分类号 _____ 密级 _____

U D C _____ 编号 _____

南华大学

UNIVERSITY OF SOUTH CHINA

硕士学位论文

基于隐马尔科夫模型和神经网络的入侵检测研究

研 究 生 姓 名： 闫新娟

指导教师姓名、职称： 谭敏生 教授

学 科 、 专 业 名 称 ： 计算机应用技术

研 究 方 向： 计算机网络与信息安全

2014 年 9 月

南华大学学位论文原创性声明

本人声明，所呈交的学位论文是本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，除了论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，也不包含为获得南华大学或其他单位的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

年　　月　　日

南华大学学位论文版权使用授权书

本人同意南华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以编入有关数据库进行检索，可以采用复印、缩印或其它手段保留学位论文；学校可根据国家或湖南省有关部门规定送交学位论文。对于涉密的学位论文，解密后适用该授权。

作者签名：　　导师签名：

年　月　　日　　年　月　　日

目　　录

摘　　要

随着计算机网络应用的普及，它提供的信息交换、资源共享、分布式处理等服务极大地方便了人们对信息的需求。然而人们在享受这些方便的同时，网络安全问题也越来越成为关注的焦点。入侵检测作为网络安全的重要一环，它不仅检测来自外部的入侵行为，同时也监督内部用户的未授权活动。本文将隐马尔科夫模型和神经网络应用到入侵检测领域，提出了一个基于隐马尔科夫模型和神经网络的入侵检测模型。该模型由数据处理模块、数据训练模块、入侵检测模块和响应模块四部分组成。

本文中首先根据被攻击者发送响应包的特点，捕获、解析了 TCP、UDP、ICMP 三种协议的数据包，并按照协议类型对不同的数据包进行了预处理，使用 Baum-Welch 算法对模型进行了训练。针对隐马尔可夫模型(HMM)应用在入侵检测领域时观察值难以确定的问题，本文基于流量控制原理及 TCP/IP 模型，提出了一种隐马尔可夫模型观察值的确定方法，这种方法显著地减小了观察值集合规模，缩短了训练时间。通过实验确定了滑动窗口大小、输入层和隐含层神经元数并建立了相应的 BP 神经网络，然后结合隐马尔科夫模型和 BP 神经网络的理论设计了入侵检测算法，该算法把隐马尔科夫模型输出的最优序列划分为相同长度的小序列作为神经网络的输入来进行二次检测，通过神经网络的输出判断是否遭到入侵，从而提高了检测率。针对遭受的攻击，给出了对应的响应算法。

本文最后采用 java 技术实现了该模型，实施了相关的入侵检测实验，结果表明该模型针对性较强，大大减少了匹配的次数，比单独使用隐马尔科夫模型和神经网络都具有更高的检测率。

关键词：入侵检测；隐马尔科夫模型；神经网络；网络安全；协议

Research on the Hidden Markov Model And Neural Network of Intrusion Detection

ABSTRACT

With the popularization of computer network applications, information exchange, resources sharing, distributed processing and other services which are provided by the network get it more convenient for people to fulfill the need for information. However, when people enjoy these conveniences, network security is increasingly becoming the focus of concern. As an important part of network security, intrusion detection not only detects outside intrusions, but also supervises activities of unauthorized internal users. In this thesis, the hidden Markov model and neural network are applied to the field of intrusion detection, in addition, an intrusion detection model based on the hidden Markov model and neural network is proposed. This model consists of four parts, data processing module, data training modules, intrusion detection and response modules .

According to the characteristics of the attacked computers, they will send responses of packet first. In this thesis, the TCP, UDP, ICMP packets of three protocols are captured and analyzed. Also, In this thesis, data packets are differently pretreated in accordance with the protocol types, and the model is trained by using Baum-Welch algorithm . As for the problem that the observation results of Hidden Markov Model (HMM) used in the field of intrusion detection is difficult to figure out, a method to find out the observation results of Hidden Markov Model is proposed by this thesis, which is based on the principle of flow control and TCP / IP model. This method significantly reduces the collection size of observation results and shortens the training time. According to this method, through continuous experiments, the size of sliding window, the input layer and hidden layer neurons number are figured out and the corresponding BP neural network is established, then hidden combined with Markov model and BP

neural network design a detection algorithm. According to this algorithm, the optimal sequence which is output by hidden Markov model algorithm will be divided into small sequences of the same length as the input of neural network for a second test, whether the invasion occurs is judged by the output of the neural network. Therefore, the detection rate is improved. As for the different types of attack, the corresponding response strategies are given in this thesis.

Finally, this model is completed with java technology. The relevant invasion detection experiment is carried out in this thesis. The results show that the model is highly targeted, significantly reduces the number of matches, and has higher detection rate than the application of the hidden Markov model nuclear neural network alone.

Key words: Intrusion Detection; Hidden Markov Model; Neural Network; Network Security; Protocol

Yan Xinjuan (Computer application technology)

Directed by Tan MinSheng

插图索引

附表索引

第一章　绪论

1.1 研究背景及研究意义

随着网民规模的持续扩大，互联网应用在资讯传播中的优势凸显，电子政务、电子商务、网络游戏、博客、微博等互联网业务正在快速扩展。网络在人们的工作、学习、生活等方面的作用越来越不可忽视，然而网络的开放性带来的不安全风险也随之增多，各种网络入侵行为的大量存在并不断攀升成为了网络安全的最大隐患。

在美国，2008 年因域名仿冒等网络钓鱼造成的损失大约是 40 亿美元，并以 25%的速度递增。《2009 年中国电脑病毒疫情及互联网安全报告》中显示[1]：2009 年金山毒霸“云安全”中心共截获新增病毒和木马 20684223 个，与 2008 年相比增加了 49%，与 5 年前新增病毒数量相比，增长了近 400 倍（如图 1.1 所示）。2010 年 360 安全中心发布监测报告中指出：2010 年国内新增游戏盗号、网银盗号和支付劫持类给中国网民造成了 51.9 亿元的直接经济损失。因此，网络攻击已成为人类社会信息化所面临的巨大威胁，同时也关系着国家网络经济发展。如何增强计算机系统和网络系统的安全性的研究成为了举世瞩目的焦点[2]。

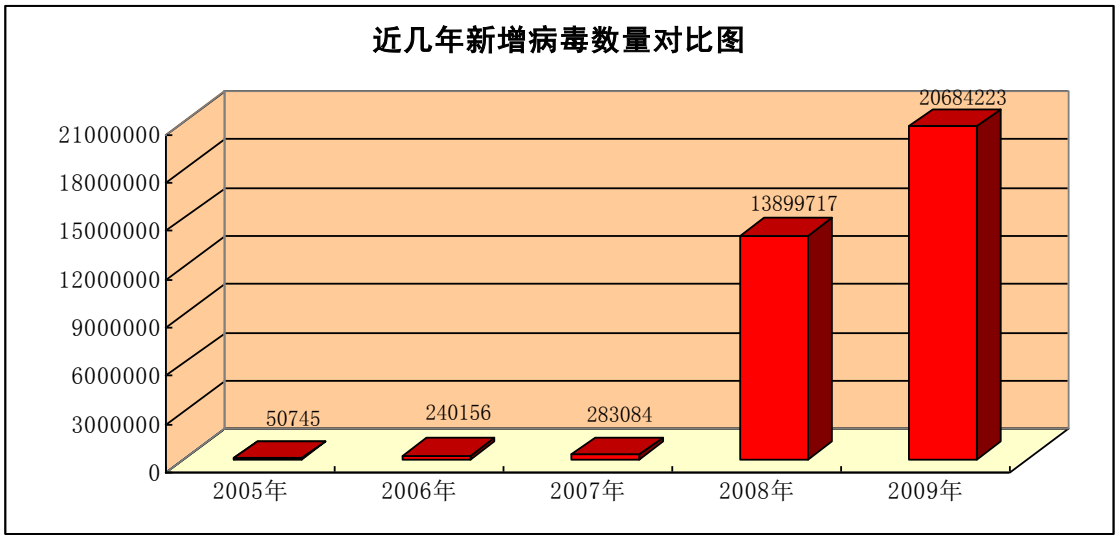

图 1.1 近几年新增病毒数量对比图

传统意义上的网络安全防御技术是通过修补软件漏洞、防火墙日志检测、系统

日志检测等来建立安全防御系统，阻止可能出现的攻击[3]。这些方法往往是在攻击已经发生的情况下采取的某些弥补措施，显得过于被动，并且不能及时有效地反制相应的攻击。于是有人提出了主动的网络安全防御体系，网络入侵检测系统就是其中的代表，所以针对入侵检测的研究自然成了国内外研究机构关注的热点。

其实无论哪种攻击，其攻击方法、步骤、危害，甚至每次针对哪种协议展开攻击都有相似之处。因此，对入侵进行检测的出发点就在于:通过对攻击的某些特征进行分析和研究，从中挖掘出这些特征的规律，并根据规律推断是否为入侵。

经过这么多的研究，入侵检测技术取得了很大的进展。概括起来，当前网络攻击预测的研究方向可以分为微观和宏观两个角度[4]。

微观角度一般是对攻击步骤和路径进行预测：通过描述并分析一些网络攻击前奏，长时间分析黑客的攻击方式、手段、规律以及习惯等，提前发现黑客攻击行为并给出响应，达到尽可能早地做出安全防范办法，为系统实施有针对性的安全防护措施提供指引，这样可最大限度地避免因安全隐患而给用户带来的损失[5]。宏观角度主要是对攻击发生的规模、趋势、使用的协议等进行预测。主要根据已知的攻击特征和信息预测即将发生的攻击，一般是通过各种数学方法分析和挖掘网络已有的攻击数据，然后从中提取感兴趣的重要特征和相关信息。

目前，从微观角度针对攻击步骤和路径的研究已经有了不错的效果，如基于攻击意图[6]、博弈论[7]等方法；而对于从宏观角度则仍处于起步阶段。正是基于这样的考虑，本文力图在这一特殊领域深入研究、寻求突破。

衡量一个入侵检测系统的好坏，主要是看其误报率和漏报率。本文把隐马尔科夫模型和神经网络结合起来应用于入侵检测的主要原因是：

隐马尔可夫模型(HMM)具有模型研究透彻、算法成熟、效率高、效果好、易于训练等优点[8]，但是这种检测算法的训练数据比较大，并且不易收敛。文献[9]就提出了一种隐马尔科夫模型的入侵检测方法，此算法虽然仅需要较少的训练数据，就可以建立起一个比较完备的轮廓数据库，但是还是会占用较大的系统资源。

BP 神经网络以自适应，自学习，自组织，较好的容错性和鲁棒性等优点而在入侵检测领域发挥了重要的作用[10]。它通过对样本进行学习，调整 BP 神经网络中的连接权值，就可以实现非线性分类等问题。更重要的是，BP 神经网络技术应用于入侵

检测系统也存在着严重不足，比如：传统的 BP 传播算法收敛速度慢，网络易陷于局部极小[11]。

本文提出的基于隐马尔科夫模型和神经网络的入侵检测模型同其它的入侵检测方法的不同之处表现在以下几个方面:第一，检测效果更好，提高了异常检测率；第二，缩减了轮廓数据库，节省了系统的存储空间；第三，收敛速度快。

1.2 研究现状

1980 年 4 月，James Anderson 的一份题为,《Computer Security Threat Monitoring and surveillance》的技术报告，首次将入侵检测引入到计算机安全领域[12]。1986 年 Denning 发表的论文“An Intrusion Detection Model”被认为是入侵检测领域的开山之作[13]。1990 年，加州大学的 Davis 分校的 Todd Heberlien 开发的 NSM 系统，第一次提出将网络数据包作为信息源[14]，从此入侵检测的研究分成了两大阵营:基于主机的入侵检测和基于网络的入侵检测。1998 年，美国国防部“先期概念技术演示计划”(ACTD)发起了名为 IA:AIDE 的项目，该项目结合信息保障，尝试利用专家系统融合来自各安全组件的信息，对入侵进行确认并发现攻击迹象 [15]。1999 年，Wenke Lee 提出了入侵检测的数据挖掘框架[16]。文献[17]首次将攻击意图作为一个单独的因素来考虑，并利用扩展的目标树对攻击意图建模，预测攻击者可能的后续攻击。文献[18]描述了一种通过构造攻击轮廓(包括攻击历史活动、攻击手段、攻击步骤、攻击动机、目标、审记标识等内容)来检测入侵的方法，但是其最大的缺点就是构造攻击轮廓本身很难并且有巨大的开销。

AndrewRathnleu 首次提出了信息战中攻击预警的概念，并研究了开放性信息决策支持系统的框架，以构成信息战攻击威胁评估和预警决策支持系统 IWAAS(information Warfare Attack Assessment System)[19]。1999 年，英国 IAAC(Information Assurance Advisory Council)启动了“信息安全保障的威胁评估与预警”项目，研究攻击评估的量化方法和预警方法，但他们的研究仅仅局限在一个子域(su-State)内对安全威胁的报警[20]。文献[21][22]通过隐着色 Petri 网(HCPN)对报警信息进行关联研究，提出了基于隐含着色 Petri 网的攻击预测模型，该模型主要利

用代理获取系统资源这个概念取代了传统着色 Petri 网中 Place 属于颜色集的概念，该模型的关键在于引入了一个称作观察值的概念。

国内目前应用在入侵检测方面的研究成果还是很喜人的，国内除了瑞星，还有金山和微点等。其中北京东方微点信息技术有限责任公司研发的微点主动防御软件处于世界领先水平。在理论方面，文献[23]将隐马尔可夫模型应用到入侵检测上，给出了一种有效确定隐马尔可夫模型观察值的方法，同时提出了一个基于隐马尔可夫模型的入侵检测模型。文献[24]提出了一种混合入侵检测模型,主要是基于主机的，从系统调用的角度考虑。文献[25]基于支持向量机的攻击预测，预测对象是风险评估的攻击态势，对预测结果的指导性不强。文献[26]中把攻击意图和隐马尔科夫模型结合在一起，提出了一种基于隐马尔可夫模型的网络安全预警技术研究。文献[27]提出了基于攻击响应事件相关性的攻击预测算法，该算法通过攻击事件间的前驱后继关系，预测今后一段时间内同一个 IP 地址可能受到的攻击行为，但由于网络攻击的复杂性，针对具体的攻击行为构造前驱后继关系是很困难的。文献[28]利用 Intemet 骨干网异常流量发现原理，通过对端口、TCP 等直接变量的监测，使用统计学中的时间序列分析方法，实时分析和发现流量异常，并发布预警信息。文献[29]给出了基于神经网络的入侵检测研究，对神经网络理论中的 BP 算法进行了改进，以满足 IDS 所需要的实时性、适应性、准确性和自学习能力等方面的需求。文献[30]给出了基于粗糙集-神经网络的入侵检测系统的研究,融合了粗糙集理论和神经网络各自的优点,提出了一种基于粗糙集-神经网络的入侵检测模型。

1.3 本文研究的主要内容

根据目前入侵检测的主要发展来看,在包这一级进行分析,以及简单地通过匹配进行报文数据的分析,已经解决不了什么实际问题。攻击发生时,真正能够判断攻击出现的,只有通过协议的整体分析。基于这样的考虑，本文在分析了入侵检测研究现状的基础上结合 TCP/IP 模型、隐马尔科夫模型和神经网络的基本理论，主要研究针对网络协议展开的入侵，提出了一种基于隐马尔科夫模型和神经网络的入侵检测模型。

本文研究的主要内容是围绕网络协议的攻击来展开，提出的基于隐马尔可夫模型和神经网络的入侵检测模型，主要从以下几个方面展开工作：

(1)分析了 TCP/IP 协议导致的安全漏洞，详细介绍了 IP、TCP、UDP、ICMP 四种网络协议，分析了相关的入侵的产生原因、特征。

(2)给出了隐马尔科夫模型中观察值的确定方法。隐马尔可夫模型观察值千差万别、难以确定，隐马尔可夫模型观测值以及观测空间的确定是应用隐马尔可夫模型的核心问题，即选择什么对象作为模型的分析对象。本文结合 TCP/IP 管理模型、流量控制原理以及协议自身的特点，提出了一种确定基于隐马尔可夫模型的观察值的方法，为入侵做了铺垫。

(3)确定神经网络模型——反向传播神经网络(BP)。随着应用研究的不断深入，神经网络模型已达几十种。比较常用的有四种基本模型：反向传播模型、多层感知器模型、Hopfield 模型(HoP)、自组织模型(SOM)。其中反向传播神经网络是最具代表性的一种神经网络模型，它利用非线性可微分函数进行权值训练，同时也可以计算输入数据和神经网络要识别的特征之间的相似度，反向传播神经网络包含了神经网路中最精华的部分，所以在本文中选择了反向传播神经网络为研究对象，并建立了神经网络。

(4)提出了入侵检测算法。入侵检测模块是本文的创新点之一，它由两部分组成：一个是隐马尔科夫检测部分、另一部分是神经网络检测部分。在隐马尔科夫模型中，有两个算法对入侵检测模型起着关键作用，即 Baum-Welch algorithm 和 vietbri 算法。Baum-Welch 算法是用来训练隐马尔科夫模型，而 Vitberi 算法用来计算隐马尔科夫模型的输出，隐马尔科夫模型的输出作为神经网络的输入。神经网络的目标是：当没有检测到攻击时输出值为 0，而检测到攻击时输出是 1。

(5)最后，本文实现了一个基于隐马尔科夫和神经网络的入侵检测模型，通过实验数据验证此模型比单独使用隐马尔科夫模型或者是单独使用神经网络的检测模型有较高的检测率。

1.4 本文的组织结构

本文主要包括六个章节：

第一章 绪论。包括课题的研究背景及研究意义、国内外研究现状，本文研究的主要内容及论文的组织结构。

第二章 入侵检测技术相关理论。主要从入侵检测系统模型、入侵检测系统分类、还介绍了 TCP/IP 模型的基本原理，分析了几种主要的协议格式，最后通过几种常见的基于协议的攻击来了解入侵。

第三章 隐马尔科夫模型和神经网络概述。介绍了隐马尔可夫模型原理及需要解决的三个基本问题，还介绍了 BP 神经网络的相关理论。

第四章 基于隐马尔科夫模型和神经网络的入侵检测模型。建立了基于隐马尔科夫模型和神经网络的入侵检测模型，给出了隐马尔科夫模型中观察值的确定方法、建立了 BP 神经网络、给出了模型中关键模块的算法。

第五章 模型的实现及结果分析。搭建了实验环境，完成模型的设计，并对实验结果进行分析。

第六章 总结与展望。对全文的研究内容进行总结，并给出文中所存在的不足以及今后的进一步工作。

第二章　入侵检测技术相关理论

2.1 引言

入侵检测（Intrusion Detection），顾名思义就是对入侵行为的发觉，是入侵检测的软件与硬件的组合。入侵检测是一种主动的安全技术，具有识别入侵行为，阻止入侵事件的发生和事态扩大等作用[31]。入侵检测是防火墙的合理补充，帮助系统对付网络攻击，扩展了系统管理员的安全管理能力（包括安全审计、监视、进攻识别和响应），提高了信息安全基础结构的完整性。它从计算机网络系统中的若干关键点收集信息，并分析这些信息，与其他安全产品最大的不同的就是：入侵检测系统更智能。

2.2 入侵检测系统模型

最早的入侵检测模型是由 Denning 给出的，该模型主要根据主机系统审计记录数据，生成有关系统的若干轮廓，并检测轮廓的变化差异发现系统的入侵行为，如图 2.1 所示：

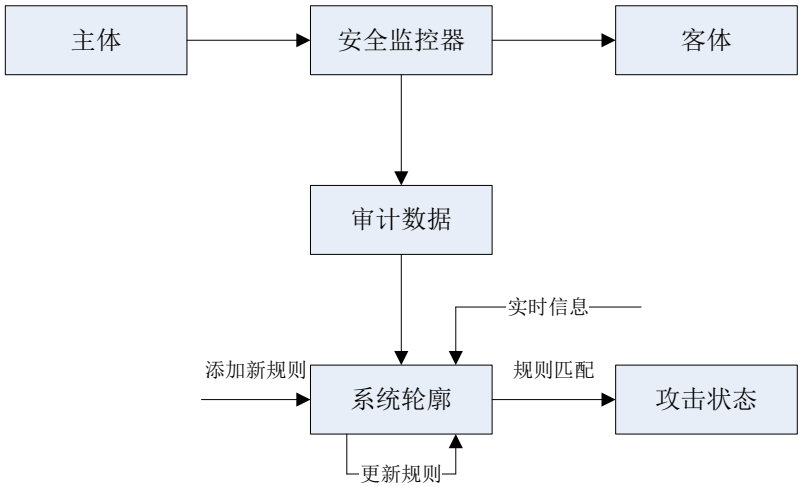

图2.1 最早的入侵检测模型

随着入侵类型越来越多，再加上许多攻击经过长时期准备，通过网上协作进行。面对这种情况，入侵检测系统的不同功能组件之间、不同 IDS 之间共享这类攻击信息就变的十分重要的。为此 chen 提出的一种通用的入侵检测框架模型，简称 CIDF[32]产生了。该模型认为入侵检测系统由事件产生器(EventBox)、事件分析器(AnalyzersBoxeS)、响应单元(Countemreasure)和事件存储(StorageBoxes)组成，如图 2.2 所示:

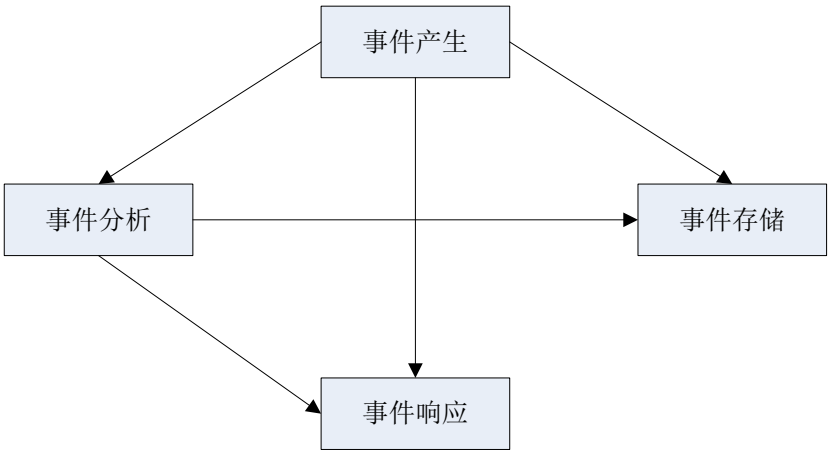

图2.2 入侵检测系统结构

CIDF 将入侵检测系统需要分析的数据统称为事件(event)，它可以是基于网络的入侵检测系统从网络中提取的数据包，也可以是基于主机的入侵检测系统从系统日志等其它途径得到的数据信息，它提供了一整套标准的应用程序接口（API 函数）。

事件产生器：从入侵检测系统外的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。事件产生器是所有 IDS 所需要的，同时也是可以重用的。

事件分析器：从其他组件接收入侵检测对象，分析得到的数据，并产生新的入侵检测对象。

事件响应：对分析结果做出响应的功能单元，它可以终止进程、重置连接、改变文件属性等，也可以只是简单的报警。

事件存储：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 标准因为没有正式确立，所以现在还没有被哪个入侵检测商业产品采用。当然正是这样使得目前的入侵检测市场上的产品在系统之间的相互操作性很差，因

此各厂商都把今后努力的方向定在按照 CIDF 进行信息交换的标准化工作上来成为了一种需要。

2.3 入侵检测系统的分类

入侵检测系统可以分为两个类：检测数据来源和检测原理。

2.3.1 检测数据来源

根据检测数据来源来分，入侵检测系统可分为基于主机的入侵检测系统和基于网络的入侵检测系统。

（1）基于主机的入侵检测系统

基于主机的入侵检测系统出现在 20 世纪 80 年代初期，它具有检测效率高，分析代价小，分析速度快等特点，能够迅速并准确地定位入侵者，并可以结合操作系统和应用程序的行为特征对入侵进行进一步分析[33]。它是从单个主机上提取数据(如审计记录等)作为入侵分析的数据源，对关键的系统文件和可执行文件进行检测，此外，大多数 HIDS 产品都监听端口的活动，在特定端口被访问时向管理员报警。所以可在本地网络的各服务器上安装基于主机的入侵检测软件，通过对系统日志进行密切监视，把检测结果及时通知给系统管理员从而识别出攻击行为。然而基于主机的入侵检测系统存在的问题是：对系统的可靠性要求比较高，为了提取入侵信息，它要求系统除了具有自身应该具备基本的安全功能外，还要具有比较合理的设置。因为这些要求严格的设置，使得对操作系统及其了解的攻击者很有可能在实施完攻击后销毁证据。因此，基于主机日志的入侵检测系统不如基于网络的入侵检测系统。

（2）基于网络的入侵检测系统

基于网络的入侵检测系统通过网络监视来实现数据提取，不需主机提供严格的数据审计，对主机资源消耗较少。网络监视可以获得所有的网络信息数据，只要时间允许，可以在庞大的数据堆中提取和分析需要的数据。可以对一个子网进行检测，一个监视模块可以监视同一网段的多台主机的网络行为，不改变系统和网络的工作

模式。也可以在基本不影响网络性能的情况下实现监视，很难被入侵者发现，也可以从低层开始分析，对基于协议攻击的入侵手段有较强的分析能力。网络监视的主要问题是监视数据量过于庞大并且它不能结合操作系统特征来对网络行为进行准确的判断。基于网络的入侵检测系统的部署对网络本身的影响很小，但对带宽有较高要求。基于网络的入侵检测系统依靠对网络数据包和网络故障等的分析来进行检测，对入侵行为能够做出快速响应，通常采用多个入侵检测器配合一个入侵检测系统指示器进行分布式协同检测。

总之，基于主机的入侵检测只能检测单个主机系统，而基于网络的 IDS 则对本网段的所有网络数据进行检测，多个分布于不同网段上的基于网络的 IDS 可以协同工作以提供更强的入侵检测能力。因此由于基于网络的入侵检测容易处理和分析数据，目前很多入侵检测系统倾向于采用基于网络的检测手段来实现。

2.3.2 检测原理

入侵检测系统根据检测原理可分为异常入侵检测系统和误用入侵检测系统[34]。

（1）异常入侵检测系统

异常入侵检测系统是指建立系统的正常模式轮廓，若实时获得的系统或用户的轮廓值与正常值的差异超出指定的阈值，就进行入侵报警。异常入侵检测(基于统计的入侵检测)技术是建立在如下假设的基础上:即任何一种行为都能由于其偏离正常或者所期望的系统和用户的活动规律而被检测出来。理想情况下，异常行为集合等同于入侵行为集合，如果发现当前的活动情况偏离了系统中存放的正常用户的行为模型，则系统发出报警信号。但是在现实中，入侵行为集合通常不等同于异常行为集合。事实上，行为有以下 4 种状况:(1)行为是入侵行为，但不表现异常；(2)行为是入侵行为，且表现异常；(3)行为既不是入侵行为，也不表现异常；(4)行为不是入侵行为，却表现异常。异常检测方法的基本思路是构造异常行为集合，从中发现入侵行为。异常检测依赖于异常模型的建立，不同模型构成不同的检测方法，它需要获得入侵的先验概率，但是如何获得这些入侵先验概率就成为异常检测方法是否成功的关键问题。异常入侵检测方法的优点是不依赖于攻击特征，立足于受检测的

目标发现入侵行为，但是如何对检测建立异常指标，并且每个指标需要随着系统的运行而更新，造成计算量庞大[35]，如何定义正常模式轮廓，降低误报率，都是难以解决的课题。

异常检测的主要统计模型有隐马尔可夫模型，人工神经网络模型，数据挖掘等。

（2）误用入侵检测系统

误用入侵检测(基于特征的入侵检测)系统是建立在对过去各种入侵和系统缺陷的知识积累上，它需要首先建立一个特征数据库，然后在各种收集到的信息中寻找与数据库项目相关的信息。当符合条件的线索被发现后，它就会报警，任何不符合特定匹配条件的活动将被认为是合法的和可以接受的，即使其中包含着隐藏的入侵行为(表明入侵特征数据库收集到的信息不全)。该模式的缺点是只局限于发现已知的攻击，对未知的攻击不能形成有效识别，优点是可以有针对性的建立高效的入侵检测系统，对已知的攻击检测成功率高。当前主流的入侵检测系统基本采用了误用检测模型。

误用检测采用的主要技术有模式匹配、专家系统和状态转换等。

2.4 协议分析

一般而言，理论攻击是技术攻击的理论基础，几乎每种技术攻击都可以最终归结为某类理论攻击，例如对路由器、路由表、DNS 服务器域名表的窜改就属于密码学上的完整性侵犯，TCP 序列号猜测攻击可归入密码学上的假冒攻击。但理论攻击却未必能成为现实可行的技术攻击。例如差分密码分析已是一种较为成熟的对迭代分组密码的理论攻击方法，然而要将其变为技术攻击手段，切实破译某一特定密文，仍存在需要获取大量明文选择及大量专家干预的困难，即存在数据复杂性和处理复杂性。同样密码学意义上的理论攻击所涵盖的对某些加密算法的攻击、对签名算法的攻击、对密钥交换和认证协议的攻击也未必能举出确实有效的实现方法。因此，到目前为止，对网络入侵的分类主要着眼于技术层次，本文研究的攻击也是集中在这一层面上的，主要是研究针对网络协议的攻击。

2.4.1 TCP/IP 协议模型

TCP/IP[36]参考模型是计算机网络的祖父 ARPANET 和其后继的因特网使用的参考模型。ARPANET 是由美国国防部 DoD(U.S.Department of Defense)赞助的研究网络，逐渐地它通过租用的电话线连结了数百所大学和政府部门。当无线网络和卫星出现以后，现有的协议在和它们相连的时候出现了问题，所以需要一种新的参考体系结构，这个体系结构在它的两个主要协议出现以后，被称为 TCP/IP 参考模型（TCP/IP reference model）。

TCP/IP 参考模型是一组用于实现网络互连的通信协议，是 Internet 网络体系结构的核心。基于 TCP/IP 的参考模型是一个四层的分层体系结构，它们分别是：网络接口层、网际层、传输层和应用层，如图 2.3 所示：

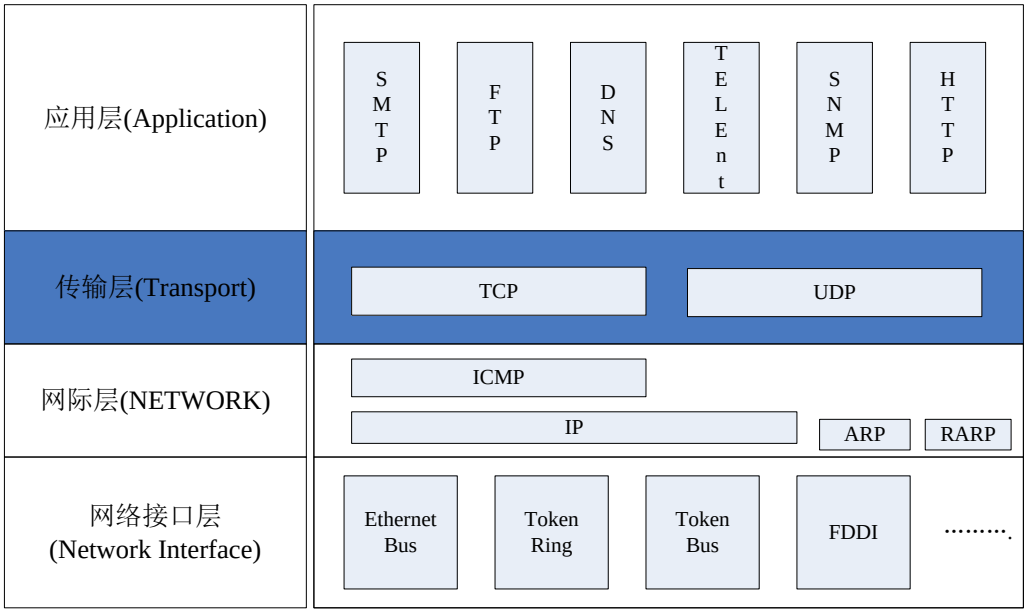

图2.3 TCP/IP参考模型

2.4.2 TCP/IP 协议中的主要协议格式

（1）IP 数据报头部格式

互联网层的基本传输单元是 Internet 数据报（IP 数据报）IP(Internet Protocol)即网际协议。它定义了在整个TCP/IP互联网上数据传输用的基本单元，IP

数据报报头格式如表2.1所示：

表 2.1 IP 数据报报头格式

版本（4 位）	IHL（4 位）	服务类型（6 位）		总长度（16 位）		
标识（16 位）				DF 1 位	MF 1 位	分段偏移（14 位）
生命期（8 位）		协议（8 位）		头部效验（16 位）		
源地址（32 位）						
目的地址（32 位）						
选项（0 或多个字）						

在 IP 数据报中，“协议号”表示被封装协议的类型，协议号的值为“06”表示被封装的上层协议是 TCP，协议号的值为“11”表示被封装的上层协议是 UDP，协议号的值为“01”表示被 IP 封装的协议是 ICMP，协议号的值为“04”表示被 IP 封装的协议是 IP。

（2）TCP 数据段的头

TCP（Transmission Control Protocol，传输控制协议）是专门为了在不可靠的互联网上提供一个可靠的端到端字节流而设计的,它在传送数据时是分段进行的，TCP 数据报文格式如表 2.2 所示：

表 2.2 TCP 数据报文格式

源端口（16 位）								目的端口（16 位）
序列号（32 位）								
确认号（32 位）								
TCP 头部(4 位)								窗口大小
校验和（16 位）								紧急指针（16 位）
可选项（0 或多个 32 位）								
数据（可选项）								

在 TCP 报文段的标准格式中有很多字段，为此主要对 TCP 头部的一些重要字段进行分析。

1)源端口:分配给发起连接的主机的虚连接端口号。

2)目的端口:由发起连接的主机分配的想连接的目的机的端口号,该号通常是保留的端口号，对应具体的连接类型。例如，如果打开到特定主机的 WWW 连接，则目的端口将被设置为“80”(80 端口是保留的用于娜 W 服务的 HTTP 端口)。

3)序列号和确认号:发送方和接收方使用这两个号以确保包没有丢失、没有重复以及可以在目的节点以正确顺序重新组装。

4)码元比特:这个字段共包含六个位(比特位置“1”表示某种控制关系)，它们是 TCP 报文段的第 14 字节的后六位，分别为:URG 位、ACK 位、PSH 位、RST 位、SYN 位和 FIN 位。

URG 位置“1”:向接收方表明一旦接收完数据就进行紧急处理。

ACK 位置“1”:表明确认号字段是有意义的。

PSH 位置“1”:表明必须迅速地将数据传递到接收方。

RST 位置“1”:表明要立即复位连接。

SYN 位置“1”:在需要同步序列号的情况下设置。

FIN 位置“1”:连接将要关闭。

（3）UDP 数据段的头

UDP（User Datagram Protocol，用户数据协议）为应用程序提供了一种方法来发送经过封装的 IP 数据报，而且不必建立连接就可以发送这些 IP 数据报。UDP 数据报文格式如表 2.3 所示：

表 2.3 UDP 数据报文格式

源端口（16 位）	目标端口（16 位）
UDP 长度（16 位）	UDP 校验和（16 位）

（1）源端口：发送方的 UDP 端口号，用于多路复用。

（2）目标端口：接收方的 UDP 端口号，用于多路复用。

（3）UDP 长度：包括 UDP 报头和数据在内的报文长度值，以字节为单位，最小为 8（包头长度)。

（4）校验和：为可选字段，如果该字段设置为 0，则表示发送方没有为该 UDP

数据报提供校验和。

使用 UDP 端口的常见应用层服务有:使用“53”端口的 DNS，使用“101”端口的 SNMP 等。

（4）ICMP 数据报的头

ICMP 报文的类型很多，且各自又有各自的代码，因此，ICMP 并没有一个统一的报文格式，不同的 ICMP 类别分别有不同的报文字段，本文以回送报文为例。ICMP 数据报文格式如表 2.4 所示：

表 2.4 回送报文格式

类型（8 位）	代码（8 位）	校验和（16 位）

对于 ICMP 报文来说，当报文类型 type=0 时，表示对 ping 命令的应答；type=8 时，表示 ping 命令请求；type=3 时，表示目的站不可达。

2.4.3 几种常见的基于协议的攻击

（1）泪滴攻击

泪滴攻击是利用在 TCP/IP 堆栈中实现信任 IP 碎片中的包的标题头所包含的信息来实现自己的攻击。对于一些大的 IP 包，需要对其进行分片传送，这是为了迎合链路层的 MTU（最大传输单元）的要求。比如，一个 4500 字节的 IP 包，在 MTU 为 1500 的链路上传输的时候，就需要分成三个 IP 包。在 IP 报头中有一个偏移字段和一个分片标志（MF），如果 MF 标志设置为 1，则表面这个 IP 包是一个大 IP 包的片断，其中偏移字段指出了这个片断在整个 IP 包中的位置。例如，对一个 4500 字节的 IP 包进行分片（MTU 为 1500），则三个片断中偏移字段的值依次为：0，1500，3000。这样接收端就可以根据这些信息成功的组装该 IP 包。如果一个攻击者打破这种正常情况，把偏移字段设置成不正确的值，即可能出现重合或断开的情况，就可能导致目标操作系统崩溃。

（2）IP地址欺骗

IP 地址欺骗是指行动产生的 IP 数据包伪造的源 IP 地址，以便冒充其他系统或保护发件人的身分。这种攻击利用 RST 位来实现。假设现在有一个合法用户

(61.61.61.61)已经同服务器建立了正常的连接，攻击者构造攻击的 TCP 数据，伪装自己的 IP 为 61.61.61.61，并向服务器发送一个带有 RST 位的 TCP 数据段。服务器接收到这样的数据后，认为从 61.61.61.61 发送的连接有错误，就会清空缓冲区中建立好的连接。这时，如果合法用户 61.61.61.61 再发送合法数据，服务器就已经没有这样的连接了，使服务器不对合法用户服务，从而实现了对受害服务器的 DOS(拒绝服务)攻击。

（3）LAND 攻击[37]

LAND 攻击利用了 TCP 连接建立的三次握手过程，通过向一个目标计算机发送一个 TCP SYN 报文(连接建立请求报文)而完成对目标计算机的攻击。与正常的 TCP SYN 报文不同的是，LAND 攻击报文的源 IP 地址和目的 1P 地址是相同的，都是目标计算机的 IP 地址。这样目标计算机接收到这个 SYN 报文后，就会向该报文的源地址发送一个 ACK 报文，并建立一个 TCP 连接控制结构(TCB)，而该报文的源地址就是自己，因此，这个 ACK 报文就发给了自己。这样如果攻击者发送了足够多的 SYN 报文，则目标计算机的 TCB 可能会耗尽，最终不能正常服务。LAND 攻击是一种 DOS 攻击。

(4) SYN FL00D 攻击[38]

SYN FL00D 攻击同样是利用 TCP 协议三次握手的机制，由攻击主机向被攻击设备发送大量的 SYN 请求报文，这些报文的源地址是一个不可达的主机地址，被攻击设备发送 SYNACK 报文后，就开始等待大量根本不可能到达的 ACK 报文，造成了系统资源的大量占用。利用这个过程，一些恶意的攻击者可以进行所谓的 TCP SYN 拒绝服务攻击:

1)攻击者向目标计算机发送一个 TCP SYN 报文。

2)目标计算机收到这个报文后，建立 TCP 连接控制结构(TCB)，并回应一个 ACK，等待发起者的回应。

3)而发起者则不向目标计算机回应 ACK 报文，这样导致目标计算机一直处于等待状态。

可以看出，目标计算机如果接收到大量的 TCPSYN 报文，而没有收到发起者的第三次 ACK 回应，会一直等待，处于这样尴尬状态的半连接如果很多，则会把目标计算机的资源(TCB 控制结构，TCB 一般情况下是有限的)耗尽，而不能响应正常的

TCP 连接请求。

(5) UDP echo flooding[39]

各种各样的假冒攻击利用简单的 TCP/IP 服务，如 chargen(字符发生器)和 Ech。(回显)来传送毫无用处的占满带宽的数据。通过伪造与某一主机的 Chargen 服务之间的一次的 UDP 连接，回复地址指向开着 ECho 服务的一台主机，这样就生成在两台主机之间的足够多的无用数据流，如果有足够多的数据流就会导致带宽耗尽的服务攻击。

(6) Smurf 攻击[40]

Smurf 攻击方法结合使用了 IP 欺骗和 ICMP 回复方法使大量网络传输充斥目标系统，引起目标系统拒绝为正常系统进行服务。Smurf 攻击通过使用将回复地址设置成受害网络的广播地址的 ICMP 应答请求(ping)数据包，来淹没受害主机，最终导致该网络的所有主机都对此 ICMP 应答请求做出答复，导致网络阻塞。更加复杂的 Smurf 将源地址改为第三方的受害者，最终导致第三方崩溃。攻击的过程是这样的：Woodlly Attacker 向一个具有大量主机和因特网连接的网络的广播地址发送一个欺骗性 Ping 分组（echo 请求），这个目标网络被称为反弹站点，而欺骗性 Ping 分组的源地址就是 Woolly 希望攻击的系统。

(7) Ping o f death 攻击[41]

Ping Of Death 攻击是一种拒绝服务攻击，方法是由攻击者故意发送大于 65535 字节的 ip 数据包给对方。TCP/IP 的特征之一是碎裂；它允许单一 IP 包被分为几个更小的数据包。该攻击数据包大于 65535 个字节。由于部分操作系统接收到长度大于 65535 字节的数据包时，就会造成重启、内存溢出、系统崩溃、、内核失败等后果，从而达到攻击的目的。

2.5 小结

本章在介绍了入侵检测的基本理论后,紧接着介绍了 TCP/IP 参考模型的四层结构及 TCP/IP 协议簇的主要协议。最后，详细介绍了几种常见的针对协议的攻击。为第四章 HMM 观察值的确定方法做好了铺垫。

第三章　隐马尔科夫模型（HMM）和 BP 神经网络的概述

3.1 引言

隐马尔可夫模型[42](Hidden Markov Model，简称HMM)是一种用参数表示的，用于描述随机过程统计的概率模型，它能够利用收集的训练样本进行自适应学习，得到相应的隐马尔可夫模型，其算法成熟，效率高目前已经常用于入侵检测的研究。而在人们提到的几十种神经网络模型中，人们用的最多的是BP网络，它是由一系列的处理单元组成，这些处理单元是高度关联的，把一组输入转化为一组输出，通过修改节点之间的连接关系，使得神经网络的输出为所想要的结果[43]。下面了解一下这两个概念：隐马尔科夫模型和神经网络。

3.2 隐马尔科夫模型理论

3.2.1 模型描述

隐马尔可夫模型自 20 世纪 80 年代以来，被广泛应用于语音识别，并取得重大成功。到了 90 年代，HMM 还被引入计算机文字识别和移动通信核心技术“多用户的检测”。近年来，HMM 在生物信息科学、故障诊断等领域也开始得到应用。隐马尔可夫模型是在马尔可夫链的基础上发展起来的一种统计分析模型，在马尔可夫链模型中，观察值和状态是一一对应的，观察者观察到了观察值，也就知道了这个观察值所处的状态。而在隐马尔可夫模型中，观察值和状态不是一一对应的，观察者只能看到观察值，不能直接看到状态，只能通过一个随机过程去感知状态的存在及其特性。也就是说，HMM 是一个双重随机过程，由两个部分组成：一个是马尔可夫链，描述状态的转移，用转移概率描述。另一个是一般随机过程，描述状态与观测序列间的关系，用观测值概率描述。即若得到一个观察值序列 $O=\{O_1，O_2，\cdots，O_T\}$，不能由此观察值序列直接得到状态序列 $S=\{S_1，S_2，\cdots S_T\}$，若要得到实际的状态序列，那么就必须知道观察值在每种状态的分布情况、状态的

初始概率、以及状态转移概率。如图 3.1 所示：

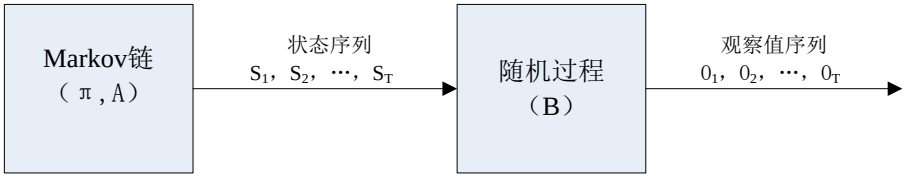

图3.1 隐马尔科夫模型双重随机过程关系图

一旦一个问题可以用隐马尔科夫模型描述，那么就可以使用 5 个参数描述一个基于 HMM 的正常模型 λ=(N,M, π,A, B)，为了方便使用，也可以使用 λ=(A,B, π) 来描述一个完整的隐马尔科夫模型。其中 HMM 涉及到的一些参数描述如表 3.1 所示[44]：

表 3.1 HMM 的参数描述

参数	描述
状态数 N	状态的有限集合 $S=\{S_1, S_2, \cdots, S_M\}$
观测值数 M	观察值有限集合 $O=\{O_1, O_2, \cdots, O_M\}$
初始状态分布 $\pi=\{\pi_i\}$	t=1 时处于状态 S_i 的概率 $\pi_i=P(q_1=S_i)$
状态转移矩阵 $A=\{a_{ij}\}$	从状态 i 到状态 j 的概率 $a_{ij}=P(q_{t+1}=S_j \mid q_t=S_i)$
观察值产生的概率矩阵 $B=\{b_j(k)\}$	在 S_j 状态下，t 时刻出现的 O_k 的概率 $b_j(k)=P(O_t=O_k \mid q_t=S_j)$

(1)状态数 N：用集合 S 表示，且 $S=\{S_1, S_2, \cdots, S_N\}$。在时刻 t 的状态为 q,，状态是被隐藏的信息。

（2）观测值数 M：在不同的状态下，状态以概率的形式表现不同的观察值，用集合 O 表示，$O=\{O_1, O_2, \cdots, O_M\}$ 是观察值相当于状态的产生值。

（3)状态之间以概率的形式转换，称为状态转换概率，定义为 $a_{ij}=P[q_{t+1}=S_j \mid q_t=S_i]$，$1 \leqslant i, j \leqslant N$，状态转换矩阵用 A 表示。

（4）观察值由所属的状态以不同的概率产生，称为观察值产生概率，定义为 $b_j(k)=P[O_k \mid q_t=q_m]$。状态和观察值间的关系，用观察值产生概率矩阵来表示，用 B 表示。

（5）在时刻 t=1 时，每个状态不同的初始概率，定义为 $\pi_i=P(q_1=S_i)$

$1 \leqslant i \leqslant N$，初始状态概率向量，用集合 π 表示。

下面以一个例子来进一步说明隐马尔科夫模型的概念。当不能通过观察天气变化来预测天气情况时，就可以根据观察到的植物海藻的表象来预测天气的变化，也就是说，天气的状态和水藻的状态密切相关。此时就有两组状态：观察状态(水藻的状态)和隐含的状态（天气状态）。因此，希望得到一个算法通过水藻和马尔科夫状态，在没有直接观察天气的情况下得到天气的变化情况。

天气例子中有两类状态的转移图，假设隐状态是一阶马尔科夫过程描述，因此他们相互连接。如图 3.2 所示：

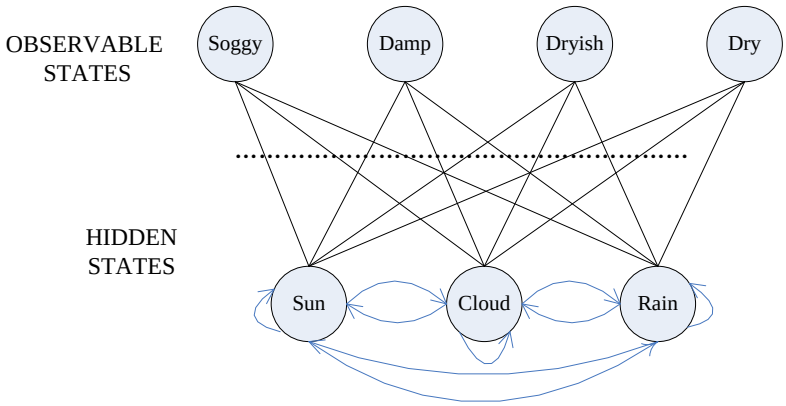

图3.2 隐马尔科夫模型原型

3.2.2 隐马尔科夫模型需解决的三个问题

一旦一个问题可以用隐马尔科夫模型来解释的话，那么就一定需要解决三个问题。

（1）评估：给定模型参数 $\lambda=(N, M, \pi, A, B)$ 及观察序列 $O=\{O_1, O_2, \cdots, O_M\}$ 由此模型如何有效计算此观察序列的概率 $P(O|\lambda)$？

问题 1 就是给定一个模型和一个观测序列，怎样去计算该模型能够以什么样的概率来得到这个观测序列。也可以把该问题看成一个模型和一个给定的观测序列之间的匹配度有多高。

（2）解码：根据给出的一个观测序列及模型 $\lambda=(\pi, A, B)$，如何选择状态序列，使得它最符合观测序列？

问题 2 解码就是尝试要去寻找隐马尔科夫模型中的隐含的部分，即去寻找一个正确的状态序列。当然，大家都明白一点，除了退化的模型，唯一正确的状态序列是不存在的。所以，在解决实际的应用时，一般只有通过寻找最优序列来解决此类问题。

（3）学习：给出一个观测序列如何调整模型的参数 A, B, π，使得 P（O|λ）最大？

问题 3 主要是用来寻找最优模型，使模型能够以最大的概率来得到一个给定的观测序列。这个用来调整模型参数值的观测序列就叫做训练序列，也就是说用来训练隐马尔科夫模型。在解决实际问题时，隐马尔科夫模型的训练问题是最很重要的一个环节，因为根据观测序列要来调整给定的模型的参数直到最优，就相当于创建一个最优的模型。

3.3 BP 神经网络理论

3.3.1 神经网路概述

1982 年，美国物理学家 Hopfield 提出了 Hopfield 模型，其演变过程是一个非线性动力学系统，引入网络能量的概念，作为网络稳定性的判断依据。1986 年，Rumelhart 和 McCelland 提出的误差反向传播算法网络模型，至今仍然影响深远，也是目前最广泛应用的一种网络学习算法。国际神经网络协会 1987 年成立，同年，第一届国际神经网络学术会议在美国圣地亚哥召开，此次会议的召开掀起了神经网络研究的新高潮。现在，神经网络理论已被广泛应用于知识工程、模式识别、信息处理、故障诊断以及控制优化等方面[45]。

神经网络是由大量神经元通过完善的链接构成的自适应非线性动态系统，由许多简单的神经元处理单元通过使用加权的连接相互作用组成，利用实例可以自适应或自学习地形成神经网络中的权函数，以使网络正确理解和解决特定的问题并达到最佳性能。

常见的神经网络模型有感知器网络、线性神经网络、BP 网络、径向基函数

网络、Hopfield 网络、自组织网络等。

3.3.2 BP 神经网路概述

BP 神经网络作为神经网络的一部分，目前广泛应用于函数逼近、模式识别、分类和数据压缩等方面。

BP 神经网络全称是 Back Propagation Neural Network，即反向传播网络。它是把一组样本的输入输出变成一个非线性优化的问题，使用了最优化中最普遍的梯度下降算法，用迭代运算求解权，加入隐节点使得优化问题的可调参数增加，从而可以逼近精确解。是一种多层前向性、采用误差反向传播学习算法的神经网络，其结构如图 3.3 所示：

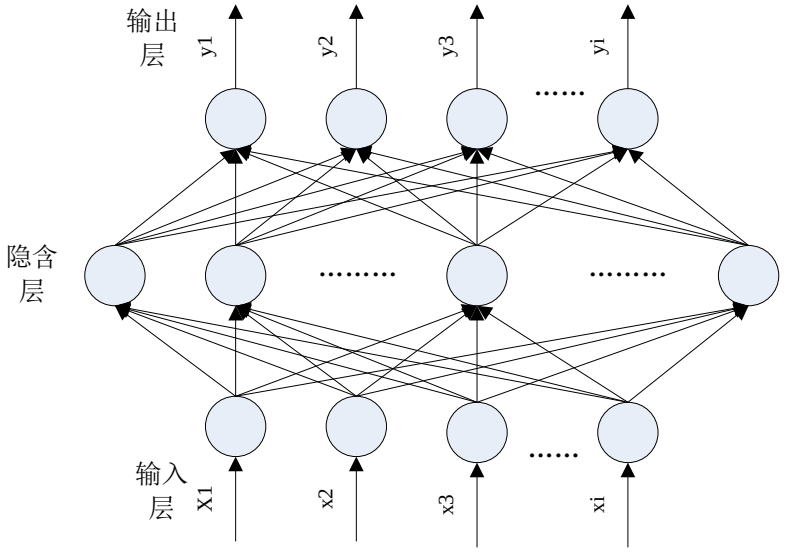

图3.3 BP神经网络结构

从图中可以看出，BP 神经网络拓扑结构包括：输入层、输出层和隐含层三部分，隐含层可以是一层或者多层（上图隐含层是一层结构），并且每层有多个神经元。BP 神经网络的特点是：各层内神经元之间无任何连接、各层神经元仅与相邻层神经元之间有连接、各层神经元之间无反馈连接。输入层接收来自网络外部的信息，然后通过向前传播把信息发送给隐含层结点，经变换函授后，再把隐含结点的信息输出。隐含层不直接接收外界的信号，也不直接向外界发送信号。

一般情况下，隐含层采用的结点变换函数通选取 Sigmoid 型函数，而输出层采用线性激活函数。由于同层节点间无任何耦合，因此，每一层的神经网络元只接受前一层神经元的输入，每一层神经元的输出只影响下一层神经元的输出。

网络的输入数据x=（x_1，x_2，…， x_t）从输入层依次经过各隐含层节点，然后到达输出层节点得到输出数据y=（y_1，y_2，…， y_t）。因此，可以把BP神经网络看成是一个从输入到输出高度非线性映射。BP神经网络通过对样本进行学习，调整BP神经网络中的连接权值，就可以实现非线性分类问题，本文的模型中就是采用BP神经网络来检测攻击行为。

3.3.3 BP 算法

BP 算法是最小均方算法的一种广义形式，它采用梯度搜索技术，按代价函授最小的准则递归地求解网络的权值，代价函授为网络的实际输出和期待输出的均方误差。它的模式识别分为两个不同的阶段：第一个阶段为学习阶段，即网络训练阶段，调整神经网络权值以表现问题域；第二个阶段为工作阶段，权值固定不变，把实际数据或实验数据输入到神经网络，网络能对其进行模式分类。网络训练开始时，初始化的权值为一组随机值，节点预先规定好输出值，当输入训练数据后，代价函数是可计算的，通过 BP 算法，误差逐层向输入层方向逆向传播，当净输出与期望输出不相符合时，误差开始反向传播。反向传播时误差通过输出层，按误度下降的方式修正各层权值，向隐含层、输入层逐层传播。这样不断的循环信息正向传播和误差反向传播过程，使得各层权值不断调整，将误差分摊给各层所有单元，进而达到修正各单元的权值的目的，直到误差的大小在预先可接受的范围内为止，其输出一般取微 S 形函数，即 $f_j(x)=1/(1+e^{-x})$。

3.4 小结

本章首先介绍了隐马尔科夫模型的基本理论及隐马尔科夫模型需要解决的三个问题，然后介绍了神经网络的基本知识，重点是 BP 神经网络。为模型的训练、检测和 BP 神经网络的建立做好了准备。

第四章　基于隐马尔科夫模型和神经网络的入侵检测模型

4.1 引言

在不同的应用领域中，人们提出了不同的 HMM 结构，在实际的应用场合，根据不同的情况可以建立不同的 HMM 模型，隐马尔科夫模型具有算法成熟、易于训练等优点，对于隐马尔可夫模型在入侵检测中的应用，已经有不少研究人员做过这方面的工作。但是它有一个明显的缺陷是：数据库建立的过程是很费时费力的且不易收敛。例如：文献[46]利用系统调用序列作为特征来建立系统正常用户行为模型，以系统调用作为 HMM 的观测值，以程序中出现的系统调用总数作为 HMM 的状态数，但由于系统调用数目过大和训练用的系统调用序列过长，导致 HMM 模型训练时间长且不易收敛。文献[20]把隐马尔科夫模型应用于入侵检测，并给出了一种观察值的方法，如文献[23]从攻击意图上把入侵检测和隐马尔可夫模型结合起来，文献[47]提出了一种基于隐马尔科夫模型的复合攻击检测方法，但他们共同的缺点都是仅局限于具体的攻击行为上，没有通用性。文献[48]虽然取得了不错的效果，但是也是从系统调用的角度去考虑问题的。因此就有学者将神经网络应用在入侵检测上面来了，主要是借助神经网络优异的模式识别能力，如文献[26][27]都是把神经网络理论应用于入侵检测,但是神经网络用在入侵检测方面的检测也有其缺点就是学习速度慢。

鉴于这个原因，本文避免隐马尔科夫模型和神经网络各自带来的不足，在了解隐马尔科夫模型和神经网络理论的基础上，借鉴上述文献的思想，利用隐马尔科夫模型易于训练和神经网络的优异的模式识别能力，将隐马尔科夫模型和神经网络结合起来，提出了一种基于隐马尔科夫模型和神经网络的入侵检测系统模型，该模型最主要的优点就是：具有较高的检测率。

4.2 基于隐马尔科夫模型和神经网络的入侵检测模型的建立

在借鉴了隐马尔科夫模型的状态转移原理和 BP 神经网络调整网络中的链接权值就可以实现非线性分类的原理，本文提出了一种基于隐马尔科夫模型和神经网络

的入侵检测模型，如图 4.1 所示：

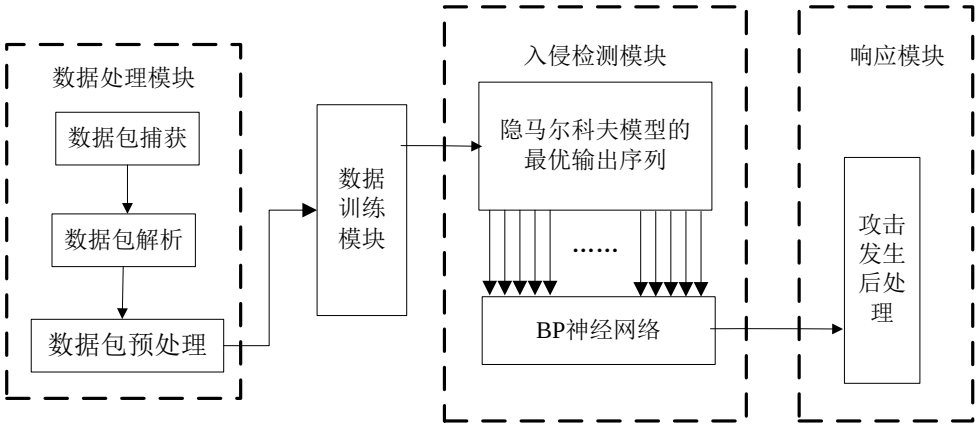

图4.1基于隐马尔科夫模型和神经网络的入侵检测模型

4.2.1 模型的工作原理

该入侵检测模型主要由数据处理模块、数据训练模块、入侵检测模块和响应模块四大部分组成。其中模型的核心是入侵检测模块，也是本文的创新点之一。入侵检测由两部分组成:第一个部分是隐马尔科夫模型，另外一部分是神经网络。

入侵检测模型的工作原理如下：首先把经数据处理模块处理后的数据（包括数据包捕获、数据包解析及数据包预处理将其转化为隐马尔科夫模型可识别的输入）来训练隐马尔科夫模型,训练好之后的模型就可以用来检测入侵行为和异常行为了，需要注意的是：在这里数据训练模块是在 BP 神经网络之外的，这里主要应用的是隐马尔科夫模型易于训练的特点。但是为了提高检测率，把隐马尔科夫模型的输出作为神经网络的输入，对状态序列进行了二次检测，神经网络的输出才是期望的真正的输出，最后根据神经网络的输出给出相应的响应。

4.2.2 隐马尔科夫模型中观察值的确定方法

HMM 应用在网络入侵检测方面的最明显的困难就是 HMM 的观测值很难确定，好的参数选择可能使得计算的效率较高,而观察值选择的不当可能导致训练时间很长，甚至不能完成训练。文献[19]是使用“三次握手”确定了 TCP 数据包的观察值，而

文献[49]是使用标志位加权和来确定观察值。本文在了解第二章 TCP/IP 协议模型及几种针对协议的基本理论的基础上并结合 TCP/IP 管理模型、流量控制原理以及协议自身的特点，提出了一种确定隐马尔可夫模型观察值的方法。

Moore 等人通过对被攻击者发送的响应包进行分析，得出一个著名结论:大多数攻击使用 TCP 包(94%)，然后是 UDP 包(2%)和 ICMP 包(2%)[50]。因此，本文从协议类型的角度只研究 TCP、UDP 和 ICMP 协议数据包。

1、 TCP 数据包

TCP 状态检测是基于网路的入侵检测中重要的一环，各种基于网络协议漏洞的入侵方式都可以通过 TCP 状态检测来早起发现，当 TCP 连接状态转换不符合 TCP 状态装换图，就发生了不正常现象，TCP 连接从开始建立连接到连接释放各状态如图 4.2 所示：

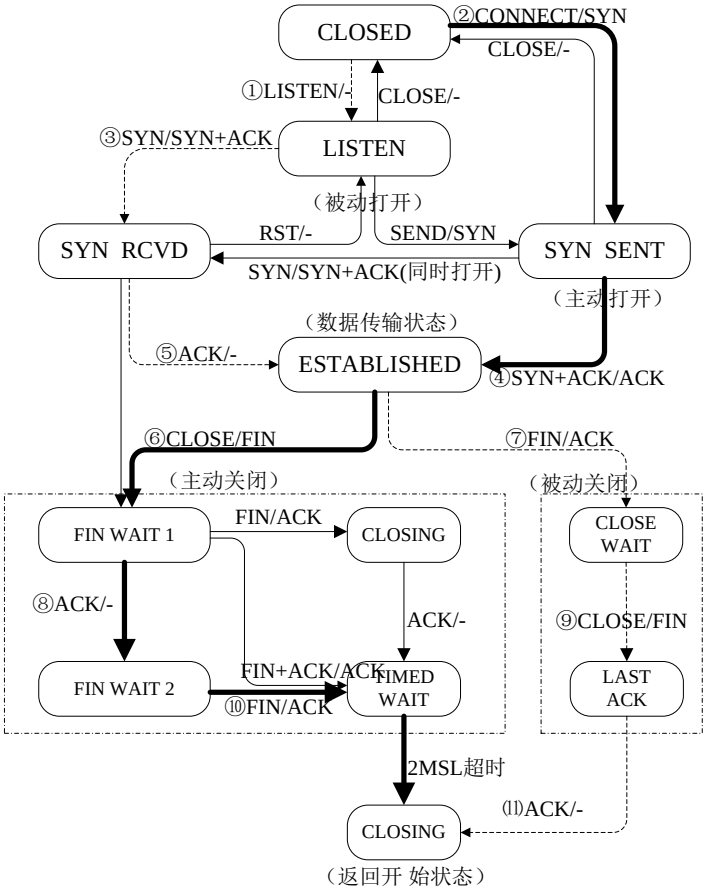

图4.2 TCP连接管理有限状态机

在 TCP 管理模型中，建立和释放连接所要求的步骤都可以用一个有限状态机来表达，每一种状态中，都存在一些合法的事件，当合法事件发生时，可能采取某个动作，当其他事件发生的时候，则报告一个错误。

图中用粗黑线表示一种常见的情形：一个客户机主动的连接到一个被动的服务器上面，虚线表示服务器部分，细黑线表示不正常事件。每条线都被标记了一对“事件/动作”，这里的事件可以是用户发起的系统调用（connect，listen，send 或者 close），也可以是一个数据段的到达事件（SYN，FIN，ACK，或者 RST），或者也可以是两倍最大分组生存期的超时事件；而动作可能是发送一个控制数据段（SYN，FIN 或者 RST），或者什么都不做。

在本文中把每一个事件到达认为是一个数据段的到达，当客户机器上面的一个应用程序发出 connect 请求连接的时候，也就是创建一个 TCP 连接（SYN=1，ACK=0），并将它标记为 SYN SENT 状态，然后发送一个 SYN 数据段。在一台机器上面可能同时有许多个连接处于打开状态，它们代表了多个应用程序。所以，状态是针对每一个连接的。

当接收到 ACK 的时候，就开始进入传送数据 ESTABLISHED 状态（ACK=1，SYN=0），现在就可以发送和接受数据了。当传输结束的时候，它执行 close 原语，从而 TCP 发送一个 FIN 数据段，当 ACK 到达的时候，发生一次状态转移，进入下一个状态 FIN WAIT2，而且连接的一个方向现在被关闭。当另一方也关闭的时候，一个 FIN 数据段会到来，然后它被确认，现在双方都已经关闭了，但是 TCP 要等待一段最大分组生存期的时间，以确保该连接的所有分组都已经消失，以防发生确认被丢失，当定时器到期之后，TCP 释放该链接（ACK=1，FIN=1）。

从服务器的角度来看一下连接管理的情况，服务期执行 LISTEN，并等待有人连接上来。当一个 SYN 进来的时候，它被确认，并且服务器进入到 SYN RCVD 状态（SYN=1，ACK=1），此时服务器接受连接。服务器也可以发送 RST 信息拒绝连接（RST=1，ACK=0）。若服务器确认了连接，而又发送了 RST 数据报，说明连接出错（ACK=1，RST=1），对于错误连接需要释放，这时等待 FIN 的到来，若到来则释放错误连接（ACK=1，RST=1，FIN=1）。若服务器接受了连接并且它的 SYN 被确认，那么服务器进入 ESTABLISHED 状态，从现在开始双方可以传输数据了。

当客户端完成的时候，数据传输完毕，它执行 CLOSE，从而发送一个 FIN 到服务器，服务器接收到信号，它也执行 CLOSE，TCP 给客户发送一个 FIN 数据段，当客户的确认回来的时候，服务器释放该连接，并删除相应的记录。

至此 TCP 建立连接和释放的所有步骤表达完毕，接下来了解流量控制原理。如图 4.3 所示：

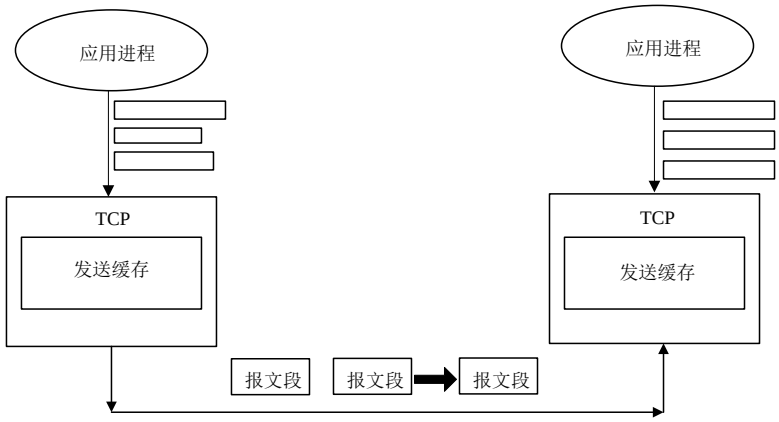

图4.3 TCP流量控制

当数据报发送到发送缓存，然后以报文段的形式到接收缓存，而接收缓存是在缓冲区满了以后，才把数据发送给应用程序，这时如果发送的某个数据报为了提高效率，需要先处理，也就是说接收方接收到数据报后不是先缓存而是直接交给应用程序处理，这里就需要设 PSH 标志位，（PSH=1，ACK=1）来紧急传输数据报。既然有了紧急传输，那么一定要采用紧急释放（ACK=1，PSH=1，FIN=1），才能起到提高效率的目的，要不然只是传输而没有释放，那么可能造成 TCP 拥塞。

由上面的分析可以看出，对 TCP 数据包标志位进行学习和检测的隐马尔可夫模型有 9 个观察值。根据 Forrest 对隐马尔可夫模型的研究表明，隐马尔可夫模型状态数粗略地等于观察值数目时，模型训练到收敛需要的时间最少[51]。因此，状态数可以粗略的等于不同观测符号的个数，即 N=9。

2、 UDP 数据包

UDP 是（User Datagram Protocol）用户数据包协议，它是一个无连接的协议，UDP 协议进行数据传输时，无需预先建立连接、不保证数据报文一定能传送到目的地，这样使得传输过程得到很大的简化。

在这里只提一下UDP没有做到的事情，UDP并不考虑流控制、错误控制，在收到一个坏的数据段之后它也不重传。所有这些工作都留给用户进程。它只利用端口的概念将数据段解复用到多个进程中，这就是它做的全部工作，所以对UDP数据包进行学习和检测的隐马尔可夫模型有2个观察值，即端口号为53和101两种。UDP协议传输示意如图4.4所示：

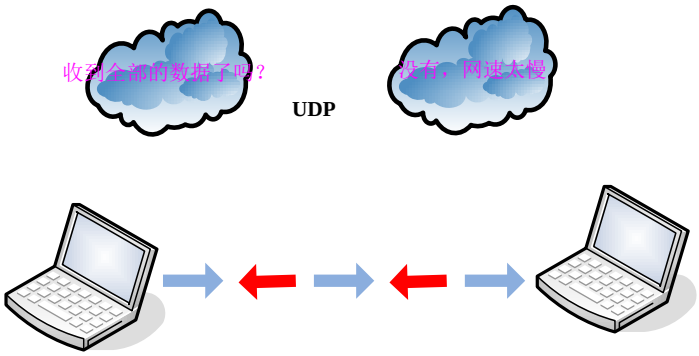

图4.4 UDP协议传输示意图

对于那些需要精确地控制分流，或者需要错误控制，或者需要时间控制的应用程序来说，UDP提供的只是“医嘱”,UDP尤其适用的一个领域是在客户—服务器的情形。

从上图可看出，客户给服务器发送一个短的请求，并且期望得到一个短的应答回来。如果这里的请求或者应答丢失的话，客户就会超时，于是它就只要重试即可，在网络上面只要两条消息就够了。根据 Forrest 对隐马尔可夫模型的研究表明，状态数可以粗略的等于不同观测符号的个数，即 N=2。

3、ICMP 数据包

lCMP 是（Internet Control Message Protocol）Internet）控制报文协议，它是一种面向连接的协议，用于在 IP 主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递、网络安全都起着极其重要的作用，是一个非常重要的协议，尤其是当要对网路连接状况进行判断的时候。当遇到 IP 数据无法访问目标、IP 路由器无法按当前的传输速率转发数据包等情况时，会自动发送 ICMP 消息，它是一种差错和控制报文协议，不仅用于传输差错报文，还传输控

制报文。其报文格式如表 4.1 所示：

表 4.1 ICMP 报文格式

类型（Type）8 位	代码（Code）8 位	效验和　16 位
其内容视类型而定		
数据		

但要说明的是：ICMP 有几种类型的报文（目标不可达报文、重定向报文、超时报文、回送请求和回送应答等），类型（Type）域用于说明 ICMP 报文的作用及格式，（Code）域用于详细说明某种 ICMP 报文的类型，所有数据都在 ICMP 头部后面。RFC 定义了 13 种 ICMP 报文类型。如表 4.2 所示：

表 4.2 ICMP 报文类型

类型代码	类型描述
0	响应应答（ECHO-REPLY）
3	不可到达
4	源抑制
5	重定向
8	响应请求（ECHO-REQUEST）
11	超时
12	参数失灵
13	时间戳请求
14	时间戳应答
15	信息请求（*已作废）
16	信息应答（*已作废）
17	地址掩码请求
18	地址掩码应答

从上表中看到，ICMP 报文有 13 种报文格式，在这里不一一讲解，只考虑响应请求和回送响应报文，做为 ICMP 的隐马尔可夫模型的观察值。其它 ICMP 报文不作处理。

日常使用最多的ping就是响应请求(Type=8)和应答(Type=0)，若向主机某个节点发送Type=8 ICMP报文，途中没有异常(例如被路由器丢弃、目标不回应ICMP或传输失败)则目标返回Type=0 ICMP报文。

因此，对 ICMP 数据包进行学习和检测的隐马尔可夫模型有响应请求和响应回送 2 个观察值。根据 Forrest 对隐马尔可夫模型的研究表明，状态数可以粗略的等于不同观测符号的个数，即 N=2。

4.2.3 滑动窗口大小的确定

在模型训练好以后，这时可以用 P（$O|\lambda$）或者计算最优状态序列来检测是否入侵。本文使用的是后者，即利用 Viterbi 算法输出最优状态序列，找到最优状态序列，作为神经网络的输入，但由于这些序列长短不一，所以文中使用了滑动窗口从序列上面滑过，这样就可以得到具有固定长度的序列，然后把这些短序列输入到神经网络中（神经网路输入元个数和滑动窗口的宽度一致）。每计算完一个短序列，窗口就向前推进一个单位，而这个窗口到底设为多少，只有通过不断地实验和分析，才能得出。本文中输入神经元取任意值 10、15、20、23 四个值时，结果如表 4.3 所示：

表 4.3 不同输入层神经元个数下，输出层值的对比

输入层神经元数	输出 1 的个数比例	输出 0 的个数比例	阈值
10	0.724	0.276	0.6
17	0.6991	0.3009	0.6
20	0.7349	0.2651	0.6
23	0.7107	0.2893	0.6

由表 4.3 可知，当输入层神经元个数为 20 的时候，在阈值(本文是按照经验给定的)同为 0.6 的情况下，输出 0 的比例最小（0 表示正常序列），说明检测出异常的比例较大，所以说滑动窗口的大小为 20。

4.2.4 BP 神经网络的建立

建立 BP 神经网络最主要的就是确定网络中输入层神经元个数、隐含层神经元个数、隐含层数以及输出层神经元个数，确定了这些参数也就确定了网络模型。在本文中 BP 神经网络的参数确定情况如下：

1、输入层、输出层神经元个数的确定

神经网络的输出即是期望的输出结果：被攻击或者正常。在这里规定受到攻击时输出为 1，系统正常时输出为 0。因为神经网络的实际输出一般不会为整数 0 或者 1，所以规定当输出值在一定的精度范围内接近 1（或 0）的话就认为此时系统受到了攻击（或者正常），所以输出神经元只有一个。即本文设计的神经网络是一个向量到点的映射。把正常映射到 0，异常进程的系统调用向量映射到 1。

在本文中规定滑动窗口的宽度就是神经网络的输入元个数，故而输入层神经元个数为 20。

2、隐含层个数的确定

隐含层起抽象的作用，即它能够从输入样本中提取特征。增加隐含层可以增加神经网络的处理能力，但是同时也会增加训练的复杂度和训练时间。此外，增加隐含层个数时，网络在学习过程中会更容易陷入局部极小点而无法摆脱，致使网络的误差调整难以达到全局最小的误差，严重影响学习效果。

关于隐含层数对网络处理能力的影响，Kolmogorov 定理证明对于任何一个在闭区间的连续函数都可以用包含单个隐含层的 BP 神经网络逼近；1988 年 Cybenko 指出一个隐含层就可以实现任意判决分类问题，如果要表示输入图形的任意输出函数，也只需两个隐含层就可以了[52]。本文由于只是想利用神经网络的判决分类的特性，本文涉及的问题较简单，所以没有必要使用复杂度比较高的多隐含层结构，采用单隐含层结构就够了。

3、隐含层神经元个数的确定

隐含层神经元个数的确定比起确定隐含层的个数来说要复杂很多，因为隐含层神经元的个数的多少都会影响整个网络最终的输出结果。减少隐含层神经元个数时，网络从样本中获取信息的能力就差，难以概括和体现学习过程中样本所具有的规律。

增加隐含层神经元个数时，可能使网络学习与某些需掌握的知识不相关的东西，不但会增加网络学习的时间，同时也降低了网络的检测能力。文献[53]提出了通采用 PLD 算法确定神经网络中隐含层神经元的数目，但是其缺点是必须有特定的环境，不具有普遍性。

所以到目前为止没有明确的隐含层神经元确定的方法，大部分人都是根据经验来确定的。正如 1990 年 R.C.Eberhart 等人在《神经元网络 PC 工具》一书中指出：“这是一种艺术”[54]。

文献[55]给出了一种确定隐含层神经元个数的经验公式，这个经验公式是由 BP 神经网络的研究者在大量实践的基础上，针对隐含层的结构特点提出的，目前较为常用的有 2 个：

$$A=\sqrt{a+b} \tag{4.1}$$

$$A=\frac{a+b}{2} \tag{4.2}$$

（注：A 是隐含层神经元个数，a 是输入层神经元个数，b 是输出层神经元个数）

在本系统中，根据以上经验公式（4.2）和经验得知，隐含层神经元个数为 10。所以 BP 神经网络的结构如图 4.5 所示：

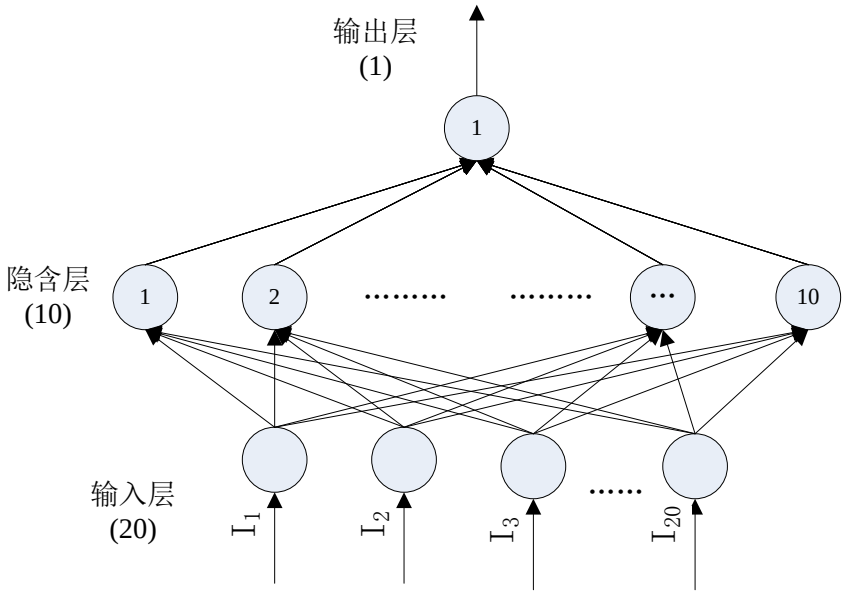

图 4.5 20*10*1BP 神经网络的确定

4.3 入侵检测模型的训练

模型的训练程度很大程度决定了检测模型最终的检测率，在本模型中训练部分是使用了隐马尔科夫模型中的 forward 算法和 Baum-Welch 算法来训练模型。Baum-Welch 算法 (Baum,1972)它采取递归的思想，使得 P (O| λ)局部最大。

Baum-Welch 算法也称前、后向算法，是对一组观测值序列进行训练模型,然后建立用于检测的隐马尔可夫模型，模型训练的流程图如图 4.6 所示：

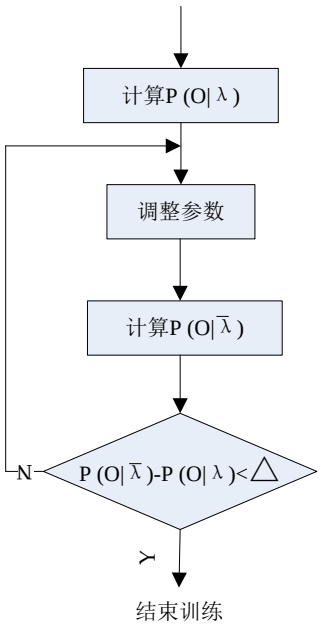

图 4.6 模型训练流程图

在模型训练时，需要先读取观察序列文件中的观察值参数，然后初始化 t=1 时刻所有的转移的局部概率：$\alpha_1(\mathrm{i})=\pi_{\mathrm{i}}\mathrm{b}_{\mathrm{i}}(\mathrm{o}_1)$ (4.3)

得到 t=1 时刻的概率后，要想计算 t+1 时刻在状态 j 时的概率时有：

$$\alpha_{\mathrm{t+1}}(\mathrm{j})=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]_z b_{\mathrm{ij}}\ (o_{i+1}) \quad (4.4)$$

上式表示 t+1 时刻在状态 j 时的概率为前一时刻所有状态的概率与相应的转移概率的积，t+1 时刻的概率只和 t 时刻有关。因此公式 4.4 采用的递归的思想，可以

计算每个时刻的局部概率。

直到得到观察序列的概率：$P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i)$ (4.5)

计算 $P(O|\lambda)$ 的算法如下：

1. $t \leftarrow 1$
2. for $i \leftarrow 1$ to N
3. $\alpha_1(i) \leftarrow \pi_i * b_i(o_1)$
4. for $t \leftarrow 1$ to T
5. for $i \leftarrow 0$ to N
6. $\alpha_t(i) \leftarrow 0$
7. for $j \leftarrow 0$ to N
8. $\alpha_{t+1}(i) \leftarrow \alpha_{t+1}(i) + \alpha_t(j)\alpha_{ij}$
9. $\alpha_{t+1}(j) \leftarrow \alpha_{t+1}(j) b_{t+1}(o_{t+1})$
10. $pprob \leftarrow 0$
11. for $i \leftarrow 0$ to N
12. $pprob \leftarrow \alpha_T(i)$
13. print pprob

由公式 4.5 得到的 $P(O|\lambda)$ 其实只是给定模型参数 $\lambda=(N, M, \pi, A, B)$ 及观察序列 $O=\{O_1, O_2, \cdots, O_M\}$ 得到的概率值，训练模型时，就需要根据观测序列 O 和选取的初始模型 $\lambda=(\pi, A, B)$ 不断的调整参数 $\bar{\pi}_i$，$\bar{a}_{ij}$，$\bar{b}_{jk}$，这就用到著名的 Baum-Welch 重估公式：

$$\bar{\pi}_i = \gamma_1(i) = \gamma_1(i) = p(q_t = S_i | O, \lambda) \quad (4.6)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \quad (4.7)$$

$$\bar{b}_j(k)=\frac{\sum_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)} \tag{4.8}$$

其中$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j)$表示 t 时刻，从 i 到 j 的转移概率

$\sum_{i=1}^{t-1}\gamma_t(i)$=整个过程中从状态 S_i 转出的次数（number of time）的预期

$\sum_{i=1}^{t-1}\xi_t(i,j)$=从 S_i 跳转到 S_j 次数期望值

由公式(4.6)、(4.7)、(4.8)就可以得到新的模型，$\bar{\lambda}=\left(\bar{\pi},\bar{A},\bar{B}\right)$，重新利用前向算法就可得到一个新的 P（0|$\bar{\lambda}$），该算法的发明者 Baum-Welch 指出，一般情况下，P（0|$\bar{\lambda}$)都会大于 P(0|λ)。因此，重复这个过程逐步改进模型参数，直到 P(0|$\bar{\lambda}$)- P(0|λ)<⊿，⊿为给定的一个足够小的数，此时的$\bar{\lambda}$就是所求的模型。

重估$\bar{\pi}_i$，$\bar{a}_{ij}$，$\bar{b}_{jk}$的算法如下：

//重估$\bar{\pi}_i$

1. for $i\leftarrow 1$ to N-1
2. $\bar{\pi}_i\leftarrow\gamma_0(\mathrm{i})$

//重估$\bar{a}_{ij}$

3. for $i\leftarrow 1$ to N
4. for j← 1 to T
5. $num\leftarrow 0$
6. $den\leftarrow 0$
7. for t ←1 to T-1

8. $num_{ij} \leftarrow num_{ij} + \xi_t(i,j)$

9. $den_i \leftarrow den_i + \gamma_t(i)$

10. $\bar{a}_{ij} \leftarrow \frac{num_{ij}}{den_i}$

//重估 $\bar{b}_{jk}$

11. for k ← 1 to N
12. for t ← 1 to T
13. $numb \leftarrow 0$
14. $denb \leftarrow 0$
15. for j ← 1 to M
16. $if(o_t == k)$ then
17. $numb \leftarrow numb + \gamma_t(j)$
18. endif
19. $denb \leftarrow denb + \gamma_t(j)$
20. $\bar{b}_{jk} \leftarrow \frac{numb}{denb}$

对参数重估完了以后，下一步就是利用迭代的方法，判断 $P(O|\bar{\lambda}) - P(O|\lambda) < \Delta$ =0.001, 本文中给定⊿为 0.001，当满足条件时，结束训练，$\bar{\lambda}$ 为需要的模型。

21. 初始化 hmm
22. $p_1 \leftarrow$ forward(hmm)
23. do
24. $p_2 \leftarrow p_1$
25. hmm.a ← $\bar{a}_{ij}$
26. hmm.b ← $\bar{b}_{jk}$
27. hmm.π ← $\bar{\pi}_i$
28. $p_1 \leftarrow$ forward(hmm)

29. while($(p_1 - p_2) \geq \Delta$)

30. print p_1

4.4 入侵检测算法

入侵检测是模型的核心部分，也是本文的另一创新点。入侵检测算法设计的好坏，决定了系统的检测率，本文利用隐马尔科夫模型的 Baum-Welch 算法训练模型，模型训练好以后，利用 viterbi[56]算法会得到由 N 种状态表示的最优状态序列，然后利用滑动窗口理论把最优状态序列划分成一系列具有固定长度的短序列。最后，把这些短序列作为神经网络的输入，利用 BP 算法通过 S 变换函数输出期望的结果。

对神经网络的输出，用 0 和 1 表示输出，0 表示正常，1 表示入侵。对于一个最优状态序列，如果它输出 1 的比例超出了预先给定的阈值，则认为是一个入侵，反之则认为正常。当然阈值的取值，很大程度上影响了检测率，本系统经过反复实验，取阈值为 0.6（也就是当输出 0 的比例超过 60%，则认为正常，否则为入侵）。具体的入侵检测的流程图如图 4.7 所示：

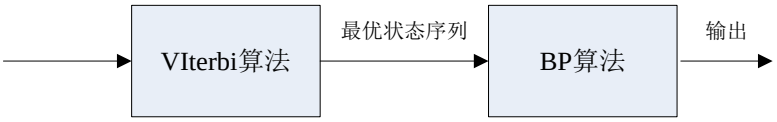

图4.7 入侵检测流程图

首先寻找最优状态序列，需要先定义$\delta_t(j)$，$\delta_t(j)$表示 t 时刻状态 S_i 的最优状态序列的概率，其表达式如下：

$$\delta_t(j) = \max_{q1\cdots q_{t-1}} p[q_1,\cdots q_{t-1},\ q_t = S_i,\ O,...O_t|\lambda] \tag{4.9}$$

初始化 t=1 时刻的局部概率：

$$\delta_1(i) = \pi_i b_j(o_1) \tag{4.10}$$

$$\psi_1(i) = 0 \tag{4.11}$$

如果 S∈(1,N)，那么在任意时刻，都可能是最优状态序列，所以在每一个状态 S_i 得到$\delta_t(i)$，为了减小计算的复杂度，有递归得：

$$\delta_{\mathrm{t}}(\mathrm{j})=\left[\max_{1\le \mathrm{i}\le \mathrm{N}} \delta_{\mathrm{t\text{-}1}}(i)a_{ij}\right]b_j(o_t) \tag{4.12}$$

$$\psi_t(i)=\left[\arg\max_{1\le i\le N} \delta_{t-1}(i)a_{ij}\right] \tag{4.13}$$

公式（4.13）是用来计算使括号中表达式的值最大的索引的，它是由前一时间 t 步骤的局部概率和转移概率计算的。

最后有结果 $\overset{*}{p}=\max \delta_T(i)$ (4.14)

$$\overset{*}{q}_T=\arg\max_{1\le i\le N} \delta_T(i) \tag{4.15}$$

所以得到的最优状态序列是：$\overset{*}{\mathrm{q}}_{\mathrm{t}}=\psi_{\mathrm{t+1}}\left(\overset{*}{\mathrm{q}}_{\mathrm{t+1}}\right)$ (4.16)

最优序列的算法如下：

//初始化 t=1 时刻的局部概率

1. $t\leftarrow 1$
2. for i ← 1 to N
3. $\delta_1(i)\leftarrow \pi_i * b_j(o_1)$
4. $\psi_0(\mathrm{i})\leftarrow 0$

//递归

5. for t ←1 to T-1
6. for j←1 to N
7. $\max\delta_{t\text{-}1}(\mathrm{j})\leftarrow 0.0$
8. $\max\delta_{\mathrm{t\text{-}1}}(\mathrm{i})\ \mathrm{a_{ij}}\leftarrow 1$
9. for i ← 1 to N
10. $\delta_{\mathrm{t}}(i)\leftarrow \delta_{t-1}(i)* a_{ij}$
11. if（$\delta_{t\text{-}1}(\mathrm{j})>\delta_{\mathrm{t\text{-}1}}(\mathrm{i})\mathrm{a_{ij}}$）then

12. $\max \delta_{t-1}(j) \leftarrow \delta_{t-1}(j) a_{ij}$

13. $\max \delta_{t-1}(i)\ a_{ij} \leftarrow i$

14. endif

15. $\delta_t(j) \leftarrow \max \delta_{t-1}(j) a_{ij} b_j(o_t)$

16. $\psi_t(i) \leftarrow \arg\max \delta_{t-1}(i) a_{ij}$

//计算最后结果

17. $pprob \leftarrow 0$

18. $\chi_{T-1} \leftarrow 1$

19. for i ← 1 to N

20. if($\delta_{(t-1)}(i) \succ pprob$)

21. $pprob \leftarrow \delta_{t-1}(i)$

22. $\chi_{t-1} \leftarrow i$

//输出最优状态序列

23. for t ← T-1 to 1

24. $\chi_t \leftarrow \psi_{t+1} \chi_{t+1}$

25. for x ← 0 to T

26. print χ_x

有了最优状态序列之后，就把它利用滑动窗口的工作原理把它划分成小序列，作为 BP 神经网络的输入，神经网络的输出就是期望结果，本文中的神经网络已经在前面确定了参数：是一个具有 20 个输入层神经元个数、10 个隐含层神经元个数以及 1 个输出层神经元个数的三层结构的 BP 神经网络。

BP 神经网络的工作原理：逐一根据输入序列计算出实际输出和误差测度，对各权值做一次调整，重复这个循环，直到总的误差测度小于预先给定的值，循环结束，最后输出结果，流程图如 4.8 所示：

在神经网络中神经元的数学模型为：

$$y = f\left(\sum_{i=1}^{m} w_i \chi_i - \theta\right) \tag{4.17}$$

其中，χ_i 为输入向量、y 为输出、w_i 为权系数、θ 为阈值、f(x)为激发函数，本文中用到的是 S 型函数 $f(x) = \frac{1}{1+e^{-x}}$。

因为本文中最优序列是经过滑动窗口分段后的，才输入神经网络中的，所以使用了改进的神经网络算法(消除样本顺序影响的 BP 算法)。E 表示误差测度、w 表示权值、L 表示样本的长度。

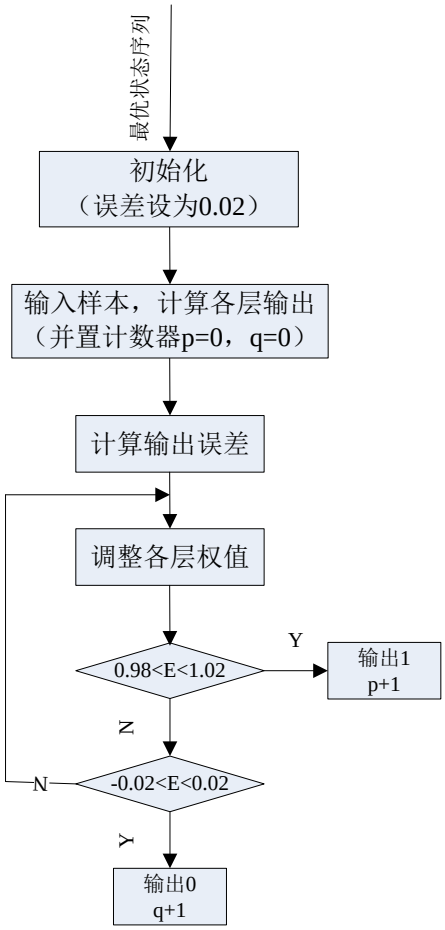

图 4.8 BP 神经网络工作流程图

算法的核心步骤如下：

1. for k←1 to L

2. 初始化 w^k

3. 初始化精度控制参数 ε

4. E= $\varepsilon+1$

5. while E> ε

6. $E \leftarrow 0$

7. $w_{ij}^{k} \leftarrow 0$

8. 对每一个样本(χ_p， y_p)

9. 计算 χ_p 对应的输出 o_p

10. 计算 E_p

11. $E \leftarrow E+ E_p$

12. 计算 $\Delta_p w_{ij}^{l}$

13. $\Delta w^{l}{}_{ij} \leftarrow \Delta w^{l}{}_{ij} + \Delta p w_{ij}^{l}$

14. $k \leftarrow L-1$

15. while $k \neq 0$

16. 计算 $\Delta p w_{ij}^{k}$

17. 计算 $\Delta w^{k}{}_{ij} \leftarrow \Delta w^{k}{}_{ij} + \Delta p w^{k}{}_{ij}$

18. $k \leftarrow L-1$

19. 计算 $w^{k}{}_{ij} \leftarrow w^{k}{}_{ij} + \Delta w^{k}{}_{ij}$

20. $E \leftarrow E/2.0$

21. if（0.98<E<1.02）then

22. $p \leftarrow p+1$

23. Print 1

24. if（-0.02<E<0.02）then

25. $q \leftarrow q+1$

26. Print 0

在得到每一个序列经神经网络输出后，需要把输出结果0和1的个数进行统计，0 表示正常、1 表示入侵（接近 0 或者 1）。如果 0 或者 1 的个数超过了给定的阈值(0.6)的话，就认为是一次入侵，否则正常。

4.5 入侵响应算法

响应就是根据神经网络的输出来判断系统是否受到了攻击，如果受到攻击应该做哪些事情阻止攻击的继续进行的方法。其实响应可以分为报警响应、人工响应和自动响应三种，报警响应就是只是对入侵进行报警而不做其它操作，它是属于被动响应的一种。人工响应是根据入侵人为做的一些操作如：重新配置路由器和防火墙、重建攻击过程。自动响应是目前比较流行的一种响应方式，因为入侵响应的决策过程很大程度上在于对事件详细分析的基础上，不同的事件应该采用不同的响应措施，这就要求对同一事件要从不同的角度分析，建立一种自适应的响应系统。本文中就是用了最简单的被动报警方式：声音报警，即受到攻击时，播放告警声音。具体的流程如图 4.9 所示（注：q 表示输出 0 的个数，p 表示输出 1 的个数）：

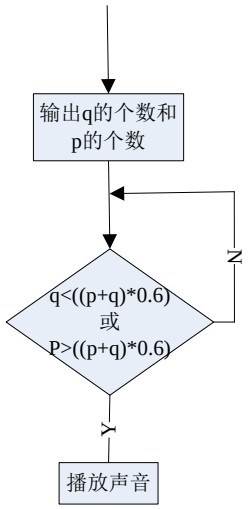

图4.9 响应流程图

文中响应算法的思想是：输出1和输出0的个数之和的60%要是小于输出1的个数，或者输出1和输出0的个数之和的60%大于输出0的个数，就认为是入侵。

核心算法如下：

1. if（q>（p+q）*0.6）||（p>(p+q)*0.6）then
2. music.play

4.6 小结

本章在给出了入侵检测模型的基础上，阐述了模型的工作原理、给出了 HMM 中的观察值的确定方法、设计了 BP 神经网络的结构、利用 HMM 的 forward 算法和 Baum-Welch 算法对模型进行了训练，重点给出了入侵检测算法，最后给出了模型的响应算法。

第五章　模型的实现及实验结果分析

在上一章中，提出了一个基于隐马尔科夫模型和神经网络的入侵检测模型，本章的主要目标是设计一个入侵检测系统来实现上一章提出的模型。

5.1 入侵检测模型中算法之间的关系

本文所研究的基于隐马尔科夫模型和神经网络的入侵检测模型主要利用隐马尔科夫模型成熟的算法和神经网络优异的模式识别能力来设计的。

首先捕获网络中传输的所有数据包，从中提取感兴趣的数据包。然后，训练模块对捕获的数据包通过隐马尔科夫模型的 Baum-Welch 算法来进行训练，模型训练好之后通过 Viterbi 算法，输出具有 N 种不同符号组成的最优序列。这些序列可以很好的判断是否为攻击。但是为了提高检测率，利用神经网络优异的分类特性，对输出的状态序列进行进一步的判断，即把隐马尔科夫模型的输出作为神经网络的输入，通过 BP 算法使得最后的输出结果为期望结果，以此来达到进一步判断入侵，模型所包含的算法之间的关系如图 5.1 所示：

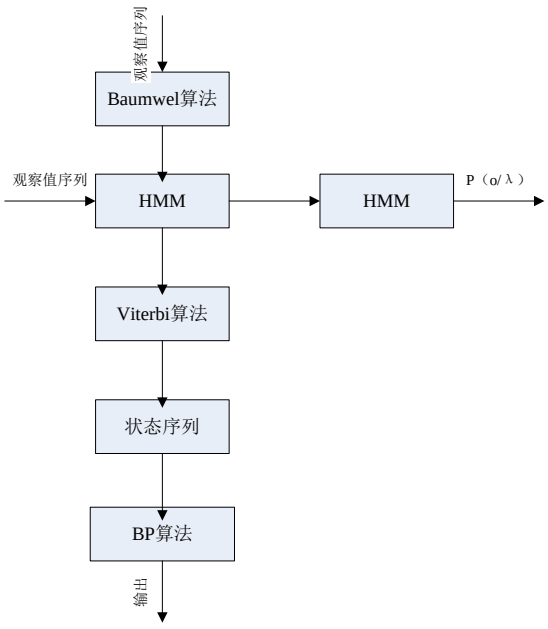

图5.1 算法之间的关系图

5.2 开发环境

开发环境为 Java 语言平台（JDK1.6.2），Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言 Java 平台的总称，是一种面向对象的程序设计语言，本文用它作为开发语言主要因为它具有如下特点：

（1）Java 语言是简单的。Java 丢弃了令人迷惑的一些操作如：操作符重载、多继承、自动的强制类型转换。特别地，Java 语言不使用指针。

（2）Java 语言是分布式的。Java 语言支持 Internet 应用的开发，它提供了用于网络应用编程的类库，包括 URL、URLConnection、Socket、 ServerSocket 等。

（3）Java 语言是可移植的。这种可移植性来源于体系结构中立性，另外，Java 还严格规定了各个基本数据类型的长度。

（4）Java 语言是多线程的，并提供多线程之间的同步机制（关键字为 synchronized）。

5.3 入侵检测模型的实现

5.3.1 数据处理模块

数据包处理模块是模型的基础，所有的数据流都要经过该模块才能被识别，该模块主要由数据包捕获、数据包解析和数据包预处理三个部分组成。

1、数据包捕获

到目前为止，有两种比较常用的数据包捕获方法，一种是采用网络数据捕获专用设备；另一种是利用普通计算机与网络连接的通用硬件网络适配器，即网卡来捕获。如果想捕获所有的数据包，就必须绕过系统正常工作的处理机制，直接访问网络底层，一般是将网卡设置为混杂模式(Prmoiscuous Mode)，混杂模式下计算机能够接收所有流经该网段的信息。

数据包捕获作为整个模型的基石，是在 windows 操作系统下面，可以利用 Windows 下比较成熟的伯克利包过滤器 BFP(Berkeley Packet Fitler)机制和网络安全

工具开发函数库 Winpcap，本文是利用 java 中的 Jpcap 工具包进行捕包。

Jpcap 是一个能够捕获、发送网络数据包的 java 类库包，它的整个结构大体上跟 wincap/libpcap 是很相像的，例如 NetworkInterface 类对应 wincap 的 typedef struct _ADAPTERADAPTER，getDeviceList()对应 pcap_findalldevs()等等。Jpcap 所支持的数据包有：ARPPacket、DatalinkPacket、EthernetPacket、ICMPPacket、IPPacket、TCPPacket、 UDPPacket 等。

如果有多块网卡，那么在捕获数据包之前先需要选择网卡，核心代码如下：

```
public NetworkInterface[] getDrives() {     //返回获取的接口列表
        return drives;
    }
    public void setDrive(NetworkInterface drive) {    //初始化当前接口对象
        this.drive = drive;
    }
    public    getdrive(String t_str)       //构造函数
    {
        drives=JpcapCaptor.getDeviceList();
        for(int i=0;i<drives.length;i++)
            if(t_str.equals(i+"、"+drives[i].description))
            {
                drive=drives[i];
            }
       }
```

选择好了网卡，下一步就要选择捕获的数据包类型，在这里为了捕获所有流经该网段的信息，将网卡设置成混杂模式，只捕获 TCP、UDP、ICMP 三种类型数据流中一种或者全部，如图 5.2 所示。核心代码如下：

```
public void run()   //多线程中核心方法，用来启动线程
    {       openmydrive(); //调用本类方法，依次实现对选定端口的监控、记录工作
        while(flag)     //用一个循环结构实现对当前端口数据包的不间断捕获   {
```

```
Packet    pkt=jc.getPacket();    //捕获到数据包并进行处理
 if(pkt!=null && (pkt instanceof TCPPacket || pkt instanceof UDPPacket ||pkt
instanceof ARPPacket ||pkt instanceof ICMPPacket))       {
                                mainfram.textareaadd(pkt);    }
        }
    }
```

图5.2选择网卡和数据包类型

2、数据包解析

数据包解析模块主要是将捕获到的 IP、TCP、UDP、ICMP 数据包进行解析，其通过分析其结构、检查数据包的报头,确定是哪一类型的数据包,从而能提取出该类数据包的特征，本文要用到的数据包解析方法已经在第二章给出了详细的说明，解析流程图如图 5.3 所示：

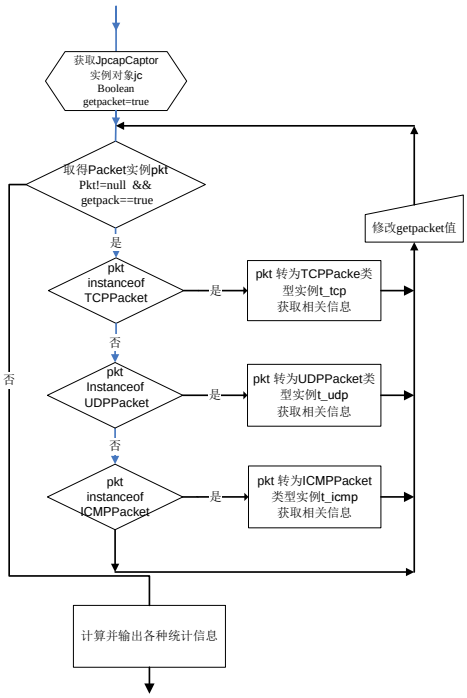

图5. 3 数据包解析流程图

数据包解析后如图 5.4 所示：

图5. 4 数据包解析

（3）数据包预处理

数据预处理也就是把数据格式转换为隐马科夫模型能识别的格式，即对捕获的数据包特征进行编码处理，

(1)TCP 数据包的预处理

在第四章根据有限状态机和 TCP 流量控制原理，确定了 TCP 数据包的观察值和隐含值有 9 种，其 TCP 观察值序列的编码情况如下：

(SYN=1，ACK=0)→a　　请求连接；

(ACK=1，SYN=0)→b　　可以发送和接受数据；

(ACK=1，FIN=1)→c　　释放该链接；

(SYN=1，ACK=1)→d　　服务器接受连接；

(RST=1，ACK=0)→e　　拒绝连接；

(ACK=1，RST=1)→f　　说明连接出错；

(ACK=1，RST=1，FIN=1) →g　　释放错误连接；

(PSH=1，ACK=1) →h　　传输紧急数据报为；

(ACK=1，PSH=1，FIN=1) →i　　释放紧急传输。

（2）UDP 数据包预处理

对于 UDP 数据包，UDP 协议提供不可靠的面向非连接的服务，攻击者一般采取设置源端口和目的端口的方式构造恶意报文，所以分析起来比较简单，就只对数据包的端口进行编码，其编码如下：

如果 UDP 数据包的端口号为 7 或 19 时，编码为 a，否则编码为 b，即 UDP 数据包的端口号不是 7 或 19 时，编码为 b。

（3）ICMP 数据包预处理

对于 ICMP 数据包,攻击者一般只对其报文中 8 位的类型字段和 8 位的代码字段感兴趣，在本文中只考虑 ICMP 报文 13 种类型中的回送请求和回送响应报文，其它 ICMP 报文不作处理，其编码如下：

如果 ICMP 报文类型为回送请求时，编码为 a，否则编码为 b，即 ICMP 报文类型为回送响应时，编码为 b。

5.3.2 训练模块

作为衡量入侵检测系统的性能的一个很重要的部分，训练的程度决定了检测率的高低，具体的算法步骤在第四章已经给出了。训练模块主要是利用隐马尔科夫模型的forward算法和Baum-Welch算法来进行，训练的具体实施就是通过不断的调整参数，使得P（$O|\lambda$）最大。核心代码如下：

```
public class    Reparameters (){
… … … … …
    for(i=1;i<myhmm.n;i++) {
            alpha[0][i]=myhmm.pi[i]*myhmm.B[i][(this.o[0])-1];
        } //估算初始状态分布 t=1 时 π̄i，由于数组下表是从零开始的:
        for(i=1;i<myhmm.n;i++)   {
          dena=0.0;
            for(j=0;j<myhmm.n;j++)
                suma=0.0;}
              for(t=0;t<this.t_t-1;t++) {
                suma+=alpha[t][i]*myhmm.A[j][i];
                  for(j=0;j<myhmm.n;j++)
                    alpha[t+1][j]=suma*(myhmm.B[j][(this.o[t+1])-1]);
                                }}//估算从状态i到状态j的概率 āij
        for(k=1;k<myhmm.m;k++) {
          Sumb=0;
         for(t=0;t<this.t_t-1;t++) {
            if(o[t]==k)
                alpha[t+1][j]=sumb*(myhmm.B[j][(this.o[t+1])-1]); }}
                //估算t时刻出现的o的概率 b̄jk
```

… … … … … …

5.3.3 入侵检测模块

入侵检测模块作为本文的核心，主要分两个部分完成：一是隐马尔科夫模型部分另一个就是神经网络部分。其中隐马尔科夫模型部分是用来输出最优序列的，本文是通过 Viterbi 算法计算观察序列的最优状态序列，然后把最优状态序列作为神经网络的输入，神经网络的输出就是期望的结果。

最优状态序列核心代码如下：

```
//1、initialization       //以下数组行下标0、1、2...依次表示时刻1、2、3...
  for(i=0;i<phmm.n;i++)   {
       delta[0][i]=phmm.pi[i]*(phmm.B[i][o[0]-1]);
       psi[0][i]=0;}
  //2、recursion
  for(t=1;t<T;t++) {
       for(j=0;j<phmm.n;j++)
       {   maxval=0.0;
            maxvalind=1;
            for(i=0;i<phmm.n;i++)
            {   val=delta[t-1][i]*(phmm.A[i][j]);
                 if(val>maxval)
                 {     maxval=val;
                      maxvalind=i; } }
            delta[t][j]=maxval*phmm.B[j][o[t]-1];
            psi[t][j]=maxvalind; } }
 //3、termination
  pprob=0.0;
  q[T-1]=1;
```

```
    for(i=0;i<phmm.n;i++)    {
        if(delta[T-1][i]>pprob)    {
          pprob=delta[T-1][i];
            q[T-1]=i;} }
  //4、path(state sequence)backtracking
    for(t=T-1;t>=0;t--)   {
          q[t]=psi[t+1][q[t+1]]; }
    for(int x=0;x<T;x++)     {
        System.out.print(q[x]);}
```

找到最优状态序列之后，剩下的工作就是把这个输出输入到神经网络了，经BP算法后输出期望的结果。

BP 神经网络定义如下：

```
public class BpNet
{
static int IN_MUN=20;                                          //输入层神经元数目
static int HideN=10;                                           //隐层神经元数目
static int OutN=1;                                             //输出层神经元数
static double Weight_In_Hide[][] = new double[HideN][IN_MUN]; //输入层至隐层权值
static double V_Hide_Out[][] = new double[OutN][HideN];        //隐层至输出层权值
static double YU_HN[] = new double[HideN];                     //隐层的阈值
static double YU_ON[] = new double[OutN];                      //输出层的阈值
static double X[] = new double[HideN]; //隐层的输入，各个隐藏层的神经单元的内积
static double Y[] = new double[OutN];//输出层的输入，各个输出层的神经单元的内积
static double H[] = new double[HideN];          //隐层的输出，s 型函数之后的输出值
static double O[] = new double[OutN];          //输出层的输出，s 型函数之后的输出值
static double d_err[] = new double[OutN];                      /隐层到输出层的误差
static double e_err[] = new double[HideN];                     //输入层到隐层的误差
static double err_m[] = new double[N];                         //第 m 个样本的总误差
```

核心代码如下：

```
        public void    Result()
                {
                    System.out.println("输入层到隐藏层权值：");
                     for(int i=0;i<HideN;i++)
                            for(int j=0;j<IN_MUN;j++)
 System.out.println("Weight_In_Hide["+i+"]"+"["+j+"]"+"="+Weight_In_Hide[i][j]);
                           System.out.println("");
                           System.out.println("隐藏层到输出层权值：");
                            for(int i=0;i<OutN;i++)
                               for(int j=0;j<HideN;j++)
System.out.println("V_Hide_Out["+i+"]"+"["+j+"]"+"="+V_Hide_Out[i][j]);
                             System.out.println("");
                    System.out.println("隐藏层阈值：");
                    for(int i=0;i<HideN;i++)
                                System.out.println("YU_HN["+i+"]="+YU_HN[i]);
                                System.out.println("");
                    System.out.println("输出层阈值：");
                    for(int i=0;i<OutN;i++)
                                System.out.println("YU_ON["+i+"]="+YU_ON[i]);
                                 System.out.println("");
                                   System.out.println("输出层输出值：");
                    for(int i=0;i<OutN;i++)
                                System.out.println("O["+i+"]="+O[i]);
                                        }
```

5.3.4 响应模块

本文的响应模块是：通过调用声音文件而播放一段音乐达到告警的目的，同时将告警事件记入日志中。实现声音报警需要有声音源文件，具体的实现步骤是①定位声音文件的位置②取得声音文件并播放。核心代码如下：

```
String location=sounds;
if(sounds.i('|')>=0) {
//定位声音文件的位置
Int start=i;
if((i=sounds.i('|'，i}，，index))<0){
   location=sounds.substring(start);
      i=start;
      }
      else
        {location=sounds.substring(start.i++);
          }
       }
       If(location.length()>0){
                  music=getmusic(new location(getfile()，location)):
                   music.play();
                       }
```

5.4 系统的封装

本文所设计的“基于隐马尔科夫模型和神经网络的入侵检测模型”经过封装系统可以单独运行在被保护主机上（如图 5.5 所示）。系统实现的具体功能有：捕获、训练、检测、响应等。系统的运行从“捕获”开始，点击“配置”菜单，选择网卡和数据包类型就可以开始捕获了。点击“实时统计”选项卡，系统将执行检测功能，

当攻击行为被发现时，系统将提示发现入侵，并报警。如图 5.6 所示：

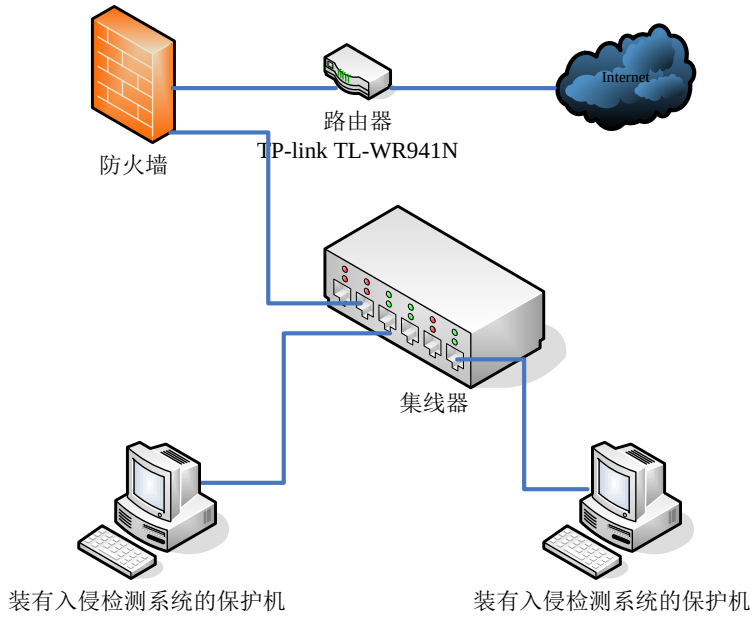

图5. 5 实验拓扑图

图5. 6 系统的检测结果

5.5 实验结果及分析

实验过程主要针对“检测率”这个参数进行，在这里使用了从美国麻省理工大学林肯实验室提供的 DARPA98 作为数据检测源。

由于数据量大，本实验仅选取了部分数据进行测试。为了使实验具有可比性，抽取了 5 种典型的攻击作为本模型的实验数据，五种攻击为 Neptune (SYN Flooding)、Satan、PortSweep、Buffe-overflow、Guess-passwd，本实验选择的攻击包含了攻击的四大类。如表 5.1 所示：

表 5.1 实验中的攻击类别

攻击类别	本实验选择的攻击
Dos	Neptune
Probing	PortsWeep、Satan
U2L	Buffer-overflow
R2L	Guess-passwd

实验的步骤如下：

第一：使用所有数据的 60%来进行训练，这些数据包括入侵数据和正常数据；

第二：训练完毕之后，用另外 40%的数据来测试模型；

第三：输出结果。为了判断入侵，在输出最优序列的时候设置了一个滑动窗口，这样最优序列被划分为若干个固定长度的短序列作为神经网络的输入，然后通过判断从神经网络得到的 0、1 的量来判断是否入侵。即，如果由一个序列所得到的 1 的量超过了事先设定的一个门限值的话，就认为是一个入侵。反之，则认为正常。对于门限，在实验中，都是通过设置不同的值来进行比较得到一系列的检测值，最后的检测结果如表：5.2 所示：

表 5.2 基于隐马尔科夫模型和神经网络的入侵检测模型实验记录

攻击种类	normal	Neptune	Satan	PortSweep	Buffe overflow	Guess passwd
normal	6030	0	20	1	6	2
Neptune	1	3834	22	9	0	0
Satan	2	18	795	0	0	0
PortSweep	0	9	3	165	0	0
Buffe-overflow	6	0	0	0	8	0
Guess-passwd	1463	0	0	0	0	14
（检测率%）	80.25	99.32	94.59	94.3	62.3	75.6

最后本文的实验结果和同样采用了DARPA98作为数据检测源的基于神经网络的入侵检测研究[57]和基于协议的隐马尔科夫模型入侵检测系统研究[58] 做了比较。文献[57][58]中的测试结果分别如表5.3和表5.4所示：

表 5.3 BP 神经网络模型实验记录

Predicted actual	normal	Neptune	Satan	PortSweep	Buffe overflow	Guess passwd
normal	6038	0	20	1	0	0
Neptune	1	3834	22	9	0	0
Satan	2	19	795	0	0	0
PortSweep	0	9	3	165	0	0
Buffe-overflow	11	0	0	0	0	0
Guess-passwd	2183	0	0	0	0	0
检测率	73.3%	99.2%	94.6%	94.2%	0	0

表5.4 隐马尔科夫模型实验记录

攻击类型	检测包个数及检测率
Probe（Portsweep）	3336(91.7%)
Dos（Neptune）	6063(95.2%)
R2L（Guess-passwd）	3649(73.5%)
U2L（Buffer-overflow）	1733(51.5%)
Data	14(25%)
Total	147185(79.5%)

通过表5.2的检测结果可以看出，基于隐马尔科夫模型和神经网络的入侵检测模型对Buffer-overflow和Guess-passwd的检测效果不是很好，主要是因为这两种攻击都是利用系统安全漏洞，获得该目标主机的当地访问权限或管理员权限，而本系统是依靠分析网络数据包来发现入侵，但是总体来说这种基于隐马尔科夫模型和神经网络的入侵检测模型比单独使用隐马尔科夫模型或者神经网络的入侵检测的检测率还是要高。

5.6 本章小结

本章将 java 作为开发语言，实现了入侵检测模型的关键模块。最后，通过实验验证了该模型比单独使用隐马尔科夫模型或者神经网络的检测率都要高。

第六章　总结和工作展望

6.1 总结

计算机网络作为当代生产、生活必不可少的工具的同时，网络的安全现状也令人忧心忡忡。传统的网络安全技术已不能满足目前多变的网络安全防护的需要，入侵检测技术成为了网络安全这一领域研究的热点。入侵检测可以研究系统结构也可以研究检测算法。文中利用隐马尔科夫模型状态转移原理和神经网络良好的分类能力，提出了一个基于隐马科夫模型和神经网络的入侵检测模型并对其进行了实现，主要的工作如下：

(1)在了解国内外入侵检测技术的基础上，重点分析了 TCP/IP 协议导致的安全漏洞,详细介绍了 IP、TCP、UDP、ICMP 四种网络协议，分析了相关的网络攻击产生的原因。

(2)针对隐马尔科夫模型应用在入侵检测领域观察值难确定的问题,提出了一种确定隐马尔科夫模型观察值的方法。由于网络数据量过于庞大与瞬息万变，隐马尔可夫模型观察值千差万别、难以确定参数。文中根据 TCP/IP 管理模型、流量控制原理以及协议自身的特点，提出了一种确定隐马尔可夫模型观察值的方法，该方法大大的缩短了训练的时间。

(3)给出了入侵检测算法。入侵检测算法是入侵检测领域的核心，本文设计的入侵检测算法主要由隐马尔科夫模型和神经网络两部分组成。通过滑动窗口的原理，把隐马尔科夫模型的输出序列划分为相同长度的短序列，通过反复试验建立了 BP 神经网络结构，然后把短序列作为神经网络的输入，神经网络的输出即为最后的期望输出，这种设计的好处在于减少了数据库、有利于收敛。

(4)提出了基于隐马尔科夫和神经网络的入侵检测模型。目前利用隐马尔科夫模型理论或者神经网络理论去研究入侵检测的不少，并取得了不错的效果，但是大多数研究都是从系统调用的角度去考虑，而系统调用得缺点就是数目过大，难以训练。真正把二者结合起来从协议的角度联系起来的并不多，实验结果表明，把二者结合

起来进行入侵检测比单独使用隐马尔科夫模型或者神经网络都有较高的检测率。

6.2 工作展望

下一步的工作主要集中在以下四个方面:

1)本文只能处理 IP、TCP、UDP 和 ICMP 协议，它还可以加入其他协议的处理，特别是直接对应用层协议的处理。

2)本文只是给出了一种确定观察值的方法，并没有对文中提出的方法和其他的方法进行比较，下一步有待于在这面加强。

3）可以在这个方法的基础上,结合其他的一些入侵检测方法,来建立一个实际的入侵检测系统，比如隐马尔科夫模型和遗传算法结合起来检测入侵。

4）自动入侵响应的研究，因为入侵响应的决策过程很大程度上建立在对事件详细分类的基础，因此可以考虑多角度确定事件的属性，制定自适应的响应措施。

总之,基于隐马尔科夫模型和神经网络的入侵检测系统虽然具有一些创新意义，但是还只是入侵检测系统这一研究方向中一次小小的尝试，它还有待于进一步的改进和完善。

参考文献

[1] http://www.czedu.gov.cn/Disp.Aspx?serid=7917

[2] 孙宇. 网络入侵防御系统（IPS）架构设计及关键问题研究[D]. 天津：天津大学.2005

[3] 郑成兴. 网络入侵防范的理论与实践[M]. 北京：机械工业出版社，2006

[4] 刘文星. 网络攻击频率混沌时间预测[D]. 湖南：国防科技大学.2007

[5] 张良均，曹晶，蒋世忠. 神经网络实用教程[M].北京：机械工业出版社，2008

[6] 张松红，王亚弟，韩继红. 基于攻击意图的复合攻击预测方法研究[J]. 计算机工程与设计，2007,28（21）：21-28

[7] 曹晖，王青青，马义忠等. 基于动态贝叶斯博弈的攻击预测模型[J]. 计算机应用，2007，42(7):64-67

[8] 赵玉明. 基于隐马尔科夫模型的网络入侵检测系统研究[D]. 广东:广东工业大学,2005

[9] Y.Qiao， X.W.Xin， Y.Bin and S. “Anomaly intrusion detection method Based on HMM”，Electionics Letters，20th June 2002，vol.38，no.13，pp 663-664.

[10] 梅挺,代群,任伟. 基于误差反向传播的入侵检测系统的研究[J]. 通信技术,2008,41(05):125-127

[11] 殷莹. BP 网络在设备故障诊断中的应用与实现[J]. 微计算机信息，2007，23(08):139-141

[12] Anderson J P. Computer security threat monitoring and surveillance[R]. Technical Report，James P AndersonCo. For washington，USA，April，1980

[13] DorothyE Denning. An intrusiondetection mode[J]. IEEE Transaction on Software Engineering. 1987,SE-13(2):222-232

[14] L．T．Heberlein， G..v.Dias, K.N.Levitt, B.Mukherjee, J.woodWolber. A Network Security Monitor.Proc. of the IEEE Symposium on Research and Privay Oakland,CA,May 1990, pages 96-304

［15］ADCT. FY1998 Information Assuranee:Automated Intrusion Deteetion Enviroment(LA:AIDE)[EB/OL].http://www.acq.osd.mil/actd/descript.htm.1998

［16］LeeW， StolfoSJ， MokKW. A Data Mining Framework for Building Intrusion Detection models [DB/OL]. http://www.cse.msu.edu/wuming/Papers/lee99data.Pdf

［17］Ming-YuhH， Jasper R J， Wicks T M. A Large-scale Distributed Intrusion Detection Framework Based on attack Strategy Analysis〔J] . Computer Networks，1999(3):2465-2475

［18］ShyhtsunJ Y， FelixW， Fengrnin G， Ming-Yuh H. Intrusion Deteetion for an on going attack[EB/OL]. http://www.minlab.cs.depaul.edu/seminar/fall2002/idsonggoing.1999

[19] Andrew Rathmell， Richard Overill， LorenzoValeri. Information Warfare Attack Assement system （IWAAS）[EB/ol]. http://www.kcl.ac.uk/orgs/icsa/old/iwaaappr.PDF.1997

[20] RathrnellA， Dorsehoer J， knights M. Project:Threat Assessment and Early Warring Methodologies for Information Assurane[EB/OL]. http://www.icsa.uk/projects/ropa.html.1999

[21] Dong Yu，Deborah Frincke,A Novel Framework for Alert Correlation and Understanding[A]. Proceedings of Applied Cryptography and Network Security,Second Internation Conferce[C]:Lecture Notes in Computer Science,2004:452-466

[22] Dong Yu，Deborah Frincke. Improving the quality of alerts and predicting intruder’s next goal with Hidden Colored Petri-Net[J] Computer Networks,2006(51):632-654

［23］赵玉明，腾少华. 基于隐马尔可夫模型的网络入侵检测系统研究[D]. 广东:广东工业大学，2005

［24］张小强. 几类高效入侵检测技术研究[D]. 西南交通大学，2002

［25］张翔，胡昌振，刘胜航等. 基于支持向量机的网络攻击态势预测技术研究[J]. 计算机工程，2007，33(11):10-12

[26] 张松红. 一种基于隐马尔科夫模型的复合攻击预测方法[J].计算机工程,2008
[27] 柳亚明，许峰，吕志军，黄浩. 基于攻击意图的报警信息关联研究[J]. 计算机科学，2005，32(9):61-64
[28] 顾荣杰，晏蒲柳，邹涛. 基于统计方法的骨干网异常流量建模与预警方法研究[J]. 计算机科学，200633(2):92-95
[29] 祝洪杰. 基于神经网络的入侵检测研究[D]. 山东：山东大学，2008
[30] 吴霞. 基于粗糙集—神经网络的入侵检测系统的研究[D]. 湖北：武汉理工大学.2009
[31] 唐正军. 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
[32] S Chen， B.Tung， G:D.Schnackenber. “The Common Intrusion Detection Framework dataFomrats[R]”，internet draft itefcidff data formats00.txt 1998
[33] 蒋建春，冯登国，网络入侵检测原理与技术[M]. 国防工业出版社，2007.7
[34] S.Kumar. “Classification and Detection of Computer Intrusions”，Dissertation，Purdue University，2006
[35] 张耀疆. 聚焦黑客攻击手段与防护策略[M]. 人民邮电出版社,2002
[36] Andrew S.Tanenbaum. 计算机网络（第四版）. 北京：清华大学出版社
[37] Lasse Huovinen， Jani Hursti. Denial of Service Attacks：Teardrop and land. http://www.hut.fi/huovine/hacker/dos.html,March 1998
[38] CERT. CERT advisory CA-1996-21: TCP SYN Flooding and IP Spoofing Attacks.1996 http://www.cert.org/advisories/CA-1996-21.html
[39] CERT. CERT advisory CA-1996-01: UDP Port Denial service Attack.1996 http://www.cert.org/advisories/CA-1996-01.html
[40] CERT. CERT advisory CA-1998-01: Smurf IP Denial of services Attack.1996 http://www.cert.org/advisories/CA-1998-01.html
[41] CERT. Denial of Service Attack via Ping.1996 http://www.cert.org/advisories/CA-96.26.ping.html
[42] Mark Stamp. A Revealing Introduction to Hidden Markov Models. January 18，2004

［43］The open source network intrusion detection system. http://www.snort.org

［44］Rabiner L R. A tutorial on hidden Markov models and selected applications in speech recognition [J]. Proceedings of the IEEE. 1989,77(2):257-289

［45］Widrow B. Neural Network Application in Industry，Business and Science[J]. Communication of the ACM，2004，37:93-105

［46］Forrest S. S.A.Hofmeyr, A.Somayaji. 1998.Intrusion detection using sequencies of system calls,Journal of Computer Security Vol.6,pp.151-180.

［47］Dirk Ourston， Sara Matzner， Willian Stump， Bryan Hopkings. Application of Hidden Markov Models to Detecting Multi-stage Network Attacks[A]. Proceedings of the 36th Hawaii Intemational conference on System Sciences[C]. HICSS03，2003:1411-1523

［48］李金玲. 基于隐马尔科夫模型的无线局域网入侵检测系统研究[D]. 中国科学技术大学，2009

［49］韩景灵，孙敏. 一种入侵检测报警信息融合系统的构建与实现[J]. 计算机技术与发展. 2007，17(6):159-162

［50］D.Moore， G Voelker， S.Savage. Inferring intemet Denial of Service Activity[C]. Proceeding of the 10th usenix security Syrnposium. 2001:9-22

［51］Warrender C， Forrest S， Pearlmutter intrusion using system calls:alternative data model[A]. Proceedings of 1999 IEEE Symposium on Computer Security and Privacy[C]. Oakland,California:IEEE Computer Society Press,1999，133-145

［52］姜浩. 基于神经网络的入侵检测研究[D]. 湖北：华中科技大学, 2007

［53］田国钰，王海洋. 神经网络中隐含层的确定[D]. 信息科技，2010. 10

［54］李春艳. 利用神经网络技术实现星敏感器的星图识别[J]. 大连：辽宁师范大学图书馆，2003

［55］Srinivas Mukkamala, Andrew H.Sung. Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques, Iinternational Journal of Digital Evidence, Winter 2003,4:63-69.[J]

［56］R.Dugad and U.B.Desai. “A tutorial on hidden Markon models”，Technical

[57] 潘志松. 基于神经网络的入侵检测研究[D]. 南京：南京航空航天大学博士论文，2003

[58] 韩景灵. 基于协议的隐马尔科夫模型入侵检测系统研究[D]. 山西：山西大学，2007

成果目录

[1] 闫新娟，谭敏生，严亚周，吕明娥. 基于隐马尔科夫模型和神经网络的入侵检测研究. 计算机应用与软件,2011（已录用）

[2] 闫新娟，谭敏生，吕明娥. 基于行为分析的主动防御技术研究. 计算机安全,2010,（10）：38-39

[3] 严亚周，闫新娟. C/S 模式考试系统中 OFFICE 题目自动判卷技术的研究. 电脑知识与技术，2009，5（24）:7065-7067

致　　谢

值此学位论文完成之际，谨向那些为我授过课的老师、曾教导过我的师长，帮助过我的同学和朋友，支持我的亲人，表达我最诚挚的谢意！

首先，我要深深感谢我的导师谭敏生教授。导师渊博的知识，崇高的品德，严谨的治学态度和实事求是的科学作风，都给我留下了深刻的印象。感谢谭老师三年来在做学问的态度和思考问题的能力上对我的培养和指导；感谢谭老师在我学习中遇到困难时给我的鼓励，支持我从事网络安全这一具有时代性、挑战性的研究；感谢谭老师在我论文撰写过程中不厌其烦的指导，谭老师带领我走上学术之路，对我从事科研工作影响巨大。

另外，特别感谢计算机学院 2008 级 511 的全体同学，感谢你们在这三年来对我的帮助。在此，我表示深深的谢意。

还要感谢三年来，为我授课的所有老师。也正是你们的耐心讲授和过硬的专业知识，使我对计算机网络这一领域有了全新的认识！

感谢我的父母，始终尊重我的选择；感谢我的爱人，在我这三年求学生涯中给我支持和鼓励。

最后，对在百忙中抽出时间来评阅我的论文并提出了宝贵意见的各位专家和评委，表示深深的感谢！