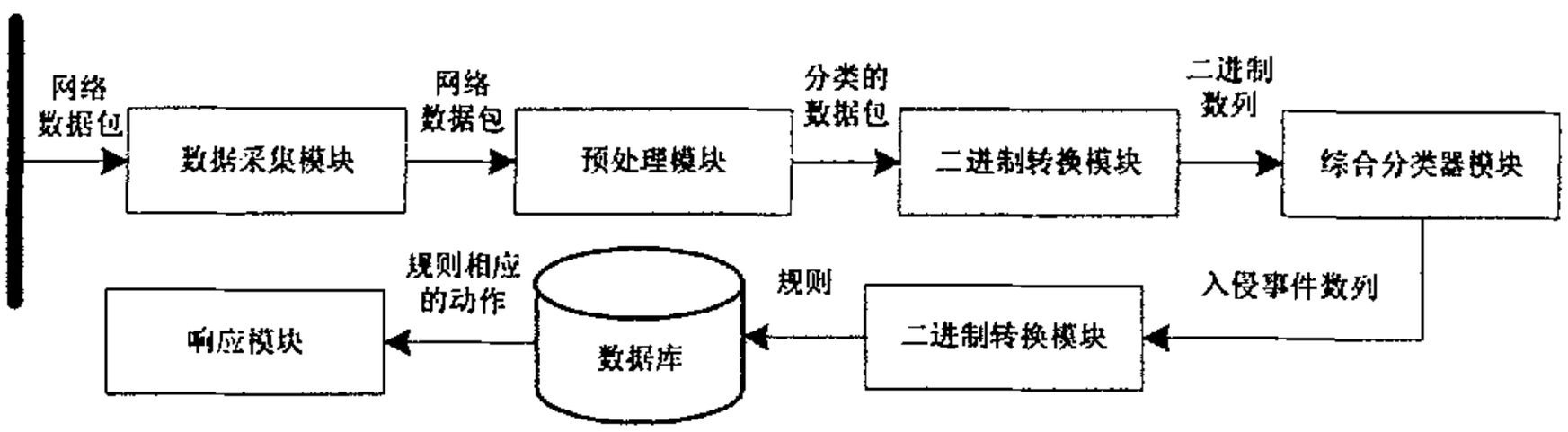

图 4-1b 基于神经网络的入侵检测系统检测流程

流程思想表达如下：

1. 首先离线训练综合分类器：从数据库中提出相当数量的规则语句，转换为二进制编码，输入到综合分类器中进行学习，将获取的知识以隐性形式储存在综合分类器中。

2. 学习完毕后，将综合分类器并入系统中，开始进行入侵检测：对局域网中传输的数据包进行分析，提取符合条件的数据包中的特征信息，转换为二进制编码，输入到综合分类器中进行分类。辨别是否为入侵行为：如果不是，则不做出响应，继续下一个数据；如果是，则提取编码，解码为相应的规则，匹配数据库中规则，做出对应的响应。对于新的规则，则存入数据库中，作出默认响应。

系统中各个模块的功能描述如下：

1. 数据采集模块

数据采集模块负责抓取网络中的每一个数据包，进行初步过滤后送入预处理模块。首先将网络接口设置成混杂模式，用来监听整个网段中的数据。从中过滤出系统所需的数据。最后将数据包送入预处理模块作进一步的处理。

2. 预处理模块

将从数据采集模块得到的数据进行分析，根据相应的协议把这些数据处理后放到指定的数据结构中，供上层模块调用。对于有分片的IP数据包，进行重组。最后将其送入二进制转换模块。

3. 二进制转换模块

负责对数据包的编码，以及规则数据库中的规则的编码和解码。编码用于综合分类器学习或判别，而解码则用于还原规则，用于作进一步的响应。

4. 综合分类器模块

综合分类器是整个模型的核心部分，其采用的神经网络可以看成是由多个较小的具有一定功能的神经网络构成。这些较小的神经网络能够独立运行，用来检测针对某一种类型的攻击，我们称之为基础分类器。因此，可以将复杂多样的检测任务划分为多个小的只针对某一种类型的简单任务，分别由基础分类器来完成。

本文中的综合分类器主要由针对IP、TCP、UDP、ICMP等不同数据包的基础分类器组成。如图4-3所示：

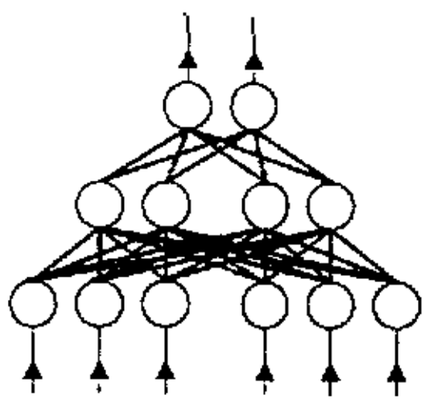

图4-3a 基础分类器

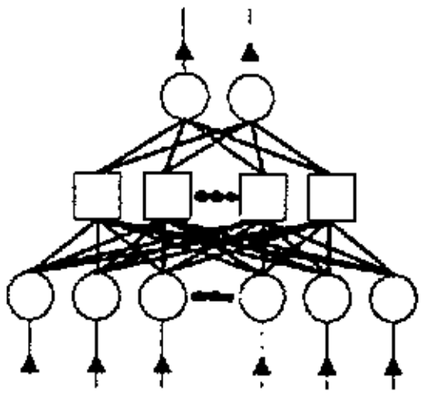

图4-3b 综合分类器

（其中图4-3b中方框部分表示基础分类器）

5. 响应模块

响应模块对综合分类器产生的结果进行处理。当检测结果为异常时，首先显示报警信息，报警信息主要包括：时间、IP地址、端口、入侵类型等，同时把发现的新规则输出到规则数据库；当入侵等级达到一个阈值之后，除了采取报警之外，还要采取一定的主动措施。当检测结果为正常时，不做出反应，继续下一条信息。

4.4 R_NNIDS的主要技术特点

1. 数据采集模块采用Windows下Winpacp库来捕获局域网中数据包。

2. 在阐述了 ICMP, IP, TCP, UDP 四种协议的协议格式的基础上，先根据数据链路层的类型进行解码，紧接着进行高层协议的解码的双层解码方式提取原始数据报的信息。在进行 IP 数据报解码过程中，对于有分片的数据报，进行重组。

3. 利用二进制编码模块将数据转换为二进制数列，作为神经网络输入。

4. 系统采用与 snort 相兼容的规则描述方法，使用了一种简单的、轻量级的描述语言来描述网络上带有攻击标识的数据包，具有较强的描述能力。

5. 利用粗糙集约简算法对神经网络中的输入和隐含层节点进行优化，加快神经网络收敛速度。

6. 利用 Matlab 程序语言实现神经网络引擎程序。

4.5 小结

本章介绍了 R_NNIDS 系统的设计可行性和设计目标。并且着重介绍了组成入侵检测系统的五个模块：数据采集模块、预处理模块、二进制转换模块、综合分类器模块、响应模块。最后，简述 R_NNIDS 的主要技术特点。

第五章　智能入侵检测系统的设计与实现

5.1　数据采集模块的设计与实现

5.1.1 局部网络工作原理

一般情况下，局域网采用的都是以太网，以太网是一种典型的广播信道网络，采用争用型介质访问控制协议(CSMA/CD)来解决在相互竞争的用户之间如何分配信道的问题。正是出于以太网采用这种广播信道争用和共享介质的方式，使得各个站点可以获得其它站点发送来的数据，这就是进行网络监听的物质基础。

以太网的数据传输是基于“共享”原理的：所有的同一本地网范围内的计算机共同接收到相同的数据包。这意味着计算机直接的通讯都是透明可见的。在以太网中网络设备的标识是通过每个网络适配器的唯一的一个48位的MAC地址来区分的。

网络适配器常用的两种接收模式是普通模式(Normal mode)和混杂模式(Promiscuous mode)。在普通模式下，网络适配器只接收数据包中目的MAC地址与该适配器MAC地址相同的数据包，并将接收到的数据包交给高层协议来处理。而对于除了广播数据包(Broadcast Packet)以外的其它包一律丢弃。在混杂模式下，所有与检测系统主机在同一个网段上的数据包都会被接收，而不判断数据包中的MAC地址与主机适配器的MAC地址是否一致。这样，我们通过将网络适配器设置为混杂模式，来达到监听并采集网段上的所有传输数据包的目标。

5.1.2 Windows下网络监听的实现原理

在Windows环境中网络驱动接口标准NDIS (Network Driver Interface Specification)定义了通信协议程序和网络设备驱动程序之间相互通信的Windows规范，并提供了大量的操作函数。它为上层的高层协议驱动提供服务，屏蔽了下层各种网卡驱动的差别。NDIS工作在ISO/OSI参考模型中的第三层和第二层，即网络层和数据链路层。而网络接口卡(NIC)是在OSI参考模型中第二层中的MAC层与网卡驱动程序相互交换数据的。

NDIS向上支持多种网络协议，并且提供一个完备的NDIS库。但库中的各个函数都是工作在核心模式下的，用户直接操作相当复杂。VxD(VxD, Virtual Device Driver)驱动程序和WDM(Win32 Driver Mode)模式驱动程序是Windows提供给用户的Win32环境下的NDIS与用户监听程序之间的两种接口。在Windows下要想实现网络监听的功能必须通过虚拟设备驱动程序或者WDM模式驱动程序，它们提供了外部程序

和网卡NIC之间的接口。如图5-1所示：

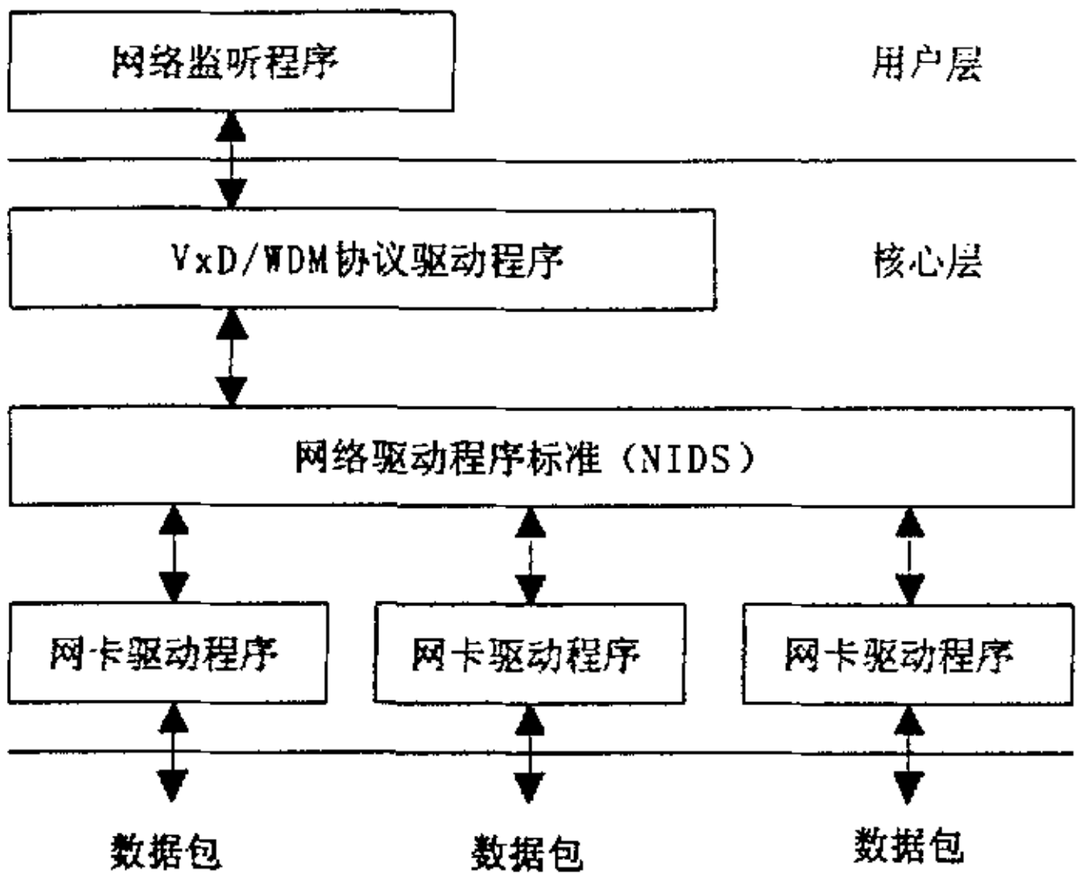

图5-1 在Win32环境下的协议驱动程序

5.1.3 Winpcap的体系结构

在Winpcap（Windows Packet Capture Library，Windows数据包捕获函数库）中提供了VxD/WDM驱动程序。Winpcap与libpcap(Packet Capture library，即数据包捕获函数库)开发包类似，支持WIN32平台上信息包的捕获和网络分析。Winpcap是基于UNIX的Libpcap和BPF（BSD Packet Filter: BSD包过滤器）模型开发的，包括内核级的包过滤驱动程序NPF(Netgroup Packet Filter)、低级动态连接库(Packet.dll)和高级系统无关性库(Wpcap.dll)。其中的NPF是在BPF的基础上开发的。下面，介绍一下Winpcap和NPF。

Winpcap数据包捕获程序是将虚拟设备驱动增加在Window的内核中，可以捕获和发送通过原始套接口的原始数据包(Raw Packets)。它的核心部分是包过滤驱动程序NPF。

NPF过滤器负责将监听到的所有数据包进行过滤，只把用户关心的数据包中的信息提交给用户程序，把一些无关紧要的数据过滤掉。NPF有两个主要部件Network Tap和Packet Filter。Network Tap是一个回调函数(Callback Function)，并不是直接由NPF执行。当一个新的数据包到达时，由网络适配器的设备驱动程序激活。它从内核中的网络设备驱动程序里搜集数据，拷贝并转发到Packet Filter。Packet Filter按照用户事先定义好的过滤规则决定是否接收该数据包，是否将该数据包直接拷贝到用户缓冲区中。NPF将Network Tap放在了网卡程序和NDIS上，接收来自于网卡驱动程序的未经高层协议驱动处理过的原始数据包，并对数据包进行过滤，将符合用户要求的数据包提交给用户程序来处理。

Packet.dll提供了用来直接访问NPF驱动程序的一些API函数，Wpcap.dll是在

Packet.dll的基础上开发的与系统无关的高级系统函数库。Winpcap的结构如图5-2所示：

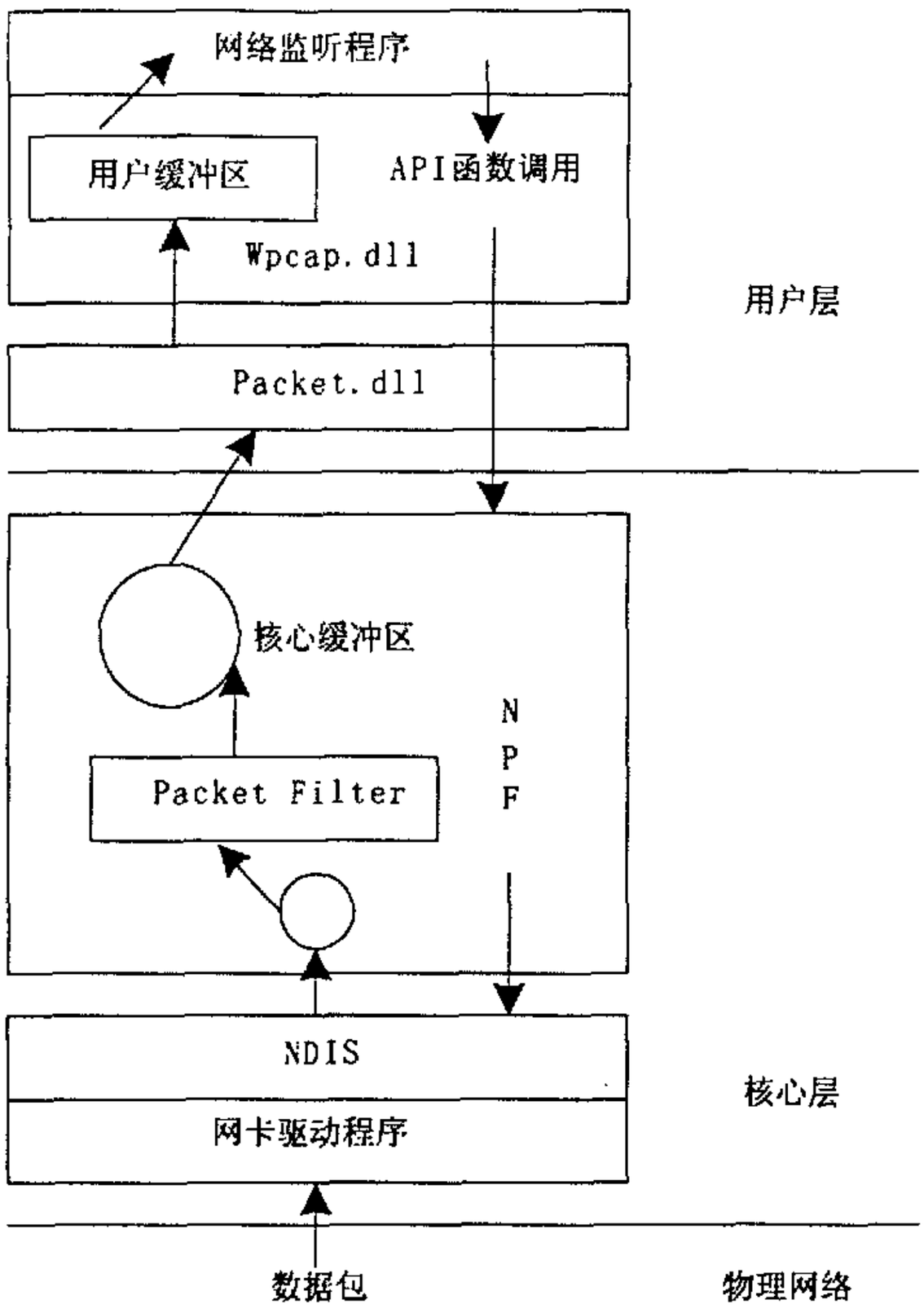

图 5-2 Winpcap 体系结构图

当一个数据包到达网络接口时，链路层驱动程序将其交给Network Tap，Network Tap会把数据包发送给过滤器(Packet Filter)，过滤器将不符合过滤条件的数据包过滤掉后，将符合条件的数据包提交给与过滤器直接相联的系统缓冲区。然后等待系统缓冲区满后，再将数据包拷到用户缓冲区中。监听程序可以直接从用户缓冲区中读取捕获的数据包。由于采用了这种内核过滤、数据包环形缓冲、数据包批量拷贝的方式，减少了用户态进程调用内核的次数，极大地提高了数据包的处理能力。

5.1.4 程序中采用的主要函数

利用Winpcap开发数据包采集程序主要是通过调用其中的Packet.dll和Wpcap.dll中两个动态连接库提供的API函数，将网卡上收到的数据包存储到应用程序中。本程序主要是利用Packet.dll中的函数实现包的捕获，下面列出用到的一些主要函数[29]：

1. ULONG PacketGetAdapterNames(PTSTR pStr,PULONG BufferSize)

取网卡名函数。从注册表中取出系统中安装的网卡的名字，并将其存放到pStr指向的缓冲区，BufferSize是缓冲区的大小。

2. LPADAPTER PacketOpenAdapter(LPTSTR AdapterName)

网卡打开函数。根据适配器的名称打开相应的网卡，AdapterName是要打开的网卡的名字，返回该网卡的结构指针。

3. BOOLEAN PacketSetHwFilter (LPADAPTER AdapterObject,
ULONG Filter)

设置网卡工作模式函数。该函数对接收进来的包设置过滤器，使网卡按指定的Filter模式工作，若成功则返回TRUE，失败返回FALSE。其中Filter的设置在ntddndis.h中定义，其值可以是:

NDIS_PACKET_TYPE_PROMISCUOUS：混杂模式；

NDIS_PACKET_TYPE_DIRECTED：直接模式；

NDIS PACKET_TYPE_BROADCAST：广播模式；

NDIS_PACKET_TYPE_MULTICAST：多播模式；

4. PacketSetBuff(LPADAPTER AdapterObject,int dim)

设置缓冲区函数。该函数为AdapterObject指向的网卡设置缓冲区，其大小由dim指定。若设置成功函数返回TRUE，若系统内存不足以分配新的缓冲区则返回FALSE。

5. PacketInitPacket(LPPACKET lpPacket,PVOID Buffer,UINT Length)

数据包初始化函数。该函数初始化一个包结构:JpPacket指向要被初始化的数据包结构，Buffer指向存放数据包的用户缓冲区，Length是缓冲区大小。

6. PacketReceivePacket(LPADAPTER AdapterObject, LPPACKET lpPacket,
BOOLEAN Sync, PULONG BytesReceived)

数据包接收函数。该函数从AdapterObject指向的网卡中取出数据包，并将数据包存入lpPacket指向的数据包结构，Sync是同步、异步标志，实际接收的字节数用BytesReceived表示。

7. PacketFreePacket(IpPacket)

数据包释放函数。该函数释放lpPacket指向的数据包结构。

8. PacketCloseAdapter(lpAdapter)

关闭网卡函数。该函数释放网卡结构，并关闭lpAdapter指向的网卡。

9. BOOLEAN PacketSendPacket(LPADAPTER AdapterObjcet,
LPPACKET lpPacket,BOOLEAN Sync)

数据包发送函数。lpAdapter是已经打开的网卡的结构指针，lpPacket是要发送的数据包的结构指针，Sync是同步、异步标志，TRUE为同步，FALSE为异步。该

函数从AdapterObject指示的网卡上向网络发送原始数据包。

10. PVOID PacketAllocatePacket(VOID)

数据包申请函数。该函数分配一个数据包结构并返回一个指向数据包结构的指针，该数据包结构应被PacketInitPacket()初始化。

11. PacketSetReadTimeout(LPADAPTER AdapterObject,int timout)

读延时函数。该函数设置网卡的读延时时间。

12. PacketGetStats(LPADAPTER AdapterObject,struct bpf_stat *s)

统计函数。该函数可以统计出两个驱动程序的内部变量，一个是从打开网卡后由该网卡接收到的数据包个数；一个是由网卡接收了但在核心级丢失了的数据包个数。

5.1.5 模块具体设计

在R_NNIDS系统中，数据采集模块主要由NNPcap类组成。其主要的接口函数如表5-1所示：

模块名称	对应类	接口函数	说明
数据采集模块	NNPcap	InitDevice()	初始化设备
		StartPcap()	开始捕获包
		OutputPcap()	输出数据包
		SetDevice()	设置监听设备
		SetFiter()	设置过滤规则

表5-1 数据采集模块接口函数表

数据采集模块程序流程如图5-3所示：

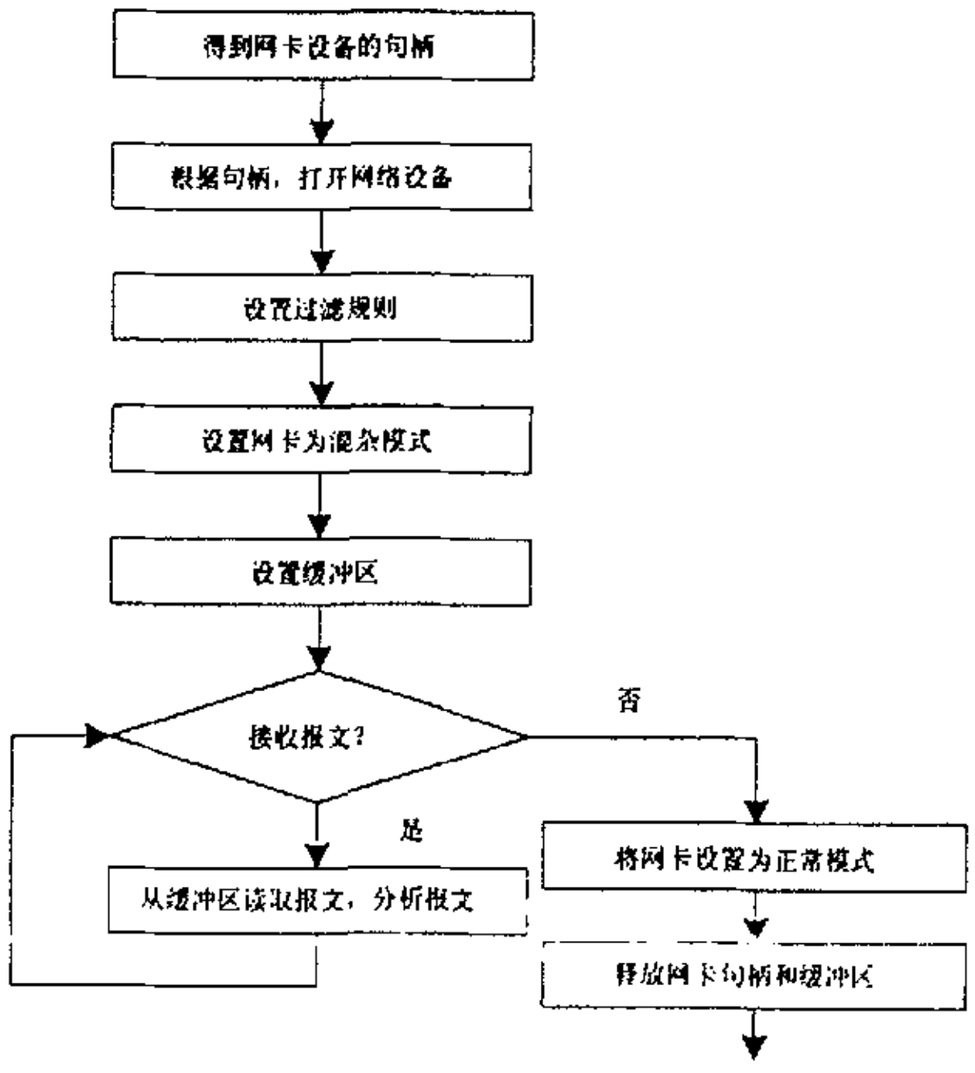

图5-3 数据采集模块程序流程图

5.2 预处理模块设计与实现

数据采集模块所捕获的链路层数据传送到预处理模块后，预处理模块必须对这些数据进行分析，根据相应的协议把这些数据处理后放到指定的数据结构中，供上层模块调用。此外预处理模块还要对这些数据包进行一些基本的校验，丢弃出现错误的数据包，以及完成IP分片重组等工作。

5.2.1 预处理模块支持的协议[31,32]

在R_NNIDS系统中，系统支持对ICMP，IP，TCP，UDP四种协议的处理。

1)数据链路层协议

不同的局域网采用不同的数据帧，包括令牌环网数据帧、FDDI数据帧、以太网数据帧、PPP数据帧、SLIP数据帧、原始数据帧、PPPoE数据帧以及一些公司自定义的一些数据帧(如Cisco公司就自定义了一些路由器级联相关的协议)。但目前主要的局域网都是采用的以太网，所以R_NNIDS系统只支持对以太网帧进行解码。

以太网的帧格式如图5-4所示。

目的地址	源地址	类型	数据	CRC
6	6	2	46-1500	4

图 5-4 以太网帧格式

2) IP报文的封装

TCP/IP网络上传送的是IP数据包，协议的基本格式如图5-5所示。有关协议字段的具体说明可以参考RFC791，在此不再赘述。

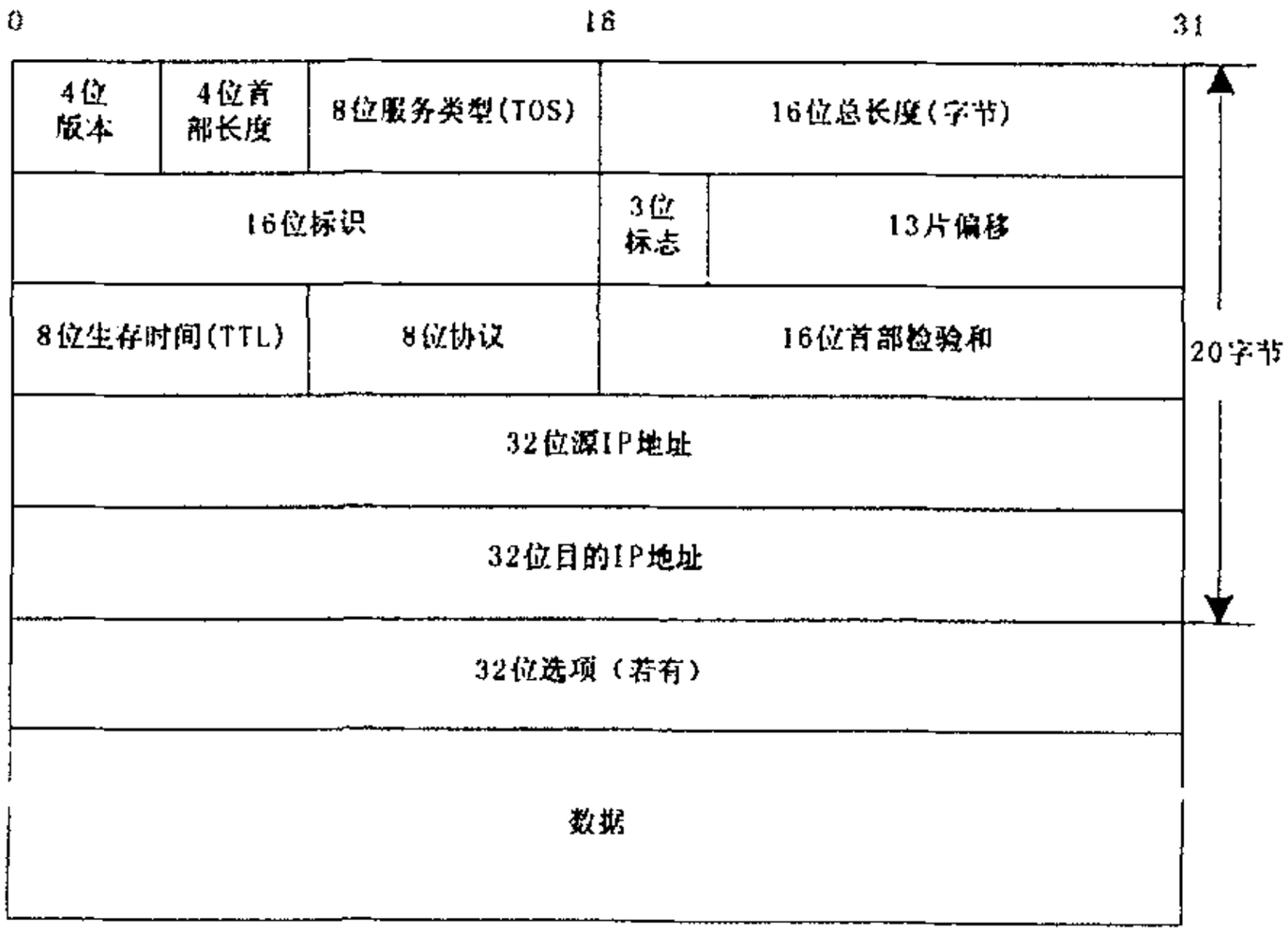

图 5-5 IP 数据包格式

IP数据包位于网络层，要由数据链路层来进行封装，因此，当数据采集模块将数据交给预处理模块之后，预处理模块要根据封装的过程对数据帧进行分析。图5-6说明了IP数据包的封装。

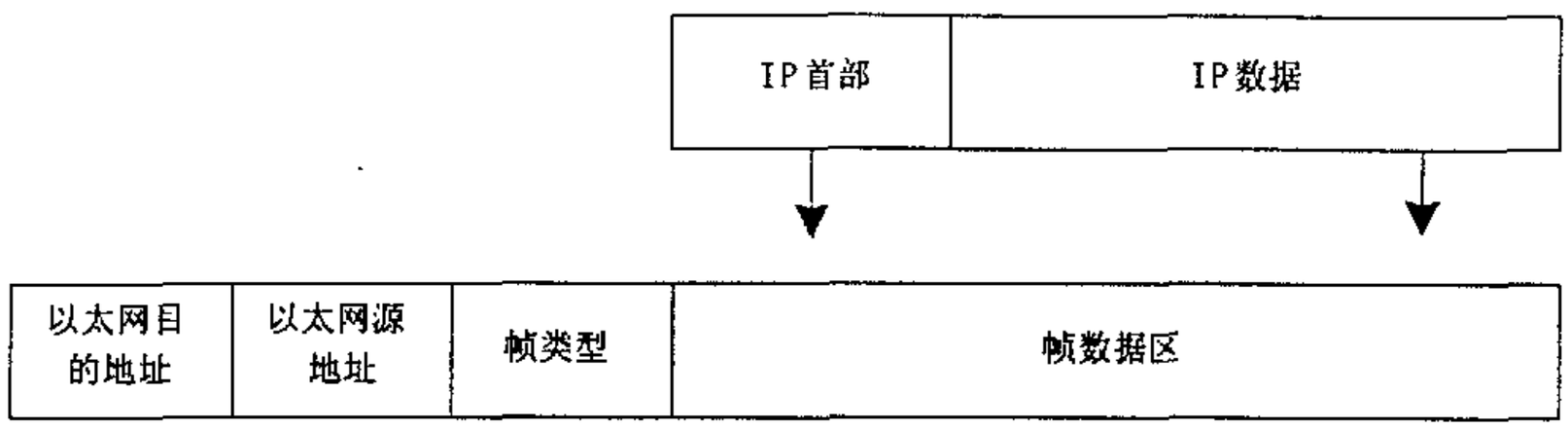

图 5-6 IP 数据包的封装

为了识别IP数据包，发送方给帧首部的类型字段分配一个特殊值，预处理模块可以根据这个特殊值提取出IP数据包。在以太网中，类型字段是0800H。

3) IP分片重组原理

在IP分组跨越网络边界时，由于MTU的变化，IP数据包转发软件(如路由器中软件)会对数据进行分片，有经验的攻击者就会利用网络的这一特性，把一个攻击放在若干个分片中，绕开系统的检测，达到其攻击的目的。因此，有必要对分片的IP分组进行重组，重组之后再进行辨别，检测出入侵。

IP数据包分片是由如图5-7的标识(Identification)、分片标志(Flags)和分片偏移(Fragment Offset)控制。标识由前16位组成，分片标志(Identification)由3位组成，分片偏移由13位组成，如图5-7所示。

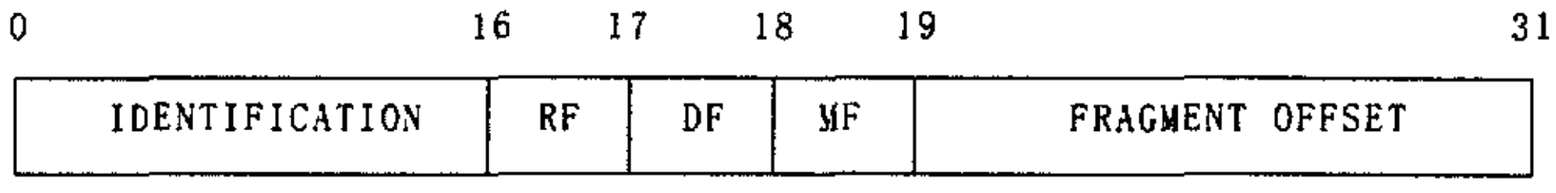

图5-7 分片控制说明

标识字段含有一个唯一标识该数据包的整数，分片软件在进行分片时，赋予同一个数据包分片一个唯一的标识，让目标主机知道每个到达的数据包分片属于哪个数据包。目的主机通过数据包分片的标识字段及源网点地址来识别数据包。发送IP数据包的主机必须为每个数据包生成一个唯一的值作为标识符。

三个比特的标识字段中的两个低比特位控制分片。

IDENTIFICATION:数据片标识，每一个数据包具有唯一的标识， 同一分片的数据包具有相同的标识。

DF:控制指定数据包是否分片，为1是表示不能分片。

MF:为1时表示本分片是原始数据包分片的最后一片，为0时表示不是最后一片

RF:保留位。

FRAGMENT OFFSET:分片偏移量是指本分片在整个数据包中的分片偏移，是十进制表示的八位组。

5) ICMP, TCP, UDP协议的封装

ICMP(Internet Control Message Protocol, Internet控制报文协议)协议用于主机或路由器报告差错情况和提供有关异常情况的报告，协议格式如图5-8所示。

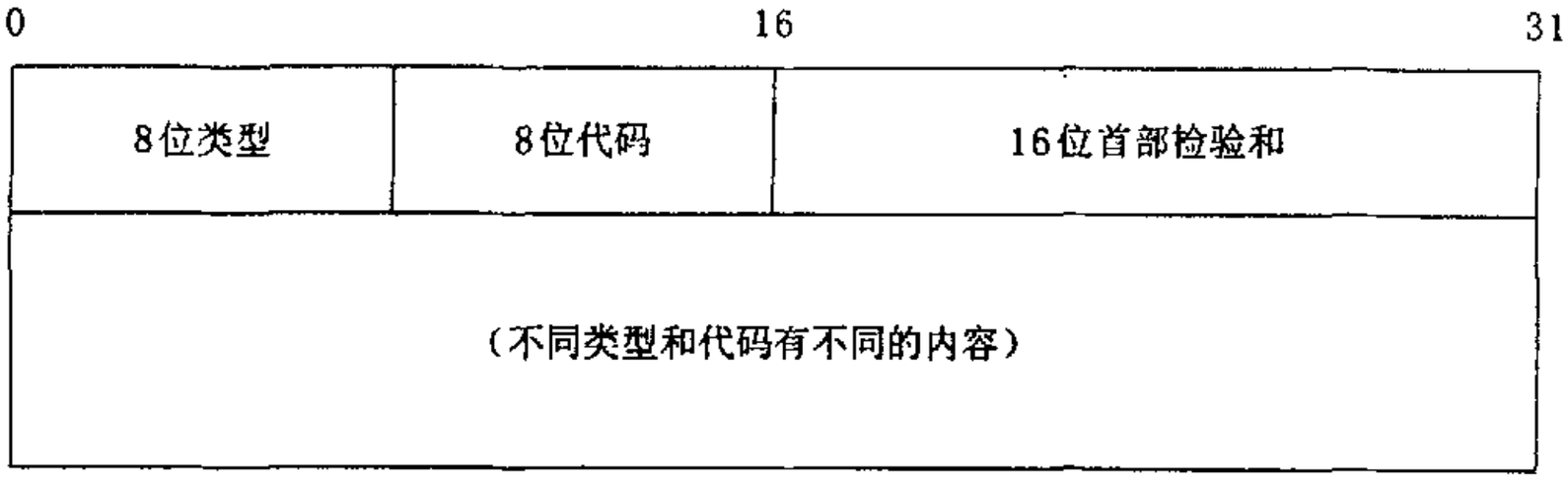

图 5-8 ICMP 协议格式

ICMP分为不同的类型，每一种类型都会有一个扩展首部。常见的ICMP类型有回送应答、目的站不可达、源站抑制报文等，具体的类型见参考文献。

TCP (Transmission Control Protocol，传输控制协议)和UDP (User Control Protocol，用户数据包协议)都位于传输层。TCP是面向连接的协议，负责两台主机之间用户数据的可靠传输。TCP协议格式如图5-9所示。

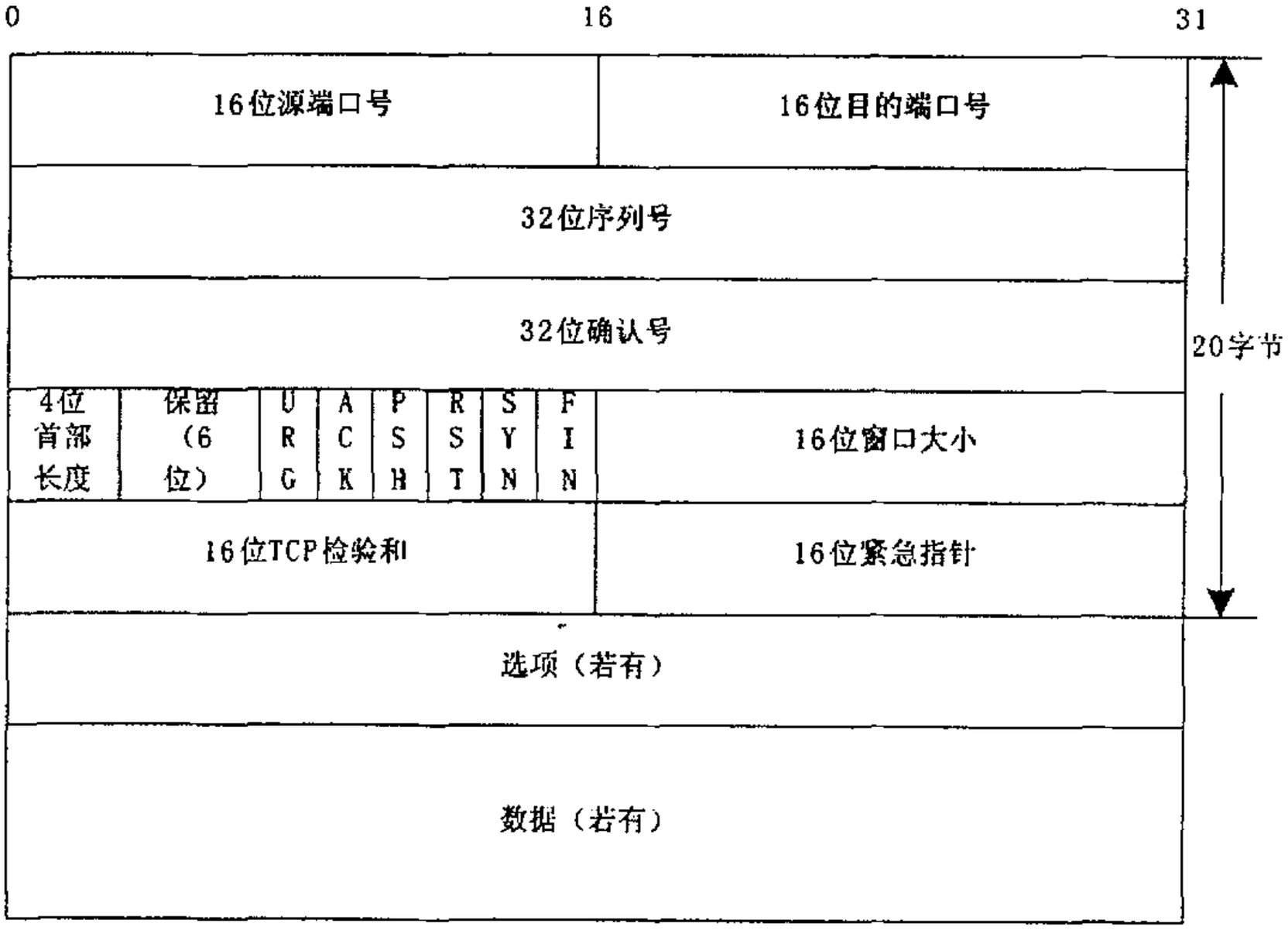

图 5-9 TCP 协议格式

UDP(User Datagram Protocol)是一种无连接协议，其协议格式如图5-10所示。

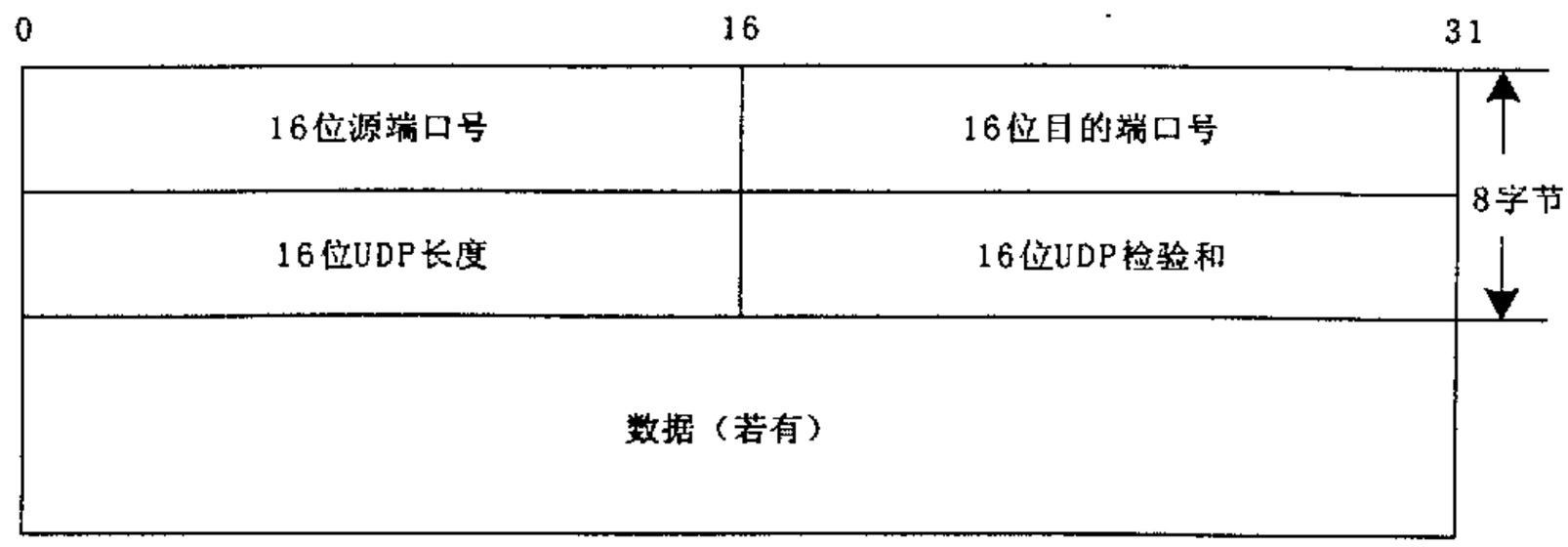

图 5-10 UDP 协议格式

每一个ICMP、TCP、UDP的报文都要放在IP数据包的数据部分中通过互联网，而IP数据包本身放在帧的数据部分中通过物理网络。也就是说这三个协议需要两次封装，如图5-11示。

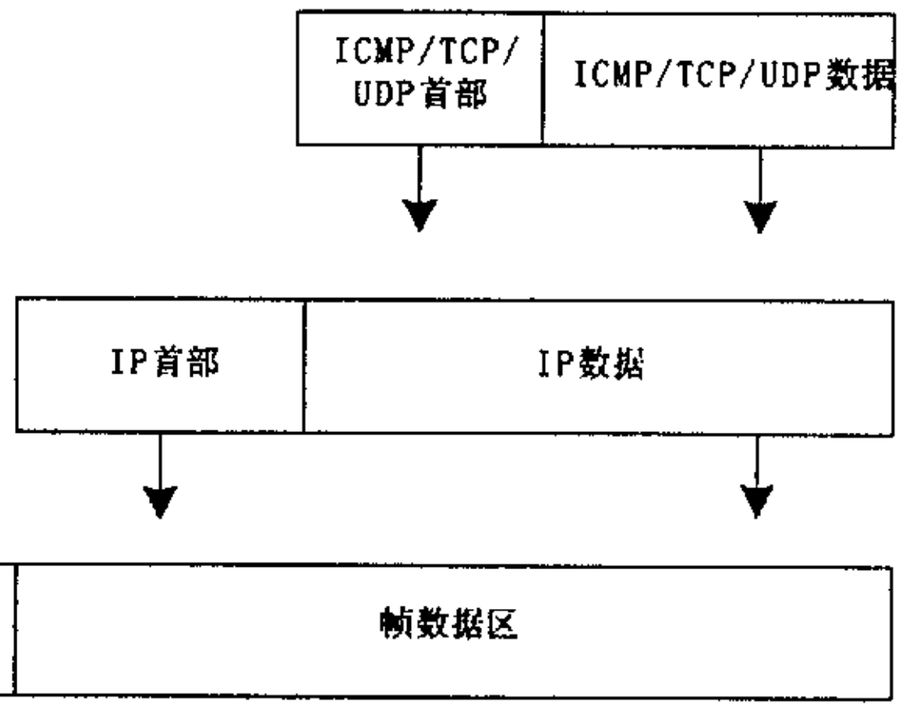

图 5-11 ICMP、TCP、UDP 协议的两次封装

要得到ICMP、TCP、UDP的报文，R_NNIDS系统的解码模块就要进行两次解码，首先要从中得到IP数据包，然后根据IP数据包的首部的协议字段得到不同的上层协议(TCP, UDP，而ICMP属于网络层协议)数据。

5.2.2 模块具体设计

在R_NNIDS系统中，预处理模块主要由NNDecodeProtocol类组成，它从数据采集模块接收数据后，根据数据链路层的类型进行解码，紧接着进行高层协议的解码。其主要的接口函数和数据成员如表5-2所示。

模块名称	类名称	接口函数	说明
预处理模块	NNDecodeProtocol	StartDecode()	程序开始
		DecodeEth()	以太网解码
		DecodeIP()	IP 数据包解码
		DecodeTCP()	TCP 数据包解码
		DecodeUDP()	UDP 数据包解码
		DecodeICMP()	ICMP 数据包解码

表 5-2 预处理模块主要函数

IPFrag类和NNIPFragPreProcess类负责IP分片重组：IPFrag类包含了同一个IP数据包各个分片，同时对这些数据包进行检查及重组，NNIPFragPreProcess类则完成所有数据包的检查与重组操作。IP分片重组主要的接口函数和数据成员如表5-3所示。

模块名称	类名称	接口函数与数据成员	说明
IP 分片重组	IPFrag	IPlist	IP 数据包分片链表
		CheckIntergrity()	完整性检查
		CheckTimeout()	检查超时
		ProcessFrag()	处理分片
		Assemble()	重组本数据包
	NNIPFragPreProcess	PreProcessInit()	初始化重组模块
		PreProcess()	程序开始

表5-3 分片重组模块主要接口函数和数据成员

预处理模块的主要流程图如图5-12所示。

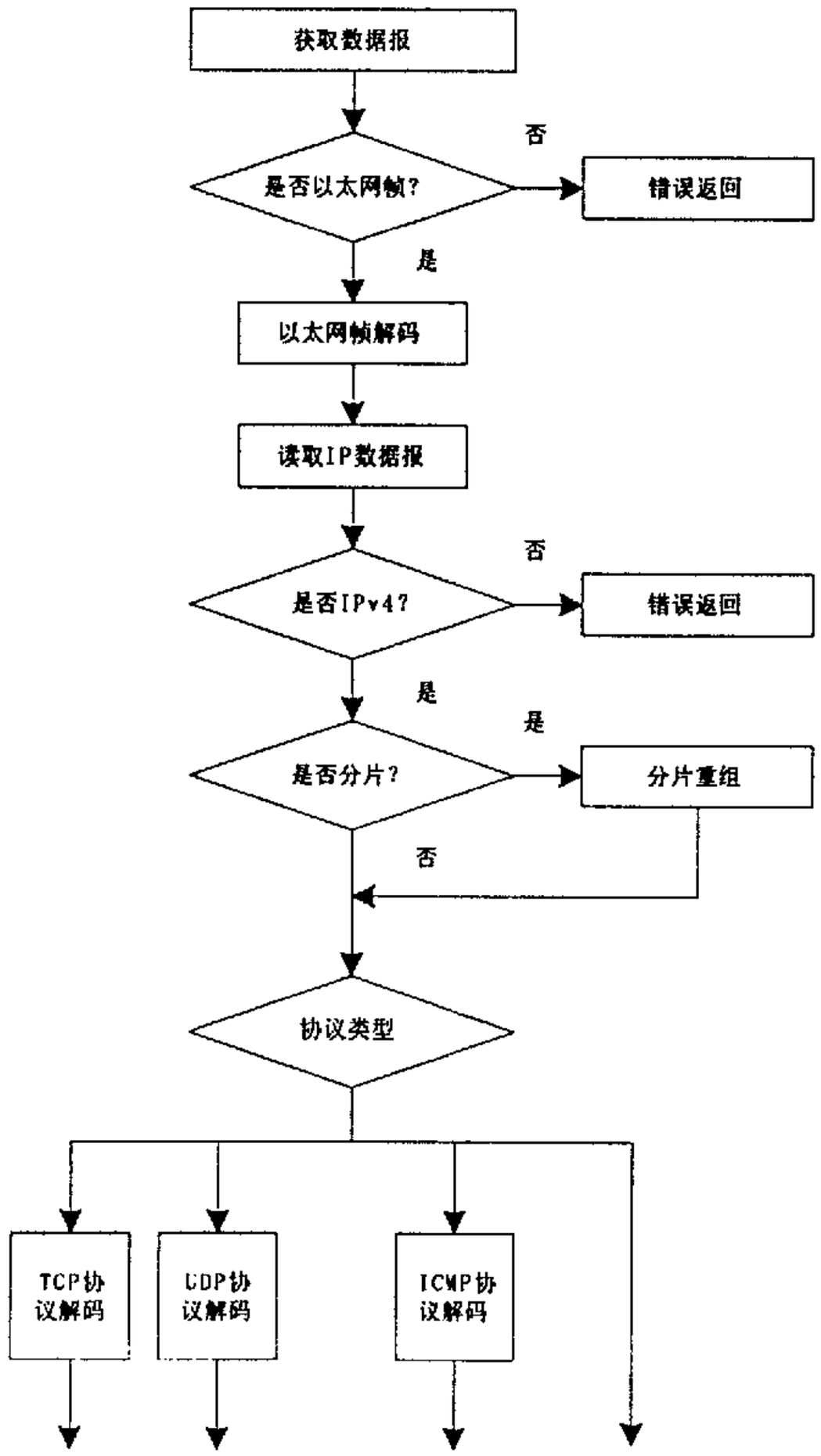

图 5-12 预处理模块程序流程图

5.3 二进制编码模块

二进制编码模块主要有三个功能，首先负责将从预处理模块得到的数据包进行编码，转换成二进制数列，作为综合分类器的输入。其次负责将规则文件中的规则语句编码为二进制数列，用来对综合分类进行训练。第三，将综合分类器中产生的新的规则数列，反编码为规则语句输出。

5.3.1 数据包编码模块

将从预处理模块接收到的数据包转换为具有158个分向量的特征向量，送往智能神经网络进行判别。具体的编码方式如下：

1) IP包：

- 1～32 位二进制数表示源IP地址。
- 33～64 位二进制数表示目的IP 地址。
- 65～66 位固定为“00”，表示传输类型。
- 67～70 位表示首部长度。
- 71～78 位表示服务类型。
- 79～94 位为IP数据包总长度。
- 95～110 位为表示IP标识。
- 111～113 位为IP标志。
- 114～126 位为表示IP片偏移。
- 127～134 位为协议字段。
- 135～158 位全为“0”，以补足158 位，从而与TCP包共用神经网络输入层。

2) TCP包：

- 1～32 位二进制数表示源IP地址。
- 33～64 位二进制数表示目的IP 地址。
- 65～66 位固定为“01”，表示传输类型。
- 67～80 位表示源端口，将端口编为14 位长的二进制数，对于16383以上的不常用端口统一用14 位的“0”表示。
- 81～94 位表示目的端口，同上。
- 95～126 位为Seq位编码。
- 127～158 位为Ack位编码。

3) ICMP包：

- 1～32 位二进制数表示源IP地址。
- 33～64 位二进制数表示目的IP地址。

- 65～66 位固定为“11”，表示传输类型。
- 67～98 位为ID位编码。
- 99～114 位为IP包片偏移的编码。
- 115～126 位为IP包分片大小的编码。
- 127～158 位全为“0”，以补足158位，从而与TCP包共用神经网络输入层。

4)UDP包：

- 1～32 位二进制数表示源IP地址。
- 33～64 位二进制数表示目的IP地址。
- 65～66 位固定为“10”，表示传输类型。
- 67～80 位表示源端口，将端口编为14 位长的二进制数，对于16383以上的不常用端口统一用14位的“0”表示。
- 81～94 位表示目的端口，同上。
- 94～158 位全为“0”，以补足158位，从而与TCP包共用神经网络输入层。

5.3.2 入侵规则语句的编码

1. 规则介绍

R_NNIDS系统采用与snort系统相兼容的规则描述方法。Snort系统是一个跨平台的、轻量级的网络入侵检测系统，是一个开源软件。它使用了一种简单的、轻量级的描述语言来描述网络上带有攻击标识的数据包，具有较强的描述能力。Snort系统中规则描述采用插件机制来管理规则，它将每一类的攻击放入到一个文件当中，系统管理员可以根据需要加载所需要的攻击规则。也可以从文件中添加或删除规则。snort规则针对每一个协议都制定了其检测的内容，其内容的组合就可以构成一个完整的检测规则。下面就规则进行具体说明[29]。

规则都存放在规则文件中，这些规则文件都是普通的文本文件。具体的规则在逻辑上可以分为两部分:规则头(Rule Header)和规则选项(Rule Option)。规则头定义了规则的行为(Action)、所匹配报文的协议、源地址、源端口、目的地址、目的端口以及源地址和目的地址的网络掩码等信息；规则选项则包含了所要显示给用户查看的警告信息以及用来判定些报文是否为攻击报文的其它信息(如tcp的flag字段、数据字段的内容等)。

2. 规则头(Rule Header)

规则头的一般形式是:

<规则行为><协议类型><源IP地址><源端口><目的IP地址><目的端口>

规则头的第一个字段是规则行为(Rule Action)，主要有如下的五种规则行为:

alert: 使用设定的警告方法生成警告信息，并记录这个报文。

log：使用设定的方法记录报文。

pass：忽略这个报文。

activate：进行alert，然后激活另一个dynamic规则。

dynamic：等待被一个activate规则激活，然后进行log。

规则头的下一个域是协议字段(Protocol)，当前支持如下协议:TCP，UDP，ICMP，IP协议。

下面一个字段是源地址、目的地址、源端口和目的端口信息。

地址采用IP地址加上一个CIDR(Classless Inter-Domain Routing)块来表示地址，CIDR块用来说明IP地址的网络掩码，"/24"说明是一个C类地址，"/16"说明是一个B类地址，"/32"指定一个主机，如192.168.57.1/24表示C类网络192.168.57.1-255，地址也可以用一个用户指定的范围来表示，如：

alert tcp 192.168.59.1/24->[192.168.57.1/24 ,192.168.57.100/24]

表示要检测的源IP地址范围是192.168.59.1-255，目的IP地址范围是192.168.57.1-100。

端口号部分可以是一个静态的数字，也可以是一个范围，如1：1024表示要检测的报文的端口范围是1-1024。

地址和端口都可以用关键字“any”来代替，表示任意的地址或端口，也可以在地址或端口前加“！”，表示取反，如在端口号前加“！1：1024”表示要检测的报文端口号范围是除了1-1024的所有其它端口。

3. 规则选项

对规则选项的分析构成检测引擎的核心，所有的规则之间用“:”来分隔，下面就目前支持的规则选项进行分析说明。

ttl:检查IP报文的TTL域的值。

格式:ttl:<数值>

tos:检查IP报文的TOS值。

格式:tos:<数值>

id:检查IP报文分片的ID域的值。

格式:id:<数值>

ipoption:检查IP报文IP选项的值，具体的检查参数如下:

rr: Record Route

eol:列表结束

nop:占位

is:时间戳

see: IP安全选项

Isrr:松散路由

ssrr:严格路由

satid:流标识号

格式:ipopts:<选项>

fragbits:检查IP报文分片比特位的值。具体的检查参数如下:

M:检查MF标志

D:检查DF标志

R:检查RF标志

格式:fragbits:<分片内容>

dsize:检查IP报文负载长度大小。

格式:dsize:[>I<]<number>

flags:检查TCP包的flags域，具体的检查参数如下:

F: FIN

S: SYN

R: RST

P: PSH

A: ACK

U: URG

格式:flags:<标识值>

seq:检查TCP包的序列号域的值。

格式:seq:<序列号值>

ack:检查tcp包的确认域的值。

格式:ack:<确认号值>

itype:检查ICMP协议的类型域的值。

格式:itype:<数值>

icode:检查ICMP协议的代码域值。

格式:icode:<数值>

icmp_id:检查ICMP协议的回应消息的序列号域的值。

格式:icmp_id:<数值>

resp:激活响应，通过发送报文来实现响应，主要的响应方式有:

rst_snd:向发送方发送TCP_RST报文。

Rst_rcv:向接收方发送TCP_RST报文。

Rst_all:向发送方和接收方都发送TCP_RST报文。

Icmp_net:向发送方发送网络不可达的ICMP报文。

Icmp_host:向发送方发送主机不可达的ICMP报文。

Icmp_port:向发送方发送端口不可达的ICMP报文。

Icmp_all:向发送方发送以上所有的ICMP报文。

格式:resp:<响应类型>

在编写具体的规则时，由于每一种入侵都有其特定的特征，可以根据入侵的主要特征，采用以上的规则头和规则选项进行合理的组合，以便能正确的描述出一种入侵。

4. 规则转换

规则转换的具体思路：打开规则文件，提取出某一规则。截取规则头中的<源IP地址><源端口><目的IP地址><目的端口>；根据不同的协议类型，提取规则选项中的相关项，例如：提取TCP中的flags、seq、ack等项。最后按数据包编码方式编码。

同理，从综合分类器得到的新规则，按相反的方式，从二进制数列中得出规则的具体描述。

5.3.3 二进制编码模块的设计

在 R_NNIDS 中，采用 NNBinaryEncode 类来完成，其主要的接口函数如表 5-4 所示：

模块名称	类名称	接口函数	说明
二进制编码模块	NNBinaryEncode	NNIPEncode()	IP 数据编码
		NNTCPEncode()	TCP 数据编码
		NNUDPEncode()	UDP 数据编码
		NNICMPEncode()	ICMP 数据编码
		NNExtractRules()	提取规则
		NNParseProtocol()	协议解析
		NNIPDecode()	IP 数据解码
		NNTCPDecode()	TCP 数据解码
		NNUDPDecode()	UDP 数据解码
		NNICMPDecode()	ICMP 数据解码

表 5-4 二进制编码模块主要接口函数表

二进制编码模块的程序流程图如图 5-13，5-14 所示：

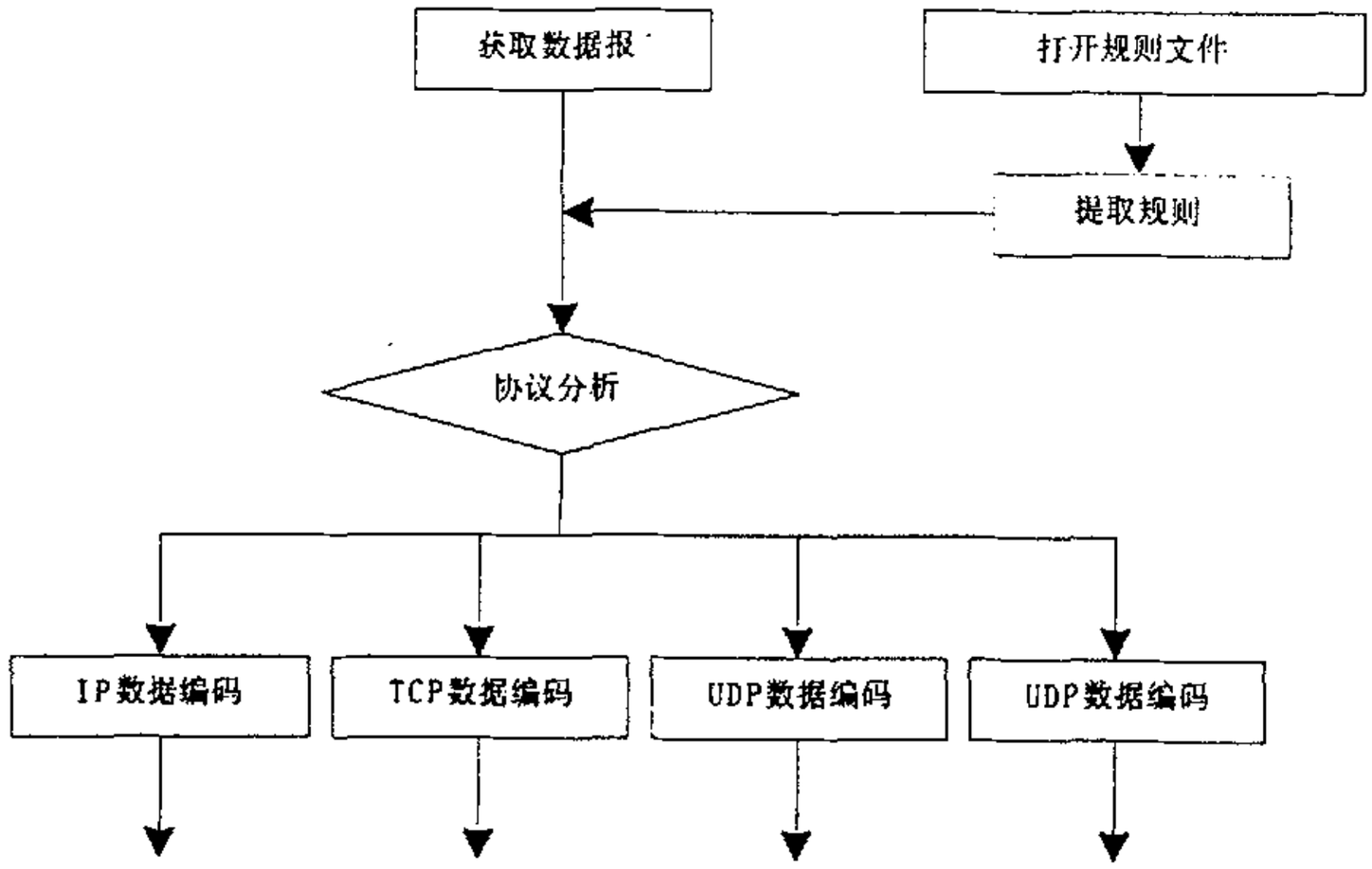

图 5-13 二进制编码程序流程图

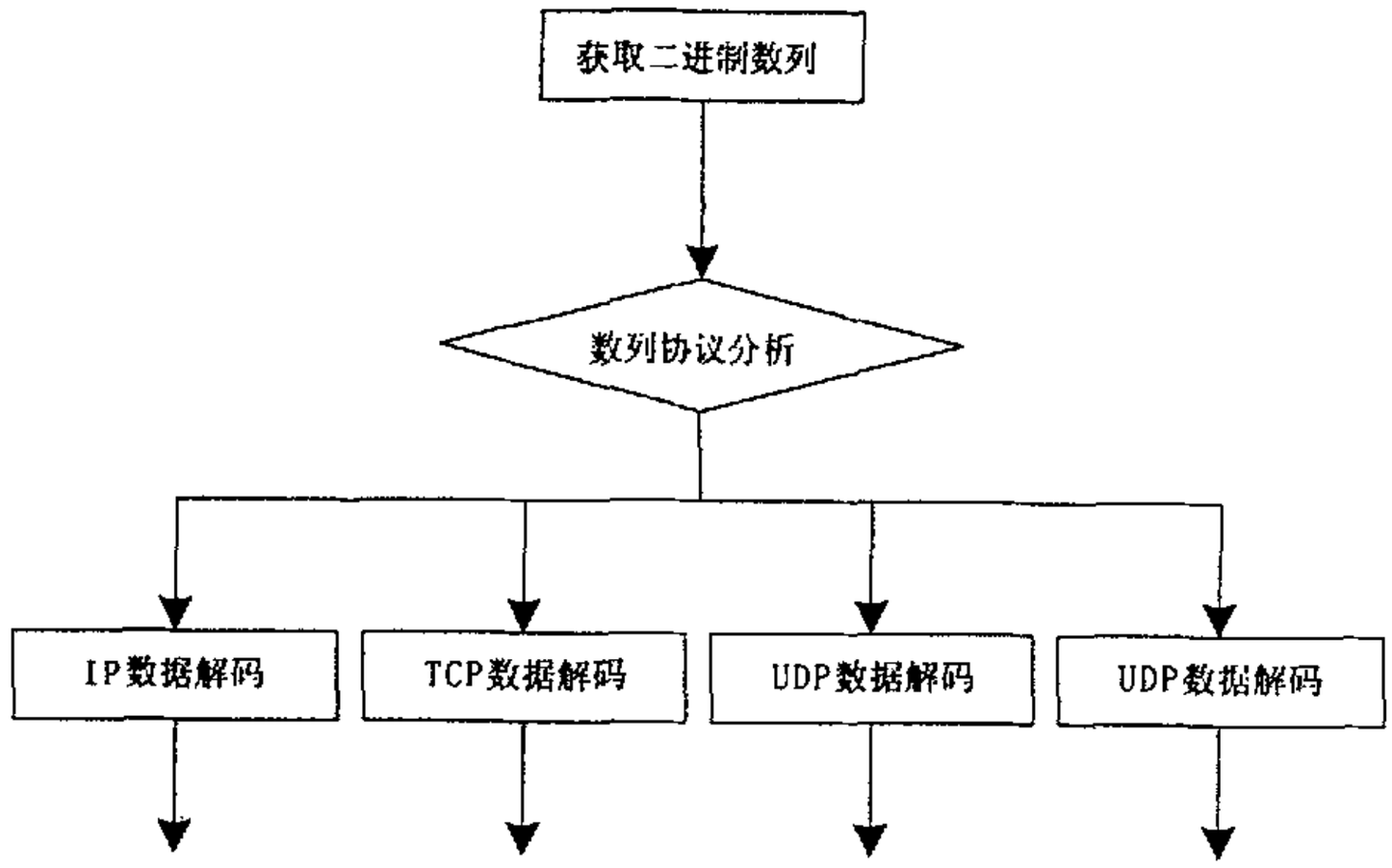

图 5-14 规则解码程序流程图

5.4 综合分类器模块设计与实现

综合分类器负责将输入的二进制数列进行学习和分类，识别出其中的入侵行为。综合分类器和基础分类器都是由基于粗糙集理论的神经网络模型组成。具体的设计过程如下：

5.4.1 分类器结构的初始设计

分类器的设计是入侵检测系统模型设计的重点。现将基础和综合分类器的初始结构设计如下[27]：

1. 1989 年数学家 Robert 证明了只含有一个隐含层 BP 神经网络可以逼近所有

的连续函数。因此在上述模型中基础分类器和综合分类器分别采用三层 BP 神经网络结构 p-q-r 和 P-Q-R。

2. 分类器的传递函数都采用常用的 Sigmoid 函数：$f=\frac{1}{1+e^{-x}}$。

3. 从图 4-3 可以看出，对于综合分类器来说，输入层节点数目由全体基础分类器的输入层节点数目决定，P=max{ $p_1, p_2, p_3 \cdots p_n$}。

4. 分类器的输出节点数目是确定的常数。其中基础分类器输出节点的数目 r 由此方面的攻击类型的数目确定，而综合分类器输出节点的数目 R 为所有攻击类型的数目确定。

5. 对于隐含层节点来说，合适的隐含层节点数目的选取一直是难点和关键。确定一个适当的隐含节点数目不仅能够减少训练时间、加快收敛，而且能够增强网络的泛化能力。在本入侵检测模型中，综合分类器的隐含层节点拥有特定的物理意义，代表基础分类器，所以其数目 Q 是已知的。因此我们只需要确定基础分类器的隐含层节点数目。我们在网络初始时假设基础分类器的隐含层节点数目初始值为 q=p-1，然后根据学习情况再逐步减少，直到最佳节点数。

6. 对于网络中的权值矩阵，基础分类器和综合分类器采用相同的设计。即输入层到隐含层的权值矩阵采用远小于 1 的随机数，而隐含层到输出层的权值矩阵则采用一半为+1，一半为-1 的方法设置。按以上方法设置的初始权值可保证每个神经元一开始都工作在使传递函数变化最大的位置，以加快网络学习速度。

在确定了分类器的初始结构后，仍然需要对其结构进行优化。因为对于确定的样本数，输入层和隐含层的节点数目过多不仅会使网络结构复杂，学习速度变慢，而且可能把噪声等非规律的内容也学会，出现“过度吻合”现象，降低了网络的泛化能力。所以在这里我们利用粗糙集理论能够从大量数据提取规律的特性，对基础分类器的输入层和隐含层进行约简，目的是在保证精度的情况下，用较少的节点来构建分类器。

5.4.2 基于粗糙集理论的结构优化

5.4.2.1粗糙集理论[33,37]

粗糙集(rough sets) 理论最早是由波兰学者Z. Pawlak于1982年提出的。近几年来，粗糙集理论已成为人工智能领域中的一个新学术热点，在机器学习、知识获取、知识发现和决策分析等领域得到了广泛的研究与应用。其主要特点是将知识和分类联系在一起，在保持决策系统分类能力不变的前提下，通过知识约简，减去多余的信息。

定义1 设S=(U, A, V, F)为知识表示系统。这里U为论域，是一非空有限集合，

即U={X1, X2, …Xn}；A是非空有限的属性集合；V是属性值的集合，即$V=\bigcup_{P\in A}V_p$，V_p是属性P∈A的值域。F_p是U×A → F_p 的映射，它为U中各对象的属性指定唯一值。如果A由条件属性集合C和决策属性D组成，并且满足C∪D=A，C∩D=Φ，称S为决策系统，一般的决策系统只有一个决策属性，常记为S=(U, C∪D, V, F)。

定义2 在一个决策系统中，各个条件属性之间往往存在着某些程度上的依赖或关联，约简可以理解为在不丢失信息的前提下，可以最简单地表示决策系统的结论属性对条件属性的集合的依赖和关联。

定义3 给定一个决策系统S=(U, C∪D, V, F)， Hu的改进的区分矩阵M=(m_{ij}) 定义为： m_{ij}={a ∈C: a(xi) ≠a(xj)}，D(xi)≠D(xj)； m_{ij}=Φ，D(xi)=D(xj)。其中a(x)是元组x在属性a上的取值，D(x)是x在决策属性D上的取值。

定义4 设决策表S=(U, C∪D, V, F)中属性a_i有V_i={V_{1i}, V_{2i}, … V_{Ki}}k个不同的属性值。则属性a_i的属性重要性函数$f(a_i)$为： $f(a_i)=V_{1i}V_{2i}+V_{1i}V_{3i}+\cdots+V_{k-2}V_{K-1}+V_{K-1}V_{Ki}$。

5.4.2.2 基于区分矩阵的属性约简算法[34-36]

属性约简是粗糙集理论中的一个重要的研究课题，人们总期望找到最小的约简，但这是一个 NP 完全问题，只能通过启发式信息进行求解。属性约简算法有很多种，在这里采用基于属性重要性的约简算法。

粗糙集中基于属性重要性的启发式算法在利用区分矩阵求约简时，通常分以下三步：首先需要求出区分矩阵。其次是求核。也就是将区分矩阵中包含单一属性的元素并起来。第三是求约简，对于非核属性，将其按属性的重要性从大到小排序，属性的重要性由定义 4 所定义的属性重要性函数$f(a_i)$的函数值来确定，值越大属性越重要。取最重要的属性将其加入核中，并删除包含此属性的所有节点。一直按此处理直到区分矩阵为空，这时所得到的集合就为所求。具体步骤如下：

输入决策表 S=(U, C∪D, V, F)。

输出约简表 S'=(U, R∪D, V', F')。其中 R 为 C 相对于 D 的约简。

步骤 1：根据定义 3 求出区分矩阵 M。

步骤 2：求 C 相对于 D 的核$CORE_D(C)$。设$CORE_D(C)$=Φ，按属性个数由小到大对 M 中属性组合排序。查询 M 中各个元素，若为单一属性组合则将该属性加入$CORE_{(D)}$，并从 M 中删除该属性。所有包含单一属性的元素的集合就是所求。

步骤 3： 设 R=$CORE_D(C)$。根据定义 4 计算属性重要性函数$f(a_i)$，取函数值最大的属性加入 R。并从 M 中删除所有的包含此属性的元素。依次取函数值最大的属性，直到 M 为 Φ，此时 R 就为所求。

步骤4 S'=(U, R∪D, V', F')即为属性约简后的决策表，其中R为C的相对于D的约简，V'、F'由V、F 去除部分属性而来。

5.4.2.3 基础分类器的结构优化

基于粗糙集约简算法的结构优化如图5-15所示：

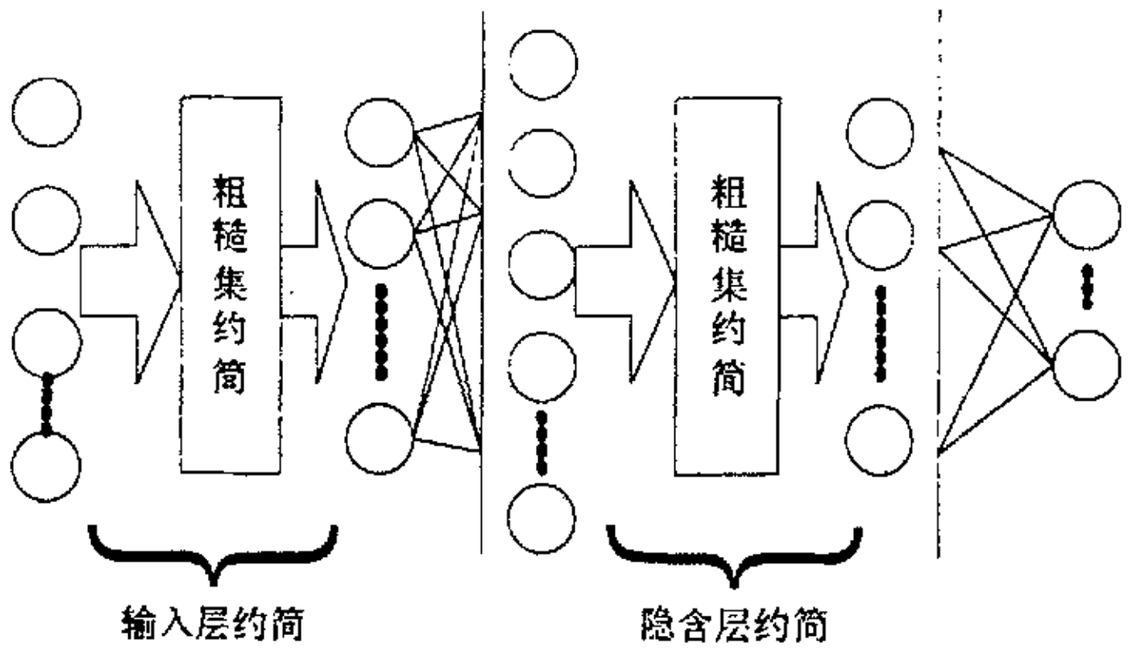

图 5-15 基础分类器模型优化

1. 输入层节点优化。根据BP神经网络的学习样本建立信息表S1，信息表的列表示属性，行表示数据值，此时信息表的所有属性就代表输入层节点。我们采用上述的启发式算法对信息表S1进行约简，从而确定输入层节点数目p。

2. 隐含层节点优化。根据输入层节点数目p，建立神经网络p-q-r，其中q初始值为p-1。使用少量的样本进行学习，学习算法采用Levenberg-Marquardt（LM）算法。学习完毕后，根据隐含节点的输出建立信息表S2，隐含节点到输出节点的权重建立信息表S3。因为隐含层的输出是输入层到隐含层的连接、权重等知识的外在体现，而隐含层跟输出层的关系体现在隐含层到输出层的权重上，所以合适的隐含层节点数目可由隐含层输出和隐含层到输出层的权重来确定。我们采用上述的启发式算法对信息表S2，S3进行约简。q=max{S2的属性个数，S3的属性个数}，从而确定隐含层节点数目q。

完成所有的基础分类器结构优化后，通过5.4.1理论我们就可以得出综合分类器的整体结构。输入层节点数目=max{p_i，i=1,2,...,n}，隐含层节点数目Q=n，输出层节点数目=所有攻击类型的数目。

5.4.3 分类器的学习

接下来，我们取经过转化的规则数据库中的攻击样本对分类器进行训练，分类器的训练步骤如下：

1. 基础分类器的学习

分别对不同的基础分类器进行训练。取一定数量的针对此分类器的攻击样本，对于每个样本，除了样本所对应的输出层节点的输出为“1”外，其余的输出层节点均输出“0”。学习算法采用LM算法。学习完毕后，保存权值。

2. 综合分类器的学习

完成基础分类器的训练后，保持基础分类器内部不变，将其并入综合分类器中，开始对整体进行训练。在学习过程中，综合分类器的权值的调整只发生在每个基础

分类器的输出节点到综合分类器的输出节点的连接权值上，调整算法采用LM算法。学习完毕后，保存权值。

3. 建立网络结构的综合评判函数

根据神经网络的学习精度A、收敛时间B和节点数目C建立综合评价函数：$f=\alpha A+\beta B+\Gamma c$，其中α、β、Γ为权重，A={(1-0.7)好，(0.3-0.7)中，(0-0.3)差}，B={(1-0.7)快，(0.3-0.7)中，(0-0.3)慢}，C={(1-0.7)多，(0.3-0.7)中，(0-0.3)少}。

4. 对学习完毕的分类器使用综合评价函数来评定。对于达不到要求的神经网络，重新提取信息表S1、S2、S3进行约简，形成一个新的分类器结构。依次进行，直到系统评价达到要求为止。

5.5.4 模型实现中使用的Matlab函数[38.39]

我们采用Matlab引擎来实现综合分类器模型，主要是因为Matlab长于数值计算，能够处理大量数据，而且效率比较高，在神经网络处理方面有一定的优势。在程序中我们使用VC++实现用户界面和端口操作，通过Matlab函数来使用Matlab的计算功能。应用的主要函数如下：

5.5.4.1 建立BP神经网络使用的函数

1. net=init(net)

该函数会根据网络的初始化函数以及它的参数值来设置网络权值和阈值的初始值，它们分别由参数net.initFcn和net.initParam表示。

2. net=newff(PR,[S1 S2 … SN],{TF1 TF2 … TFN},BTF,BLF,PF)

该函数用于创建一个N层的BP神经网络。

PR：输入向量的取值范围；

Si：第i层的神经元个数，共N层；

Tfi：第i层的传递函数，可以是任何可微函数，比如对数S型函数（logsig）、线形函数purelin和正切S型函数（tansig），缺省值为“tansig”；

BTF：BP网络训练函数，包括批梯度下降训练函数（traingd）、自适应修改学习率算法(traingda,traingdx)、有弹回的BP算法(trainrp)、比例共轭梯度算法(trainscg)、Levenberg-Marquardt算法（trainlm）等不同算法，缺省值为“trainlm”。

BLF：BP网络权值和阈值学习函数，包括learngd、learngdm。缺省值为“learngdm”；

PF：性能函数，包括均方误差（mse）、msereg和sse，缺省值为“mse”。

3. Y=sim(net,p)

神经网络仿真函数。参数 net 为神经网络名，p 为网络的输入，Y 为网络的输出。

4. [net, tr]=train(NET, p, t)

神经网络训练函数。参数 NET 为神经网络名，p 为网络的输入，t 为网络的目标输出。返回值 net 为训练后的神经网络名，tr 为训练的记录(包括训练步数(epoch)和性能（perf））。

NET.trainParam.show：两次更新显示间的训练次数。

NET.trainParam.epochs：网络训练次数。

NET.trainParam.goal：网络训练误差指标。

5.5.4.2 在 VC++中调用 Matlab 引擎使用的函数

1. Engine *engOpen(const char *startcmd)

该函数启动 Matlab 引擎。参数 startcmd 为启动 Matlab 进程的字符串，在 Windows 环境下，该字符串为 NULL。返回一个指向引擎句柄的指针。

2. int engClose(Engine *ep)

该函数用来关闭 Matlab 引擎，返回 0 表示成功，返回 1 表示关闭失败。

3. int engEvalString(Engine *ep, const char *string)

该函数向 Matlab 发送一个字符串，让 Matlab 执行。参数 ep 为事先启动的引擎指针。String 为执行的字符串。返回 0 表示成功，返回 1，表示引擎 ep 已经关闭。

4. mxArray *engGetArray(Engine *ep, const char *name)

该函数从 Matlab 引擎中获得一个矩阵。参数 name 表示希望从 Matlab 中获得的矩阵的名称。如果函数调用成功，则返回一个 mxArray 类型的指针；如果失败，则返回 NULL。

5. int engPutArray(Engine *ep, const mxArray *mp)

该函数把 mp 指向的矩阵写入 Matlab 引擎中。参数 mp 为数组指针。如果引擎中变量不存在，则创建该变量；如果引擎中变量已经存在，则覆盖该变量。返回 0 表示成功，出错返回 1。

6. mxArray *mxCreateDoubleMatrix(int m, int n, mxComplexity ComplexFlag)

该函数创建一个二维的 double 型的 Matlab 阵列。参数 m 表示阵列的行数，n 表示阵列的列数。ComplexFlag 为常数，取 mxREAL 和 mxCOMPLEX 分别表示阵列中的数据是实数还是复数。函数如果调用成功，则返回一个指向数据结构 mxArray 的指针，否则返回 NULL。

7. void mxDestroyArray(mxArray *array_ptr)

该函数用来删除数据结构 mxArray。

5.5.5 模块的具体设计

在R_NNIDS中，NNRoughSet类用于实现粗糙集的约简，NNClassifier类用于实现分类器模型。例如，对于IP基础分类器的实现算法表示如下。

5.5.5.1 粗糙集约简算法

函数RoughSetReduce的实现如下：

1) function result=RoughSetReduce(DataInput) //Matlab中的函数文件
U=DataInput; //获取要约简的数据变量，DataInput为数据变量名称
i = 1;
k = card (U);
M =Φ;
while 1≤i≤k
for j = i+1 : k
if $D(x_i)D(x_j)$
m_{ij} = {a∈C: $a(x_i)\neq a(x_j)$};
else
m_{ij} = Φ;
M = M∪{m_{ij}};
end
end
end

2) 按属性个数由小到大对M 中属性组合排序。

3) i = 1;
CORED(C) =Φ;

从头取M 中元素，若为单一属性组合则将该属性加入从头取M 中元素，若为单一属性组合则将该属性加入CORED(C)，并从M中删除该属性组合。

4) i = 1;
k = card(A);
while 1≤i≤k 取M中第i个元素m_i

if $m_i\subseteq m_j$ (j= i+ 1, i+ 2, . . . , k)
M = M-{m_j};
end
i = i+ 1;
end

5) 设出现在M 中的属性的集合S={a_1, a_2, . . . , a_n n≤card(C)},w(ai)为M中包含属性a_i的属性组合个数。

6) i = 1;
R = CORED(C);
Q = {m_i|$m_i\cap R\neq\Phi$, i= 1, 2, . . . , card(M)};

```
Repeat//重复
    M = M-Q;
    取S中的元素a_j使得w(a_j)= max_{1≤i≤card(S)} {w(a_i)};
R = R∪{a_j};
S = S-{a_j};
Q = Q∪{m_p∈M|a_p∈m_i, p=1, 2, . . . , card(M)}
for all m_n∈M
    if a_j∈m_p
      for all a_k∈m_p
         w(a_k): = w(a_k)-1;
      end
    end
  end
  i =i+ 1;
  until M =Φ
7) result=R;                                    //result为输出结果
```

5.5.5.1 IP基础分类器训练算法

函数IPClassifier用于实现IP基础分类器的构建和学习。

```
IPClassifier()
{
    Engine *ep;
    if (!ep=engOpen("\0"))
    {                          //打开Matlab引擎,建立与本地Matlab的连接
        fprintf(stderr,"\n Can't start MATLAB engine\n");
        exit(-1);
    }
    engEvalString(ep,"load IPinput.txt");
    engEvalString(ep,"result=RoughSetReduce (IPinput) ");
    engEvalString(ep,"save IPinput_new.txt result -ascii");
    engEvalString(ep,"load IPinput_new.txt");
    engEvalString(ep,"load IPinput_new.txt");
                //读入输入矩阵,其中文件IPinput_new由粗糙集约简后输出
    engEvalString(ep,"net=netff([0 1],[inputnum,157,8])");
                                //inputnum为约简后的输入层节点
    engEvalString(ep,"net=train(net,IPinput_new,T)");
    ……                          //省略部分代码
    engEvalString(ep, "close");
    engClose(ep);
}
```

5.5.5.3 主要的接口函数和成员变量

主要的接口函数和成员变量如表5-5所示：

模块名称	类名称	接口函数	说明
综合分类器模块	NNClassifier	IntegratedClassifier()	建立综合分类器
		IPClassifier()	IP基础分类器
		ICMPClassifier()	ICMP基础分类器
		TCPClassifier()	TCP基础分类器
		UDPClassifier()	UDP基础分类器
		BPInputArray	输入变量（Matlab数组）
		BPOutputArray	输出变量（Matlab数组）
		BPSetParameter()	设置各种参数,初始化
	NNRoughSet	RoughSetReduce()	粗糙集约简

表 5-5 综合分类器模块主要接口函数和成员变量表

综合分类器模块的程序流程图如图5-16所示：

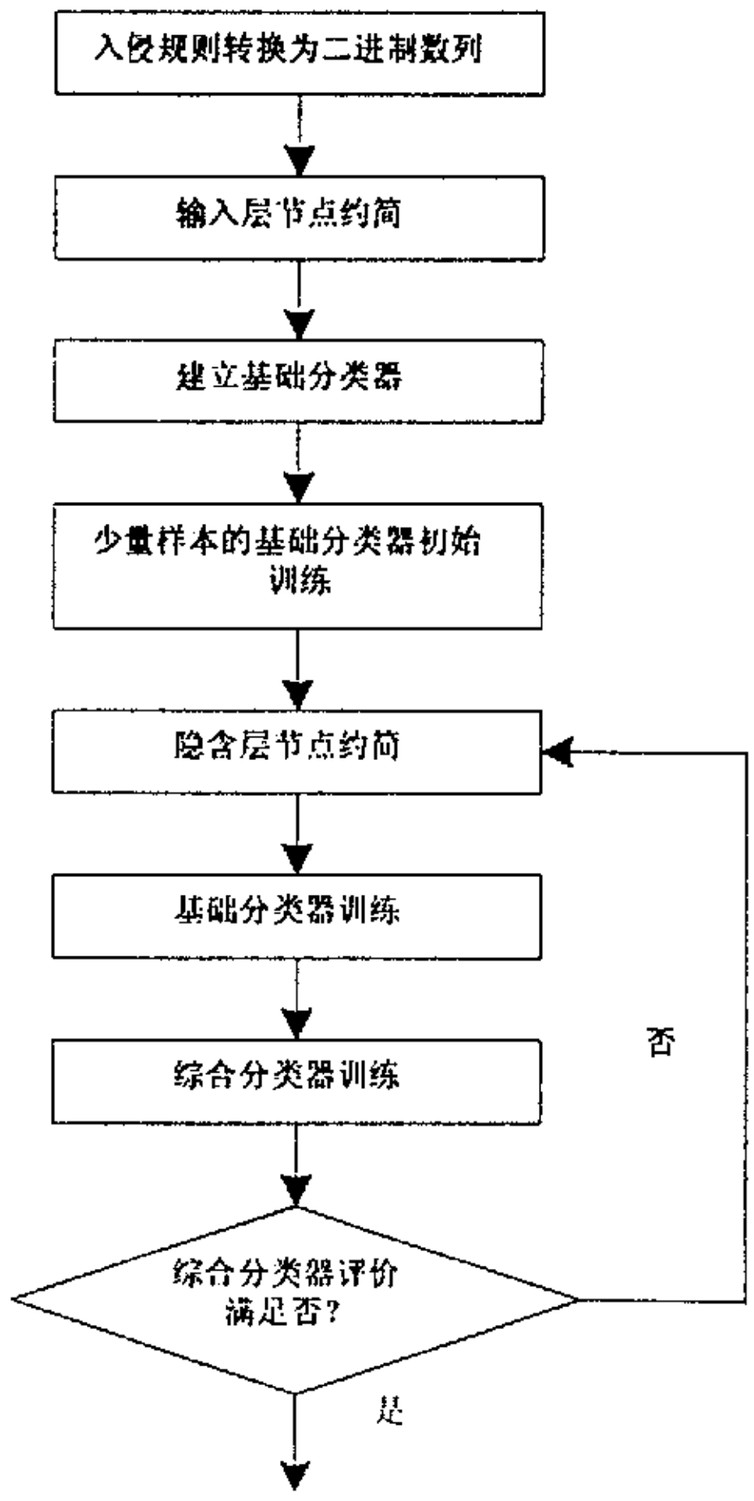

图5-16 综合分类器学习程序流程图

5.5 响应模块的设计与实现

5.5.1 常用响应技术

一、撤消连接

我们所说的连接，通常是指TCP连接。攻击者是通过一个激活的端口进行连接的。他向该端口发送一个或数个包，其中含有攻击字符串或可开发该端口的程序。对一个有脆弱性的系统来说，这是一个十分危险的时刻。安全监测系统检测到攻击字符串后，命令防火墙撤消连接。这时，被攻击的计算机正运行着超出命令长度的代码，而且可能是以超级用户的级别运行。如果它是个能捕捉异常分支类型的程序(在某个预定的端口运行的远程登录后台程序)，撤消连接也许只能为你赢得几秒种的时间。

二、回避

此处所说的回避响应，通常是针对于UDP协议。下面用回避技术继续讨论上面所说的攻击，然后说明它为什么是可供我们使用的最为重要的自动和人工技术之一。

随着攻击的进行，会有一个新的进程以超级用户级别运行，它打开一个远程登录后台进程或发回一个X_Window或攻击者所选择的任何一种可进入我们系统的后门口。此时，撤消连接帮不了我们，因为他已经做好另一个连接的准备，或者在X_Window的情况下，我们已经从自己这方对他发起了连接。这时就用到了回避技术，在正确实施的回避技术中，我们不传递任何来自攻击者或发向攻击者的包。在实施回避技术时，应该建立一个“不回避”文件，将客户和供应商的地址保存在该文件中，即不回避这些地址。这样就能防止某些攻击者假冒这些地址发出一些明显的攻击。

如果攻击者使用两个地址系列，这也是常见的，回避技术就不起作用了。如X_Window的DNS缓冲区溢出攻击，这种协同攻击正是如此：攻击来自一台主机，窗口显示信息被发送给另一台主机。然而，不要仅仅因为回避技术不能在任何情况下都起作用，就不使用这一技术。

三、隔离

隔离也是一种自动响应技术。其思想是:如果在某一时间段发生了足够多的攻击，系统就向类似的逻辑控制继电器发送命令，将路由器的电源断掉，这样做的结果是使该网络节点从Internet中隔离出来。尽管很可能发生拒绝服务的情形，但对于需要高度安全的网络节点，这也是个合理的策略。

四、SYN/ACK响应

假设系统已知某个网络节点用防火墙或过滤路由器对某些端口进行防守，当系统检测到向这些端口发送的TCP的SYN包后，就用一个伪造的ACK进行回答。这样的话，攻击者就会以为他们找到了许多潜在的攻击目标，而实际上他们得到的只不过是一些误报警。那些最新一代的扫描工具以其诱骗功能给入侵检测这一行带来很多问题和麻烦，而SYN/ACK响应正是回击它们的最好办法。

五、RESET（重启）

对使用这一技术应该持慎重的保留态度。RESET可能会断开其他人的TCP连接，曾经有过商用入侵检测系统基于误报警就做出了RESET响应。这种响应的思想是如果你发现一个TCP连接被建立，而它连接的是你要保护的某种东西，就伪造一个RESET并将其发送给发起连接的主机，使连接断开。尽管在商用入侵检测系统中很可能得到这一响应功能，但它不是经常被用到。一旦与误报警联系在一起，这个技术就变得很有意思。另外如果这个技术流行起来，那些攻击者可能很快就会修补他们的TCP程序使其忽略RESET信号。当然，还有一种方式是向内部发送RESET。

5.5.2 系统中的响应措施

系统中响应的方式采用主动和被动相结合的方式:即对于一些入侵等级不高的入侵行为记录系统日志，在屏幕上显示报警信息，通知管理员；而当入侵等级达到一个阈值之后，除了采取以上措施之外，要采取一定的主动措施，如采用前面所说的RESET技术，我们可以在截获了某用户的对网内主机的非法访问后，根据截获的TCP包的序号重新构建一个伪装成从该主机发送的TCP包，其中TCP包中的RST标志设为1，将它发送到网上，以前建立的连接将被断开，所以就实现了阻止非法用户访问网内主机的功能。

报警的频率是另外一个需要考虑的问题。如果对报警频率和数量不作限制，将会对运行监测系统的主机的内存和硬盘空间的资源造成很大影响，从而可能会影响系统的整体性能。系统中采用限制某类攻击在一定时间内的报警次数，来限制报警的频率。

5.5.3 模块的具体设计

响应模块主要是由NNRespond类组成，主要的接口函数如表5-6所示。

模块名称	类名称	接口函数	说明
响应模块	NNRespond	PreProcessInit()	初始化
		ProProcess()	接口调用函数
		SendTCPRST()	发送 TCP RST 报文
		SendAlart()	发送报警信息

表 5-6 响应模块主要接口函数

5.6 小结

本章主要介绍了组成入侵检测系统的五个模块：数据采集模块、预处理模块、二进制转换模块、综合分类器模块、响应模块的具体的设计和实现。其中在综合分类器中引入了粗糙集算法对神经网络结构进行了优化。

第六章　智能入侵检测系统的性能测试

6.1 性能测试环境

在测试评估 IDS 时，很少把 IDS 放在实际运行的网络中，因为实际网络环境是不可控的，并且实际网络环境的专用性也太强，很难对 IDS 进行准确的系统测试，所以我们构建了专门的网络环境来模拟实际的网络，进行 IDS 的性能测试。目前，用于网络流量仿真的工具很多，如 nidsbench、Tcl-DP 和 AsIs 等工具。我们选用了 Anzen 公司开发的 nidsbench。 nidsbench 包括 tcpreplay 和 fragrouter 两部分。tcpreplay 的功能是将 tcpdump 复制的数据分组重放，还原网络的实际运行状态；而 fragrouter 则可以拦截、修改、重写和重排发往特定机器的数据包，几乎可以完全控制数据包的发送方式，满足我们所需的各种攻击。测试环境如图 6-1 所示：

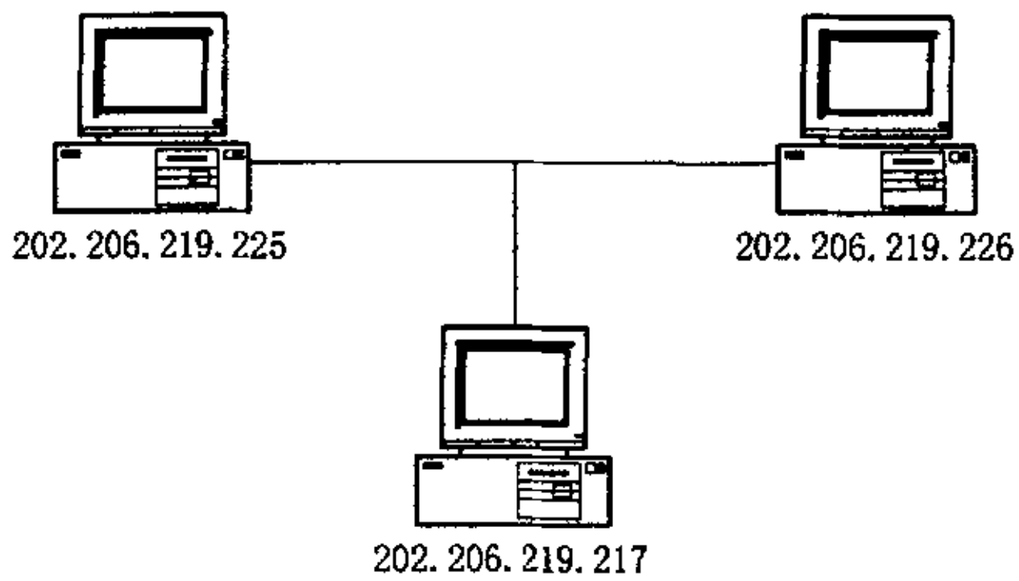

图 6-1 测试环境

测试环境中包括三台主机，其中 202.206.219.225 作为被攻击主机，安装了入侵检测系统，并且预先配置了一些有漏洞的应用系统，如 apache，在 apache 的 cgi-bin 目录中故意放了几个有漏洞的脚本，来模拟被攻击的网络和应用环境；202.206.219.226 作为攻击主机，安装了一些常用的攻击工具，如 CGI 扫描器、DdoS 工具、冰河陷阱等来模拟攻击。202.206.219.217 作为仿真攻击主机，安装了仿真软件 nidsbench，做仿真攻击和辅助测试主机。

6.2 分类器优化和学习

我们首先从规则库中提取样本完成各个基础分类器的优化，具体的过程如下：

1. 输入层节点优化。在规则库中提取同此基础分类器相关的攻击样本各100条，组成信息表S_{1i}(i=1,2,3,4)，利用函数RoughSetReduce分别对信息表S_{1i}进行约

简，约简后的信息表的属性数目就为所求。

2. 隐含层节点优化。完成输入层节点优化后，我们就可以建立初步的基础分类器模型。各取30个攻击样本对基础分类器进行初步训练，初步训练完毕后，提取出隐含层的输出信息表S_{2i}、权重表S_{3i}，对其进行约简，得出隐含层节点数目。

四个分类器的优化结果如表6-1所示：

	IP基础分类器	TCP基础分类器	UDP基础分类器	ICMP基础分类器
分类器结构	73-10-9	78-8-18	68-9-8	70-11-12

表 6-1 基础分类器优化结果

根据 5.4.1 理论，我们可以由基础分类器得出综合分类器的结构：78-4-44。

完成分类器的结构优化后，开始对分类器进行学习，首先对各个基础分类器进行学习：分别采取 8 种、17 种、7 种、11 种每种各 100 条攻击样本对基础分类器进行学习；接着采用 43 种 4300 条攻击样本完成对综合分类器的学习。

6.3 入侵检测系统的性能测试

学习完毕以后，我们将综合分类器并入入侵检测系统中，就可以对系统进行性能测试了。系统的性能测试包括四个单元：第一单元进行系统运行性能测试；第二单元进行常规攻击测试，攻击主机利用模拟攻击工具进行攻击测试；第三单元是仿真流量攻击测试，启动 tcpreplay 进行网络流量仿真，攻击主机利用模拟攻击工具进行测试；第四单元是干扰攻击测试，启动 fragrouter 干扰攻击，其他与第三单元一样。在系统性能测试中我们引入 5.4.3 提到的综合评价函数，对于不满足需要的综合分类器重新按照 5.4.3 中提到的步骤进行学习，反复几次，就会达到一个比较理想的结果。

系统性能测试最后的结果如图 6-2，图 6-3 所示：

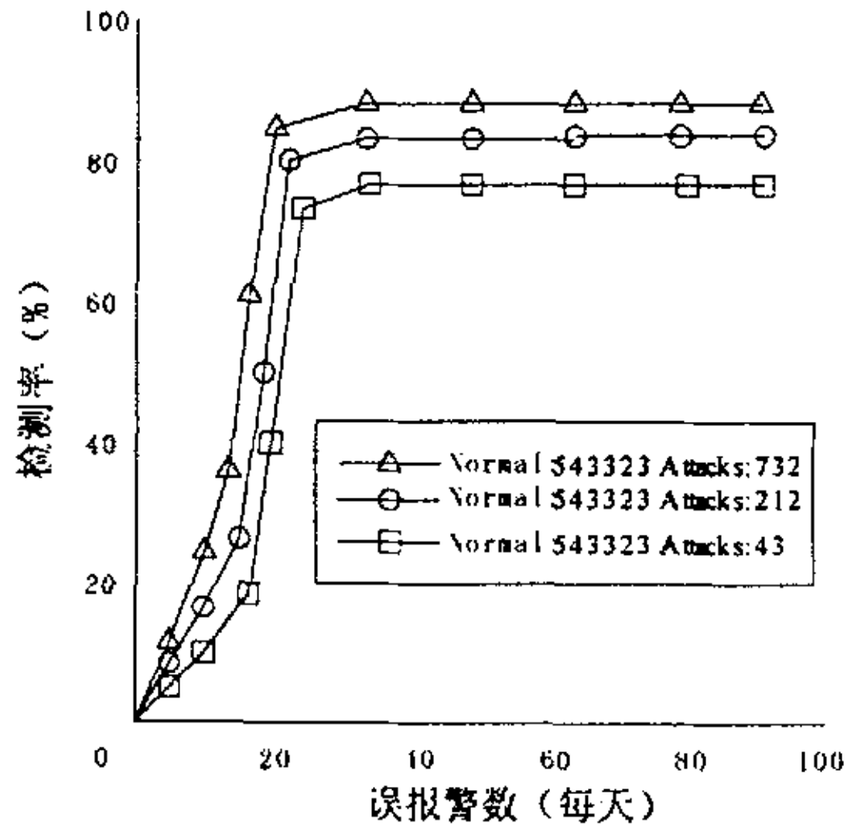

图 6-2 网络流量一定情况下的 ROC 曲线图

图 6-2 是在流量一定的情况下 ROC 曲线图，反映了 R_NNIDS 的误报率与漏报率的关系。工具 tcpreplay 每次产生 543323 个正常的数据包文，攻击主机发出 43、212 和 732 个攻击数据包文，我们得到三组测试数据，如图 6-2。随着攻击报文的增多，攻击报文占整个网络流量的比例上升，R_NNIDS 的检测率也从 73.5%增加到 88.1%，漏报率下降了 14.6 个百分点，但是误报率也有一定的上升。

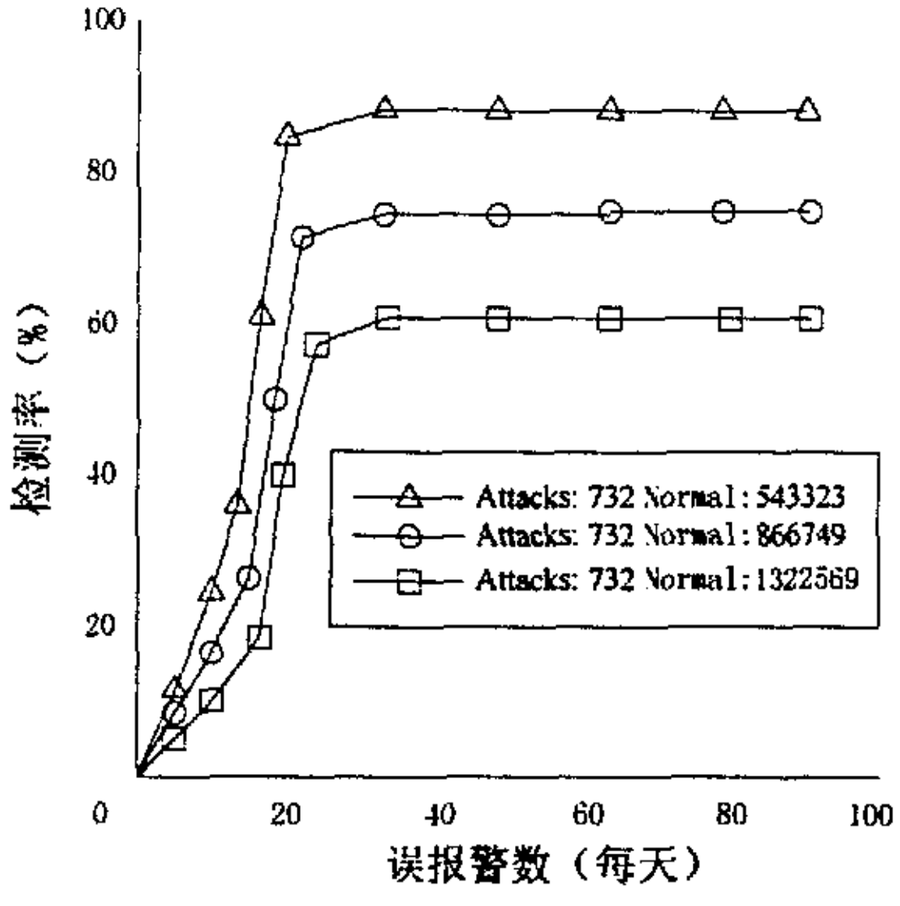

图 6-3 网络流量变化情况下的 ROC 曲线图

图 6-3 是在流量变化的情况下 ROC 曲线图，反映了 R_NNIDS 的检测率与流量负载的关系。攻击主机每次发出 732 个攻击数据包文，但 tcpreplay 分别发出 543323、866749 和 1322569 三组数据包文，每次产生的正常的数据包文明显增加，随着流量负载的增大，R_NNIDS 的检测率下降比较快，从 88.1%下降到 60.2%，漏报率下降了 27.9 个百分点，误报率也有很大幅度的下降。从图 6-3 可以看出，R_NNIDS 受流量负载的影响比较大，随着流量负载的加大，漏报的情况比较严重。

经过一段时间的运行，我们发现规则数据库中产生 14 条新规则，其中属于 IP 包规则 2 条，属于 TCP 包规则 8 条，属于 UDP 包规则 4 条，属于 ICMP 包规则 0 条。这证明了 R_NNIDS 系统能够发现一些新的入侵行为，有一定智能特点。

6.4 小结

本章阐述了智能入侵检测系统的实验测试过程，主要包括系统测试的实验环境，综合分类器的学习情况，以及系统整体测试等三个方面。

第七章 结束语

入侵检测系统(Intrusion Detection System, IDS)作为一种主动的信息安全保障措施，有效地弥补了传统安全防护技术的缺陷。它通过构建动态的安全循环，最大限度地提高系统的安全保障能力，减少安全威胁对系统造成的危害。另外，随着攻击手段的复杂化和多样化，传统的入侵检测方法已不能满足安全需求，智能入侵检测已逐渐成为入侵检测乃至整个网络安全领域的研究重点。好的入侵检测系统不但能及时准确地发现网络上的攻击行为，而且能实时有效地作出响应。它是计算机和网络，安全运行的基础和保证，也是关系到国计民生的大事。

7.1 本文做出的主要工作

本文采用神经网络模型来实现系统的智能化检测，同时为了加快神经网络的学习收敛速度，引入了能够对大量数据进行有效约简的粗糙集理论，将这两种技术融合，最后设计实现了一个新的智能入侵检测系统R_NNIDS。

R_NNIDS 系统包括数据采集模块、预处理模块、二进制转换模块、综合分类器模块、响应模块等五个模块。在 R_NNIDS 系统中除神经网络引擎采用 Matlab 设计以外，其他部分都采用 visual C++来设计，主要的技术特点包括：

1. 数据采集模块采用 Windows 下 Winpacp 库来捕获局域网中数据包。

2. 阐述了 ICMP, IP, TCP, UDP 四种协议的协议格式，采用双层解码方式提取原始数据包的信息。

3. 利用二进制编码模块将数据转换为二进制数列，作为神经网络输入。

4. R_NNIDS 系统采用与 snort 相兼容的规则描述方法，它使用了一种简单的、轻量级的描述语言来描述网络上带有攻击标识的数据包，具有较强的描述能力。

5. 利用粗糙集约简算法对神经网络中的输入和隐含层节点进行优化，加快神经网络收敛速度。

6. 利用 Matlab 程序语言实现神经网络引擎程序。

实验测试表明，本系统能够完成基本的入侵检测，有一定的实用性；能够从原始数据中发现新的规则，有一定的智能性。

7.2 进一步的工作

由于时间关系，本文在智能化理论和系统实现中，都有着一些的缺陷，需要进一步

的完善和改进。

1. 在神经网络隐含层的结构优化过程中，由于要以隐含层的输出结果和隐含层到输出层的权重为基础，所以首先要进行一定的样本训练，之后才能进行约简，确定节点数目。如果样本较多，则学习花费的时间较长，从而会影响网络整体的学习时间，失去了应用价值；当训练的样本减少时，学习花费的时间就会减少。但是较少的样本不能很好地描述隐含层节点之间的信息，这样进行约简，又会容易失真。如何平衡粗糙集约简和神经网络学习在隐含层结构优化中的比例是下一步解决的重点。

2. 从实验测试可以看出，系统的性能会随着网络流量的增大而快速恶化，尤其对百兆以上的流量，系统很难应付。可以预见，随着网络流量的进一步加大，对入侵检测系统将提出更大的挑战，在PC机上运行的纯软件系统的方式需要进一步的研究。

3. 对入侵检测系统自身的攻击和其它安全系统一样，IDS系统本身也往往存在安全漏洞。若对IDS攻击成功，则直接导致其报警失灵，入侵者在其后所作的攻击行为也将无法被记录。所以，系统自身的安全，也是一个系统的需要解决的重点。

参 考 文 献

[1] Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning about Data[M]. Dordrecht The Northland; Kluwer Academic Publishers, 1991

[2] Jelonek J. Rough set reduction of attributes and their domains for neural networks[J]. Computational Intelligence, 1995, 11 (2) : 339～347

[3] Heady R, Luger G, Maccabe A, et al. The architecture of a network level intrusion detection system[M]. Department of Computer Science, University of New Mexico, 1990

[4] Denning D E. An intrusion-detection model[J]. IEEE Transactions on Software Engineering, 1987, 13(2): 222-232

[5] Hofmeyr S A. An immunological model of distributed detection and its application to computer security[M]. University of New Mexico, 1999

[6] Debar H, Dacier M, Wespi A. Towards taxonomy of intrusion detection systems [J]. Computer Networks, 1999, 31(8): 805 - 822

[7] Spafford E H, Zamboni D. Intrusion detection using autonomous agents[J]. *Computer Networks*, 2000, 34(4): 547- 570

[8] Feiertag R, Rho S, Benzinger L, et al. Intrusion detection inter-component adaptive negotiation [J]. Computer Networks, 2000, 34(4): 605-621

[9] *Manganaris S, Christensen M, Zerkle D, et al. A data mining analysis* of R_TID alarms[J]. Computer Networks, 2000, 34(4): 571 - 577

[10] Tseng, L.Y., Yang, S.B. A genetic approach to the automatic clustering problem[J]. Pattern Recognition, 2001. 34(2): 415 - 424

[11]Denning D E. An intrusion-detection model [J]. IEEE. Transactions on Software Engineering, 1987, 13(2): 222-232

[12] 张然，钱德沛，张文杰，刘轶，等. 入侵检测技术研究综述[J]. 小型微型计算机系统，2003，24(7)：1113-1118

[13] 蒋建春，马恒太，任党恩，等. 网络安全入侵检测研究综述[J]. 软件学报，2000，1(11)：1460-1466

[14] 陈硕，安常青，李学农. 分布式入侵检测系统及其认知能力[J]. 软件学报，2001，12(2)：225-232

[15] Han J, Kamber M 著. 数据挖掘概念与技术[M]. 范明，孟小峰 译. 北京：

机械工业出版社，2001

[16] 胡侃，夏绍玮. 基于大型数据仓库的数据采掘[J]. 软件学报，1998，9(1)：53-63

[17] 边肇祺，阎平凡，杨存荣. 模式识别[M]. 北京：清华大学出版社，1998

[18] 刘科，韩宗芬，金海，等. 分布式微入侵检测系统结构研究[J]. 华中科技大学学报，2001，210(11)：45—47

[19] 王丽娜，董晓梅，于戈，等. 基于进化神经网络的入侵检测方法[J]. 东北大学学报，2002 ,23(2)：107-110

[20] 郑 宏，陆阳，徐朝农. 基于BP神经网络的入侵检测系统分类器的实现[J]. 合肥工业大学学报(自然科学版)，2003，26(2)：281-285

[21] 刘美兰，姚京松. 神经网络在入侵检测系统中的应用[J]. 计算机工程与应用，1999,(6)：37-42

[22] 李鸿培，王新梅. 基于神经网络的入侵检测系统模型[J]. 西安电子科技大学学报，1999，26(5)：667-670

[23] 王文剑. 基于改进的BP网络模型的分类器的设计与实现[J]. 计算机工程与设计，1997，18(5)：43-45

[24] 董聪，刘西拉. 广义BP算法及网络容错性和泛化能力的研究[J]. 控制与决策，1998，13(3)：120-124

[25] 张莹莹. 基于LM的神经网络偏差补偿预测控制及其应用[J]. 福州大学学报，2001，29(1)：43-46

[26] 袁曾任. 人工神经元网络及其应用[M]. 北京：清华大学出版社，1999

[27] 伍春香，刘琳，王葆元. 三层BP网隐层节点数确定方法的研究[J]. 武汉测绘科技大学学报，1999，24(6)：34-37

[28] W. Richard Stevens 著. 范建华，胥光辉，张涛，等 译. TCP/IP详解卷1：协议[M]. 北京：机械工业出版社，2000

[29] Stephen Northcutt 著. 网络入侵检测分析员手册[M]. 余青霓，王晓程，周刚，等 译. 北京：人民邮电出版社，2000

[30] 唐正军 等.网络入侵检测系统的设计与实现[M].北京：电子工业出版社，2002

[31] Douglas E.Comer 著. 用TCP/IP进行网际互连第一卷:原理、协议和体系结构(第三版) [M]. 林瑶，蒋慧，杜蔚轩 译.电子工业出版社，1998

[32] Douglas E.Comer 著. 用TCP/IP进行网际互连第二卷:设计、实现和内部构成(第二版)[M]. 张娟，王海 译.电子工业出版社，2000

[33] 张文修，吴伟志，等. 粗糙集理论与方法[M]. 北京：科学出版社，2001

[34] 王清毅，范焱，蔡庆生. 知识的约简研究[J]. 小型微型计算机系统， 2000，21(6)：623～627

[35] 代建华，李元香. 一种基于粗糙集的决策系统属性约简算法[J].小型微型计算机系统，2003，24(3)：523～526

[36] 苗夺谦，胡桂荣.知识约简的一种启发式算法[J].计算机研究与发展，1999，36(6)：681～684

[37] 刘清. Rough集及Rough推理[M]. 北京：科学出版社,2001

[38] 闻新，周露，李翔，等. Matlab神经网络仿真与应用[M]. 北京：科学出版社，2003

[39] 苏金明，黄国明，刘波 编著. Matlab与外部程序接口[M]. 北京：电子工业出版社，2004

[40] 张文修，吴伟志，梁吉业，等. 粗糙集理论与方法[M]. 北京：科学出版社，2001

[41] 王国胤. Rough理论与知识获取[M]. 西安：西安交通大学出版社，2001

致　　谢

在本文即将结束的时候，我要特别感谢在研究生学习期间曾给予我关心和帮助的老师和同学。

首先，我要深深地感谢我的指导老师朱有产副教授，本论文从开始选题，到实验验证，以及论文的最后完成都得到了朱老师的精心指导和帮助，他严谨的学术作风和渊博的知识理论使我受益匪浅。

此外，在学习和课题研究过程中，网管中心的各位老师以及实验室的同学们也给予了我很大的帮助，在此也向他们表示深深的谢意。

最后，我还要特别感谢评阅此论文和对此论文提出建议的老师。

在学期间发表的学术论文和参加科研情况

发表的论文：

[1] 商李彪，朱有产，杜海江，窦炳琳. 基于网络的电力呼叫系统的设计与实现[J]. 微机发展，2004，14（3）：93～95

[2] 窦炳琳，朱有产，商李彪，张小松. 基于 Mobile Agent 的 SNMP 网络管理框架的研究[J].华北电力大学学报，2004，31（3）：100～103

[3] 窦炳琳，朱有产，商李彪，王文志. 告警关联方法研究[J].计算机应用与软件，已录用

[4] 朱有产，窦炳琳，商李彪，余晓晔. 告警关联在网络故障诊断中的应用研究[J].计算机应用，2004，24（10）：250～252

[5] 朱有产，窦炳琳，商李彪，余晓晔. 网络故障诊断框架的研究与应用[J].通信学报，已录用

[6] 张丽静，杨继家，商李彪. SVG 技术在实时信息发布系统中的应用[J].计算机仿真，已录用.

参加科研项目情况：

[1] 完成献县电力局管理信息系统部分子系统的现场安装与培训.