**天 津 师 范 大 学**

**本科毕业论文**(**设计**)

题目：**基于OpenCV+TensorFlow的人脸识别设计与实现**

**学 院：计算机与信息工程学院**

**学生姓名：张雪宁**

**学 号：**1630090090

**专 业：计算机科学与技术**

**年 级：**2016**级**

**完成日期：**2020**年**4**月**

**指导教师：张少强**

基于OpenCV+TensorFlow的人脸识别设计与实现

摘要：近十年来，人工智能、大数据和云计算等技术发展迅猛。而计算机视觉作为人工智能下的一个分支，自上世纪80年代以来，一直备受关注。其中，包括人脸识别在内的物体检测也是计算机视觉领域内一个基础且重要的研究方向，同时衍生出人脸识别等实际应用。此外，大量研究表明，与静态人脸识别相比，动态人脸识别在实用性方面具有独到的技术优势。基于此，本文共进行了三个实验。第一，采用基于TensorFlow的MTCNN算法，利用三个级联的轻量级卷积神经网络完成人脸对齐和人脸检测，并对人脸检测的5个关键点进行标记。第二，在上个实验的基础上，通过深度卷积神经网络将图像映射到欧几里德空间，利用FaceNet算法实现人脸识别、人脸验证和人脸聚类。第三，在前两个实验的基础上，利用OpenCV库和外接摄像头，捕获动态视频并进行数字图像处理。最后，根据上述原理和实验，设计了一套简易的实时视频人脸识别系统。

关键词：人脸检测；人脸识别；深度卷积神经网络；TensorFlow；MTCNN；FaceNet

**Application of OpenCV and TensorFlow in Face Recognition**

**Abstract:** The past ten years have seen increasingly rapid advances in the field of artificial intelligence, big data and cloud computing. Computer vision, as a branch of artificial intelligence, has been received much attention since the 1980s. Among them, Object detection including face recognition is a basic and important area of interest within the field of computer vision. Extensive research has shown that dynamic face recognition has unique technical advantages in terms of practicality compared to static face recognition. Based on it, three experiments have been done in this paper: Firstly, we present an approach for face detection, which adopts a cascaded structure with three stages of carefully designed deep convolutional networks that predict face and landmark location in a coarse-to-fine manner. Secondly, based on TensorFlow and MTCNN in the last experiment, we present a system, called FaceNet, that directly learns a mapping from face images to a compact Euclidean space for face verification, recognition and clustering. Thirdly, we use the Open Source Computer Vision Library (OpenCV) and video cameras for dynamic video capture and digital image processing. Finally, according to the above principles and experiments, we design a simply video-based face recognition system.

**Key words:** Face Detection; Face Recognition; Deep Convolutional Neural Network; TensorFlow; MTCNN; FaceNet

目录

[1绪论 (1](#_Toc6672015))

[1.1 研究背景及意义 (1](#_Toc6672016))

[1.2 研究目标与内容 (2](#_Toc6672017))

[1.3 论文结构与贡献 (3](#_Toc6672018))

[2 DDoS攻击及其研究综述 (4](#_Toc6672019))

[2.1 DDoS攻击 (4](#_Toc6672020))

[2.2 DDoS攻击现状及其趋势 (4](#_Toc6672021))

[2.3 DDoS攻击检测方法 (5](#_Toc6672022))

[2.4 国内外研究情况 (6](#_Toc6672023))

[3 基于特征降维以及距离的DDoS攻击检测分类算法 (7](#_Toc6672024))

[3.1 PCA基于流量特征矩阵降维技术 (7](#_Toc6672025))

[3.1.1 主成分分析法 (7](#_Toc6672026))

[3.1.2 PCA数学模型构建方法 (8](#_Toc6672027))

[3.2 MCA特征三角形面积算法 (9](#_Toc6672028))

[3.3 数据特征挖掘及MD马氏距离检测 (9](#_Toc6672029))

[3.3.1 协方差矩阵的计算 (9](#_Toc6672030))

[3.3.2 马氏距离及阈值设定 (10](#_Toc6672031))

[3.4 实验结果与性能分析 (10](#_Toc6672032))

[3.4.1 实验数据预处理 (10](#_Toc6672033))

[3.4.2 实验结果评价指标 (11](#_Toc6672034))

[3.4.3 实验结果与分析 (12](#_Toc6672035))

[4 基于BP\_Adaboost强分类器的DDoS攻击检测算法 (15](#_Toc6672036))

[4.1 弱分类器——BP神经网络 (15](#_Toc6672037))

[4.1.1 BP神经网络结构及正向传播 (15](#_Toc6672038))

[4.1.2 误差反向传播 (17](#_Toc6672039))

[4.2 强分类器——BP\_Adaboost分类器 (18](#_Toc6672040))

[4.2.1 Adaboost算法分类问题 (18](#_Toc6672041))

[4.3 数据集介绍： (19](#_Toc6672042))

[4.3.1 数据集处理 (19](#_Toc6672043))

[4.4 实验结果与分析 (19](#_Toc6672044))

[5 基于神经网络Kohonen的DDoS攻击聚类算法 (21](#_Toc6672045))

[5.1 Kohonen神经网络 (21](#_Toc6672046))

[5.1.1 Kohonen神经网络简介 (21](#_Toc6672047))

[5.1.2 自组织过程 (22](#_Toc6672048))

[5.1.3 Kohonen算法 (23](#_Toc6672049))

[5.2 SVD去噪处理 (24](#_Toc6672050))

[5.2.1 SVD算法原理 (24](#_Toc6672051))

[5.2.2 SVD去噪原理应用Kohonen： (25](#_Toc6672052))

[5.3 实验结果与分析 (25](#_Toc6672053))

[5.4 Kohonen神经网络拓展——SKohonen(无监督→有监督) (28](#_Toc6672054))

[5.5 实验结果对比 (29](#_Toc6672055))

[6 简易检测系统的设计与实现 (31](#_Toc6672056))

[6.1 系统设计 (31](#_Toc6672057))

[6.2 系统功能实现 (31](#_Toc6672058))

[7 总结与展望 (34](#_Toc6672059))

[7.1 论文实验工作量 (34](#_Toc6672060))

[7.2 实验中出现的问题以及展望 (34](#_Toc6672061))

[参考文献 (36](#_Toc6672062))

[致谢 (37](#_Toc6672063))

[附录：代码 (39](#_Toc6672064))

1绪论

1.1 研究背景及意义

在过去的十几年里，人脸识别已经成为计算机视觉领域的热门研究方向，同时也是图像识别与处理中最成功的应用之一。人脸识别是基于人的脸部特征信息进行身份识别的一种生物识别技术，用摄像机采集含有人脸的图像或视频流，并自动在图像中检测和跟踪人脸，进而对检测到的人脸进行面部识别。

人脸识别通常包括人脸检测、人脸对齐、人脸表征和人脸匹配。

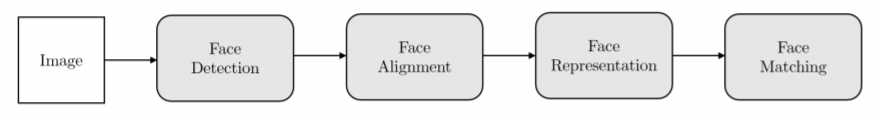


图1.1 人脸识别构建模块[7]

人脸识别系统的研究开始于20世纪60年代，在80年代后随着计算机技术和光学成像技术的发展得到提高，而真正进入初级的应用阶段则在90年代后期。[1]人脸识别的关键在于尖端的核心算法，并使识别结果具有较高实用性的识别率和识别速度。

目前，人脸识别主要应用于金融、社交和安防等行业。尽管随着深度学习在人脸识别中的应用，算法的准确率大幅提升，但现阶段的人脸识别技术还无法在所有现实复杂场景中尽善尽美，市面上大部分人脸识别应用对活体检测的效果不佳。在未来，3D动态人脸识别和活体检测等计算机视觉技术的应用将成为趋势。

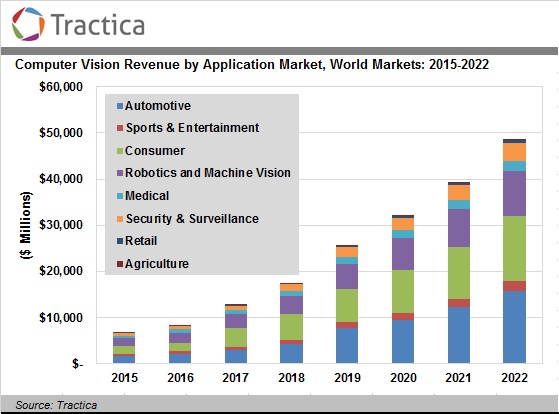


图1.2 2015-2022年计算机视觉应用市场收入预测图[2]

[Tractica](https://tractica.omdia.com/)在最新报告中指出，全球计算机视觉应用相关市场收入将从2015年的66亿美元增长到2022年的每年486亿美元，复合年增长率（CAGR）为32.9％。 [2]

1.2 人脸识别研究现状

人脸识别技术的发展可分为如下三个阶段：

第一阶段：半机械式识别阶段。这一时期的代表性论文为Parke等人发表的《Computer generated animation of faces》，在论文中，研究者实现了人脸灰度图模型，而他们也被认为是这一阶段人脸识别技术的代表性人物。这一时期的人脸识别过程主要以大量人工操作为主，识别过程几乎全部需要操作人员来完成，因此，这样的系统是无法自行完成人脸识别过程的。

第二阶段：人机交互式识别阶段。人脸识别技术在这一阶段得到了进一步的发展，研究者可以使用算法来完成对人脸的高级表示，或者可以以一些简单的表示方法来代表人脸图片的高级特征。例如AJ Goldstein、LD Harmon与AB Lesk在论文《Man-machine interaction in human-face identification》中使用几何特征参数表示人脸的正面图像；Kaya等人在论文《A basic study on human face recognition》中使用统计学方法，以欧氏距离作为人脸特征；Kanade则实现了一个半自动回溯识别系统。

但是，这部分人脸识别方法仍然需要研究人员的高度参与，例如在人脸识别过程中需要引入操作人员的先验知识，识别过程并没有完全摆脱人工的干预。

第三阶段：自动人脸识别阶段。只有将识别过程自动化才可以真正达到人脸识别的效果。而这项技术的发展，离不开机器学习的发展。早期的自动人脸识别与我们印象中的机器学习并不太相似，一般以几何特征和相关匹配的方法居多，在模型的设计上，常常会引入一些先验知识。除此之外，还有基于统计与基于子空间的识别方法。例如著名的特征脸（Eigenfaces）法就属于一种基于子空间的人脸识别方法。

在这一阶段中，人工神经网络（Artificial Neural Network，ANN）也得到了广泛的应用，由此演化出很多人脸识别中的新方法。例如当前非常热门的深度学习方法就属于人工神经网络的范畴。[3]

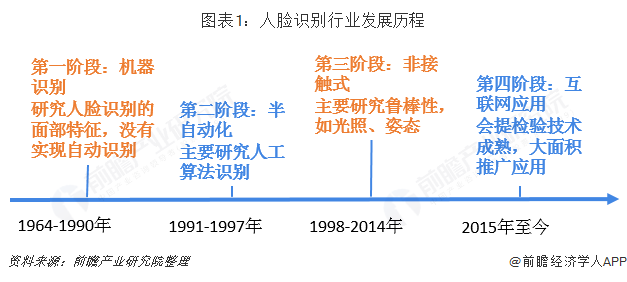


图1.2 人脸识别行业发展历程图[6]

现如今，人脸识别技术正成为IT产业下一轮技术浪潮，国内外诸多知名企业都在积极布局该领域。基于人工设计的特征和传统机器学习技术的传统方法近来已被使用非常大型的数据集训练的深度神经网络取代。在2019年7月美国国家标准与技术研究院(NIST)发布的全球人脸识别算法测试(FRVT)结果的最新排名中，最具挑战的“非约束性环境照片”人脸识别算法测试子项目冠军由格灵深瞳摘得，“签证照片”人脸识别算法测试子项目由依图科技获得，“嫌疑人照片”人脸识别算法测试子项目冠军则落入俄罗斯厂商VisionLabs囊中。而在2020年1月最新排名中，来自中国的格灵深瞳算法团队已摘得头筹，中国人脸识别技术发展水平位于世界前列。

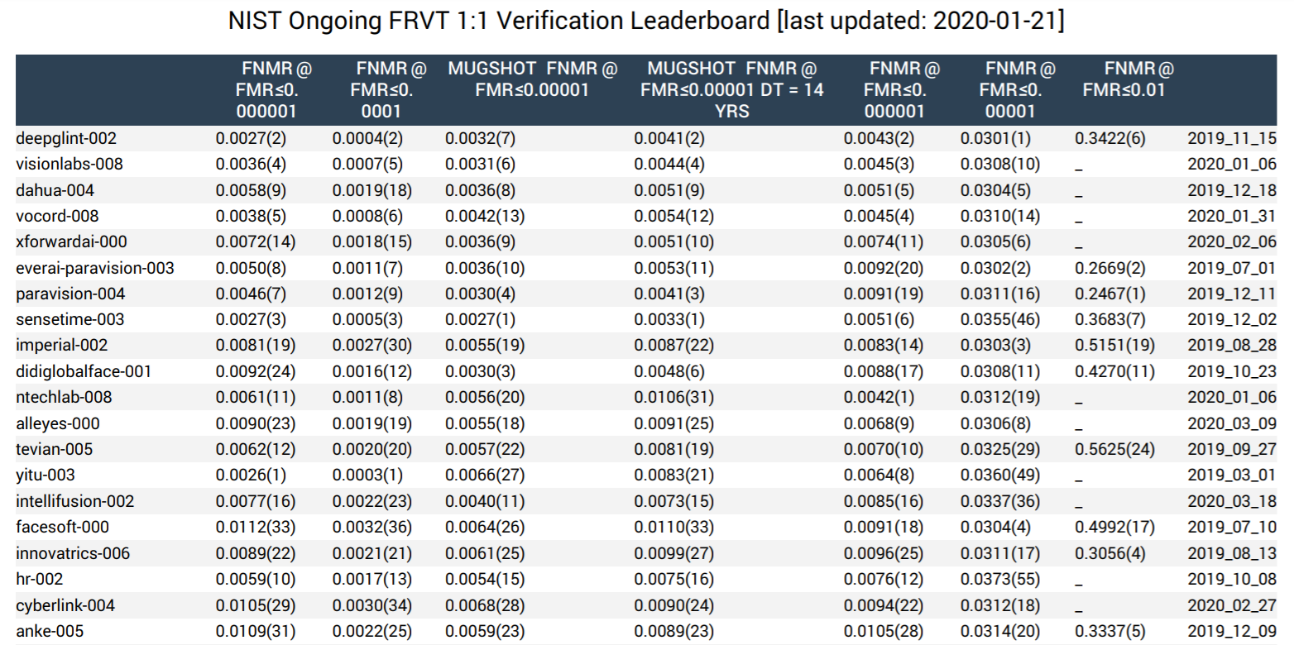


图1.3 2020年1月全球人脸识别技术FRVT中1:1识别排名情况[4]

1.3 人脸识别应用趋势

人脸识别的大致流程可以描述为：通过人脸识别模型判断图片中是否存在人脸，如果存在人脸，则定位到该人脸的区位，或者提取该人脸图像的高级特征作为该人脸图像的特征向量，并用在后续对图片的处理中。由于人脸识别的应用场景不同，上述步骤的选择和侧重点也不尽相同。[3]

人脸识别应用按照识别方式可分为三种模式：1:1、1:N和M:N模式。

|  |  |  |  |
| --- | --- | --- | --- |
| 模式 | 中文名称 | 英文别称 | 应用举例 |
| 1:1 | 人脸验证对比 | Face Verification | 人脸手机解锁、人证合一检验 |
| 1:N | 静态人脸对比 | Face Identification | 企业门禁系统 |
| M:N | 动态人脸对比 | 公共安防、天网系统 |

表1.3 人脸识别应用分类表

1:1人脸识别模式是一种静态对比，利用图像处理技术从图像中提取人脸特征值，将当前人脸与人像数据库进行快速人脸比对，并得出是否匹配的结果。其主要用于身份验证，应用场景如人脸手机解锁、人证合一等。

1:N人脸识别技术是在海量的人像数据库中找出当前用户的人脸数据并进行匹配。N的数目在千万级。其特点是动态和非配合，所谓的动态也就是识别的是一个动态的视频流，非配合是指识别对象不用感知到摄像头的位置并配合完成识别工作，而复杂的环境条件都会影响识别的准确性。其应用场景包括学校电子班牌、物业小区、新零售的客户识别等。

N:N是1:N的延伸，即同时对多张人脸进行人脸检索，需要占用更多的计算资源。它属于动态对比，通过计算机对场景内所有人进行面部识别并与人像数据库进行比较。其应用场景主要为公共安防、天网系统，例如公共场所动态监控、缉拿逃犯、人员布控等。[5]

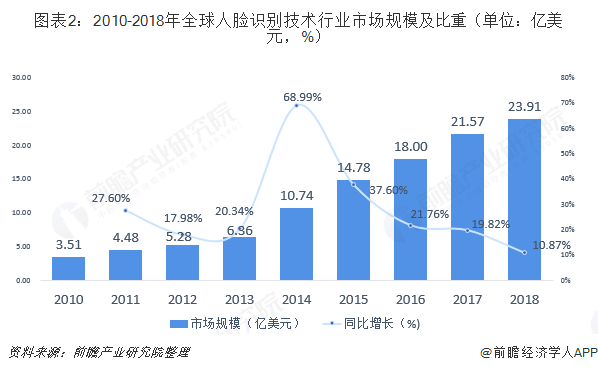


图1.4 2010-2018年全球人脸识别技术行业市场规模及比重统计图[6]

目前，国内的人脸识别技术已经相对发展成熟，该技术越来越多的被推广到安防领域，延伸出考勤机、门禁机等多种产品，产品系列达20多种类型，可以全面覆盖煤矿、楼宇、银行、军队、社会福利保障、电子商务及安全防务等领域，人脸识别的全面应用时代已经到来。[6]

1.4 本文主要工作及结构

本文结构主要包括六个章节：

1. 本章作为绪论，主要介绍了实验研究的课题背景和论文总体思路。
2. 第二章介绍了人脸检测算法的发展和MTCNN算法的实现细节。
3. 第三章介绍了人脸识别算法的理论基础，并在第二章的基础上，介绍了基于FaceNet进行人脸识别的实现细节。
4. 第四章在前三章的基础上，设计了基于TensorFlow的简易人脸识别系统。
5. 第五章作为总结与展望，阐述了针对上述实验的思考和自身的不足。
6. 第六章包含参考文献、致谢和附录代码。

本文在探究了经典的人脸检测和人脸识别算法，并根据结合深度卷积神经网络设计了一个基于TensorFlow+MTCNN+FaceNet的简易实时人脸识别系统。

2 人脸检测

2.1 人脸检测算法简介

人脸检测（Face Detection）是指对于任意一幅给定的图像，采用一定的策略对其进行搜索以确定其中是否含有人脸并返回关键位置信息。人脸检测的目标是找出图像中所有的人脸，确定它们的大小和位置，算法的输出是人脸外接矩形框的坐标和大小，还可能包括姿态（如倾斜角度等信息）。

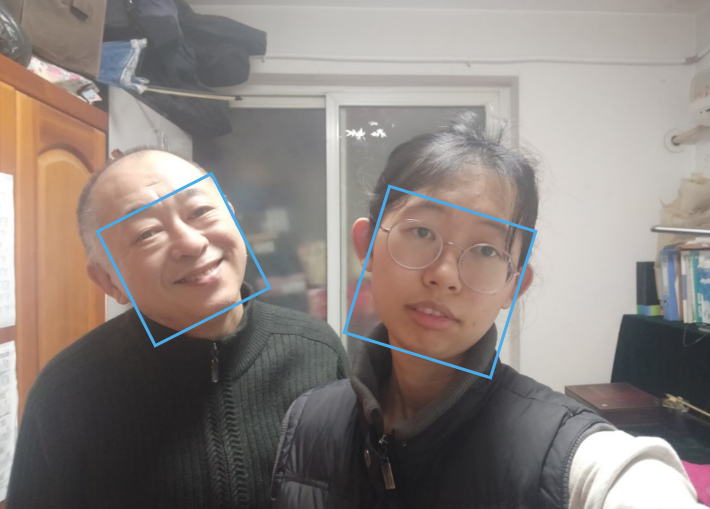


图2.1 人脸检测结果示意图

人脸检测是人脸图像分析的首要步骤，常应用于人脸识别、算法的第一步，主要解决以下几个核心问题：第一，人脸可能出现在图像中的任何一个位置；第二，人脸可能有不同的大小；第三，人脸在图像中可能有不同的视角和姿态；第四，人脸的一部分可能被遮挡（例如口罩）。

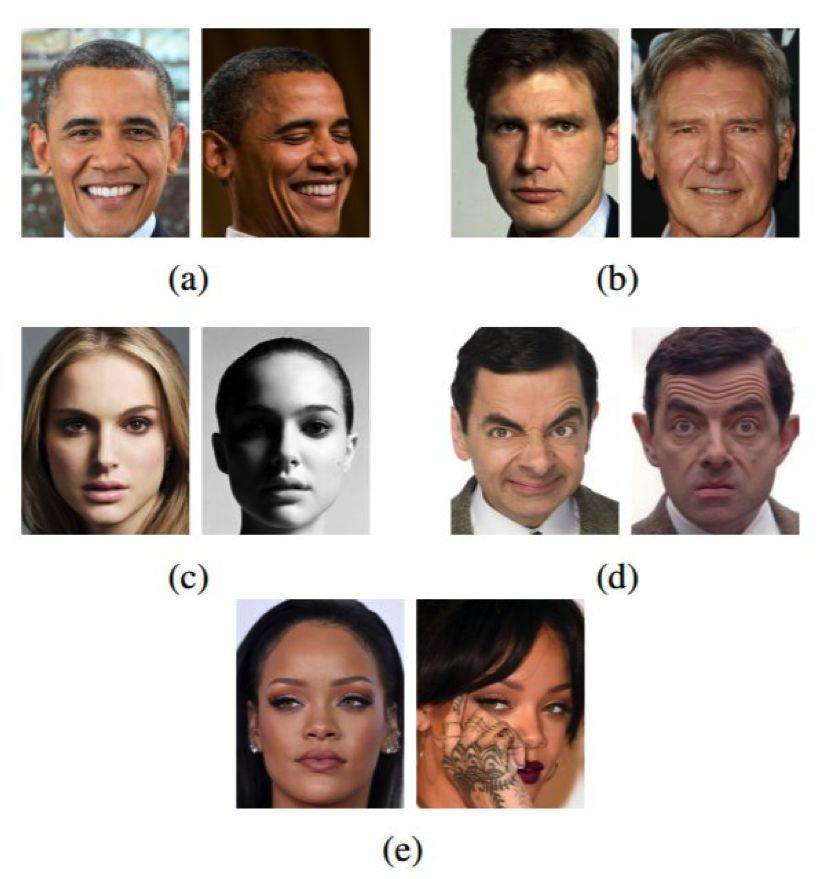


图2.2在自然人脸图像中找到的典型变化情况图[7]

1. 头部姿势（b）年龄（c）光照（d）面部表情（e）遮挡

主流的[人脸识别](http://www.anfangnet.com/renlianshibie/)算法基本上可以归结为三类，即：基于几何特征的方法、基于模板的方法和基于模型的方法。基于几何特征的方法是最早也是最传统的方法，通常需要与其他算法结合才能取得较好的效果；基于模板的方法可分为基于相关匹配的方法、特征脸方法、线性判别分析方法、奇异值分解方法、神经网络方法和动态连接匹配方法等。基于模型的方法则包括基于隐马尔柯夫模型，主动形状模型和主动外观模型的方法等。[9]

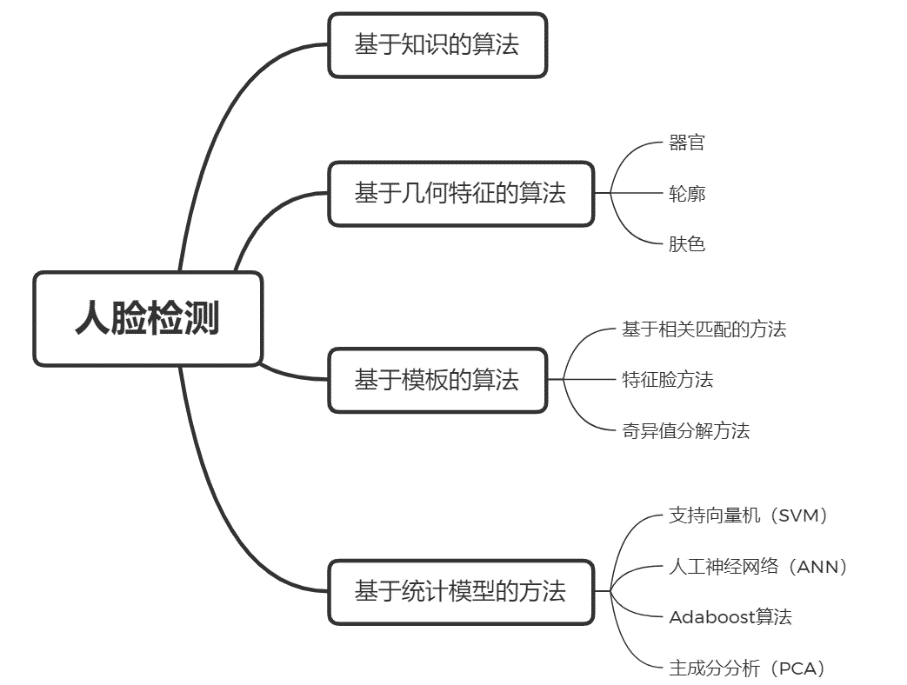


图2.3 经典人脸检测算法分类

在多年的技术发展中涌现了许多经典算法，例如由Freund等人于1995年提出的AdaBoost算法，它是一种有效而实用的Boosting算法，通过调整样本权重和弱分类器权值，从训练出的弱分类器中筛选出权值系数最小的弱分类器再组合成一个最终强分类器。[10]2001年，Viola和Jones提出了VJ框架，使用简单的Haar特征和级联AdaBoost分类器构造检测器，将检测速度提高了2个数量级并且有很高的精度。[11]VJ框架是人脸检测历史上有里程碑意义的一个成果，奠定了AdaBoost目标检测框架的基础。在深度卷积神经网络应用之前，AdaBoost算法一直处于计算机视觉检测的主导地位。

随着技术的发展，人们尝试将卷积神经网络应用于人脸检测这一问题。一般分为两种方式：一种是将适用于多种任务的目标检测网络应用于人脸检测任务，例如Faster-RCNN网络、YOLO网络和SSD网络等；另一种则是研究专门的人脸检测网络，例如cascadeCNN[12]，MTCNN等。

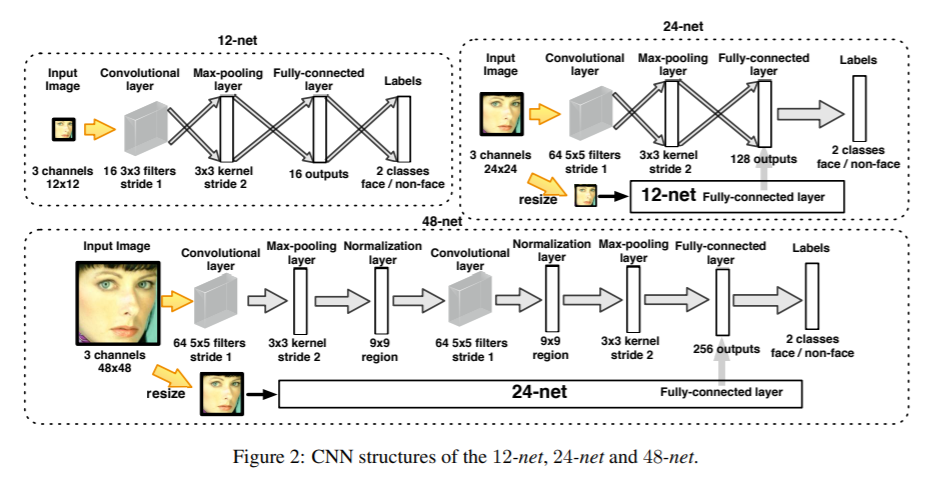


图2.4 用于矫正人脸框的cascadeCNN网络结构图 [12]

到目前为止，《ASFD: Automatic and Scalable Face Detector》提及的算法在世界权威的人脸检测公开评测集WIDER FACE的Easy、Medium和Hard三个评测子集的六项评估结果中, 综合排名第一。

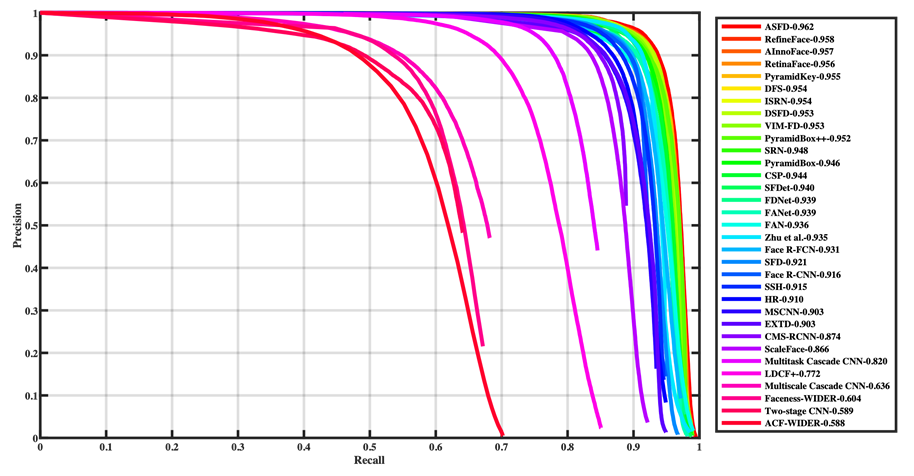


图2.5 各个算法在WIDER FAC中Medium测试组下的性能曲线[8]

本章节将简单介绍几个经典的人脸检测算法及其原理。

2.2 AdaBoost算法

AdaBoost算法，即自适应提升（Adaptive Boosting），是一种用于分类问题的算法，它用弱分类器的线性组合来构造一个准确的强分类器。

2.2.1 理论基础

⑴强分类器与弱分类器

强分类器的计算公式为

其中，x为输入向量；F(x)为强分类器；ft(x)为弱分类器；at为弱分类器的权重；T为弱分类器的数量，弱分类器的输出值为+1或-1，分别对应正样本或负样本。分类时的判定规则为

强分类器的输出值也为+1或-1，同样对应于正样本或负样本。弱分类器和它们的权重通过训练算法得到。对于二分类问题，只需要保证弱分类器的准确率大于随机猜测（50%）即可。

⑵训练算法

训练样本带有权重值，初始时所有样本的权重相等。训练时，依次训练每一个弱分类器，并得到它们的权重。在训练过程中，被前面的弱分类器错分的样本会加大权重，反之会减小权重，从而使后面的弱分类器会更加关注这些难分的样本。弱分类器的权重值根据它的准确率构造，精度越高的弱分类器权重越大。给定m个训练样本(xi,yi)，其中,xi是特征向量，yi为类别标签，其值为+1或-1。训练算法的流程如下。[13]

①初始化样本权重值，所有样本的初始权重相等：

②循环，对t=1,2,…,T依次训练每个弱分类器：

训练一个弱分类器ft(x)，并计算它对训练样本集的错误率et

计算弱分类器的权重：

更新所有样本的权重：

其中，Zt为归一化因子，它是所有样本的权重之和：

结束循环

最后得到强分类器：

是由Freund等提出的一种Boosting算法的实现版本。它的

的全程目前，机器学习算法广泛用来训练数据，而其中监督学习，无监督学习作为其中重要的两种学习思维模式常常用在异常检测中，监督学习是指在已知数据类型标签不断训练数据，使得整体模型误差值达到最小成为最优模型，根据最优模型再去处理测试数据，通俗理解则为根据已有推测未知，与之相反，无监督学习缺少了训练过程，事先并不知道所给数据具体属于哪一分类，完全凭靠机器进行自主学习探索对于数据进行聚类。对于DDoS流量数据分析，存在着分类与聚类的差别，其中前者是对于监督学习而言根据已有将未知进行分类，而后者则大多数通过计算数据之间的相似度等将相似的数据归为一类。

2.4 国内外研究情况

对于有监督学习来说，2017年，北京邮电大学贾斌博士在基于机器学习与统计分析分析[1]中，利用有监督学习设计了多元降维分析并实时检测的算法，解决了互联网中大流量攻击实时检测问题。在他的研究中，继续使用有监督算法，将若干分类器组合起来，将检测正确率都有所提高。并且实验数据为KDDCup99数据集，利用其算法将正常与异常流量区分开来。在国内也有大量学者使用贝叶斯分类器等。利用遗传算法改进卷积神经网络等监督学习方式改进分类器，实现对网络的入侵检测包括后续检测之后进行防御，这些传统的机器学习方式都起到了重大作用。

对于无监督学习方法来说，2016年北京理工大学张圆园硕士[3]改进了Kohonen网络学习权值，并且引入了隶属度，在保持运行速度的情况下提高检测率。2017年河北大学董宁硕士[4]针对Kohonen网络，添加输出层。将原本无监督网络变成监督学习，由于竞争层不能自主学习输入向量，董宁硕士对于竞争层实现优化，使得在分类检测中更加完善其功能。2017年中国矿业大学于冰洁硕士[5]将高斯混合模型引入至异常检测算法中，利用差分贝叶斯改进传统的EM算法，并且基于成分分裂的增量学习方法，实现自主聚类，不需要提前预知K值。在实验中，同样使用KDDCup99数据集做实验并且证明算法的有效性。

在国外学术报刊，国际会议上，2016年，Saied等人[13]运用人工神经网络实现了实时检测未知DDoS攻击行为，2017年，Abdulaziz等人[14]采用协方差技术，采用多元相关性方法对于数据集进行了训练，并在云环境在实现了对DDoS的有效检测。2018年，DA YIN等人[15]发表论文，针对当前物联网设备，利用软件定义物联网提出缓解DDoS攻击的有效算法。

3 基于特征降维以及距离的DDoS攻击检测分类算法

本章节针对网络中DDoS攻击简单而分类实现正异常分类提出一种PMM算法，随着大数据以及相关技术发展，网络流量呈现出高度的复杂性及突发激增，对于高维度流量信息，准确把握其特征之间的关系，将攻击流量与正常流量分离，提高对于正异常流量的辨别能力尤为重要。

本章提出的PMM(PCA&MCA&MD)算法，结合了主成分分析PCA(principal component analysis)基于流量特征矩阵的降维技术，多元相关分析MCA(Multivariate correlation analysis)基于三角形面积挖掘网络流量记录中两两特征值关系技术，马氏距离MD ( Mahalanobis distance)基于协方差计算两条流量之间相似度方法。首先对于搜集到的一条高纬度流量信息，本实验采用降维技术，在确保降维后向量仍可以到达80%以上代替原有数据，实现降维操作。其次，对于降维之后的低维数据，两两特征之间计算其三角形面积，得到新矩阵用于流量间相互关联统计特性进行分析。最后，对于处理后的矩阵进行马氏距离的计算，并且得出相应阈值，根据距离范围将正常流量与异常流量区分开来。

与前人工作相比，前人将检测模型数据分类准确度由78%平均提升至88.5%，本实验进一步准确度提升至92.12%左右，本章实验采用了著名的知识发现和数据挖掘(Knowledge Discovery and Data mining, KDD) Cup 1999数据集，实验结果表明，对于正常异常流量阈值的限定的处理上，本算法采取上下阈值取不同的值，更加符合数据信息特征从而进一步提高系统整体分类准确性。

3.1 PCA基于流量特征矩阵降维技术

DDoS攻击与正常流量相比，在短时间内产生大量流量，因此攻击流量能用过其特征属性与正常流量区分开来，对于高维度数据，有些特征所携带信息量差别不大，可能对实验会产生干扰，也会产生不必要的时间代价。因此，本节主要探讨基于主成分分析PCA的降维分析方法。

3.1.1 主成分分析法

主成分分析(principal component analysis)在面对数据维度高等问题中经常运用的降维手段，也称作为主分量分析，在矩阵特征值与特征向量基础之上实现降维，将代表一条数据的且所含信息互相不重复。使用这种方法可以使得引进多方面变量但最终实现将复杂数据归为某几个重要因素，也就是本实验说的主成分，使得大数据简单化。在线性代数的学习中，对于方阵来说，对角化之后得到求得特征值与特征向量，每一个特征值即可以代表原数据中不同维度数据所代表原数据占比，如果原矩阵不为方阵，即矩阵行数不等于列数时，采用SVD方法也可以实现求解特征值。

利用上述变化思想，PCA是将多个特征指标转化为少数具有代表性的典型特征，将旧坐标转化为新坐标，在求解新坐标的时候，首先确定第一个坐标轴方向，选择与原始数据方差最大作为初始方向，然后数据的维度作为重复求解后续坐标方向的次数，依此求解与上一个坐标正交并且方差最大的新坐标方向，根据累积的贡献率，最后决定选取几维数据最终代表整体数据。

3.1.2 PCA数学模型构建方法

假设一个包含有*n*条流量记录的矩阵*S*，每条流量记录的维度为*d*，即矩阵*S*可以表示为：

首先，根据样本矩阵*S*，所根据线性代数原理，得到其相关系数矩阵*C*，其中，相关系数矩阵又称为相关矩阵，相关系数矩阵*C*由*S*中各列间的相关系数构成(例如，*C*矩阵中的标号为(*i*，*j*)的元表示*S*矩阵中第*i*列和第j列之间的相关系数)，因此，相关系数矩阵为沿对角线对称的矩阵并且其对角线元素均为1。其次对相关系数矩阵做对角化处理，根据对角化定义，矩阵*C*为*d\*d*矩阵，其特征值为特征向量为,形成线性无关集合，以每个特征向量为列构成矩阵*A*，则其关系如公式(3.1)所示:

(公式3.1)

然后求出前*p*个最大特征值所对应的特征向量，并对它们进行正交化处理，新的特征向量组成新的矩阵，此时，就可以得到*p*个主成分。由此可见PCA降维处理，就是将原坐标系旋转产生新坐标，而新生成的坐标轴具有最大的样本方差，根据贡献率大小挑选出*p*个主成分代表原有数据。其中第一成分贡献率为如公式(3.2)所示：

(公式3.2)

其中数值比值越大，则说明此成分能够代表原数据的能力就越强。

3.2 MCA特征三角形面积算法

Z. Tan等学者[16]提出了一种基于三角形面积的MCA算法，该算法通过将一条数据流量中每两个维度数据作为三角形两边，来计算三角形面积。最终形成的MCA矩阵中，再根据下一节所要介绍的马氏距离，来挖掘两个特征值之间关系，获取相关性信息。

假设矩阵*S*1为原始数据矩阵*S*经主成分提取降维处理之后的数据矩阵，则，矩阵表达式则为公式3.3所示：

(公式3.3)

抽取其中第*i*条流量，则，其中表示第*i*条流量将用于计算三角形面积处理，*T*为标识，表示第*i*条流量的第*n*个属性特征，根据上文提到的三角形面积计算，对于每一条流量中每两个特征之间进行计算，例如，分别表示第*i*条流量的第*j*、*k*个属性，那么其三角形面积可如公式3.4表示为：

(公式3.4)

对于每一条流量来说，其三角形面积矩阵(Triangle Area Matrix，TAM)便可得到，根据两条向量求三角形公式可得，若*j=k*即连个属性相同，此时两条向量方向相同或者相反则，面积为0，因此TAM矩阵中主对角线元素都为0，、不难理解此时矩阵沿对角线对称，由于为对称矩阵，在实际计算中，本实验将每一条流量两两特征看成直角三角形的两边直接用于求面积。为了将多条流量数据求得的多个特征三角形面积归结到一个矩阵中，本实验只取其下三角矩阵即可并且归为一列，即，因此得到最终*TAM*矩阵如公式3.5：

(公式3.5)

3.3 数据特征挖掘及MD马氏距离检测

3.3.1 协方差矩阵的计算

为了最终结果能将正常流量与异常流量区分开来，可以首先将*n*条正常流量数据即数据集中标签为normal的数据构造为原始矩阵*S*，讲过上述一系列计算之后，得到正常流量的三角形面积矩阵如公式3.6：

(公式3.6)

对于其中任意两个元素可以求其协方差最终可得矩阵，如公式3.7所示：

(公式3.7)

3.3.2 马氏距离及阈值设定

利用1.3.1所得到的协方差矩阵，进行其马氏距离的计算，其公式如3.8所示：

(公式3.8)

其中对于原始n条正常训练数据来说其期望值为公式3.9所示：

(公式3.9)

在算得n条流量的马氏距离基础之上，通过设定上下阈值完成正常流量的距离范围计算，这样对于一条新流量信息，如果其距离落入正常范围之内则可认为其正常，否则为异常流量。其中本实验将限阈值的范围设定为：,下限阈值的设定为：其中对于其分别如公式3.10与公式3.11所示：

(公式3.10)

(公式3.11)

表示所有正常流量所求距离的平均值，而公式中则代表标准差。在实验过程中，根据实际情况，在取上阈值时，实验将限定在1到2之间(以0.05作为增量区间)，在取下阈值时，本实验将限定在0.4到0.9之间(以0.025作为增量区间)。到此为止，本实验已经根据PMM算法建立好检测模型，根据已有得到的正常流量的数据，如果采集到一条未知流量时，则可以计算其MD，如果距离值落入预期范围之内则认为正常，否则为异常流量。

3.4 实验结果与性能分析

3.4.1 实验数据预处理

实验语言：matlab(R2018b)。

实验服务器：Windows10家庭版，GPU版——4G显卡,型号GeForce GTX 960M;CPU版——Intel(R) Core(TM) i5-4210H CPU @ 2.90GHz，内存：8G。

实验数据集：本实验选取“KDD CUP 99 dataset”数据集，指KDD竞赛在1999年举办时所用数据，1998年林肯实验室模拟美国空军局域网网络环境共搜集九周而得。完整的数据集中一共包含有500多万条数据，由于原数据量大，此外还提供了10%流量记录的训练集与测试集用于训练与测试。在数据集中，每一条流量记录共有42项标记，其中前41项表示数据的属性最后第42项数据表示数据所对应的类别。例如对于正常流量标识则为Normal，对于异常类型，共分成了四大类，分别为拒绝服务攻击DOS，来自远程机器的非法访问R2L，普通用户对本地超级用户特权的非法访问U2R，监视和其他探测活动Probing，对于本实验研究对象为DDoS攻击，因此选择标签为normal与DOS攻击类型的流量。其中本文研究的攻击类型共涉及六种(back, land, neptune, pod, smurf, teardrop)搜集到数据之后，首先要对数据进行预处理，对于数据的处理步骤如下：

1. 在每一条数据记录中共有41项数据特征，有三种是非数值型，首先本实验需要将其转化为数值类型，详细的数据转化类型详见表3.1：

表3.1 数据转化类型表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 顺序 | 特征名称 | 类型转化1 | 类型转化2 | 类型转化3 | 类型转化4 |
| 2 | protocol\_type | TCP=1 | UDP=2 | ICMP=3 | 无 |
| 3 | service | ecr\_i=1 | private=2 | http=3 | 其他=0 |
| 4 | flag | SF=1 | 其他=0 | 无 | 无 |

1. 本实验对于正常流量的选取是通过筛选10%训练数据集中标签为normal的数据，共有97278条，在41个特征中，有三项特征均为0，因此在处理时，对于wrong\_fragment，num\_outbound\_cmds，is\_hot\_login这几项特征忽略不做考虑。
2. 在使用PMM算法中，进行第一步降维处理时，为了可以获得更多有效成分信息，将累计贡献率设置为72.05%。

3.4.2 实验结果评价指标

本实验的目的是为了将流量信息根据所得马氏距离范围准确进行二分类，如何判断分类是否准确，则需要一系列的评价指标，首先，根据分类情况，需要了解如下四个信息：TP(False Negative)对于本实验来说指的是将攻击流量正确地预测为攻击的个数，FP(False Positive)是指将正常流量错误地预测为攻击的个数，TN(True Negative)则代表的是将正常流量正确地预测为正常的个数)，FN(False Negative)代表将攻击流量错误地预测为正常的个数，在得到以上四个信息之后，由此可以计算如下典型指标：

DR(Detection Rate)：攻击样本中正确预测为攻击所占比例。此指标可以直接反应PMM算法对于攻击流量的识别能力。

TNR(True Negative Rate)：正常样本中仍被认为为正常流量的比例。此指标反映出模型算法对于正常流量识别能力的准确性。

Accuracy正确率：指所有测试样本集中正确分类的样本所占比例，他在整体上反映了检测模型对于流量正常与否区分性能的好坏。

Precision精准率：在所有被认为是攻击样本中真正的攻击样本所占有比例。它反映出，检测模型将来运用在实际当中，对于攻击流量做出辨别的准确度。

FPR(False Positive Rate)：被错误的认为是攻击流量的正常流量占全部正常流量的百分比。与TNR一起，可以用于反应此检测模型对于正常流量识别能力的强弱。

上述指标其计算方法分别如下：

DR=TP/(TP+FN) (公式3.12)

TNR=TN/(FP+TN) (公式3.13)

Accuracy=(TN+TP)/(TP+FP+TN+FN) (公式3.14)

Precision=TP/(TP+FP) (公式3.15)

FPR=FP/(FP+TN) (公式3.16)

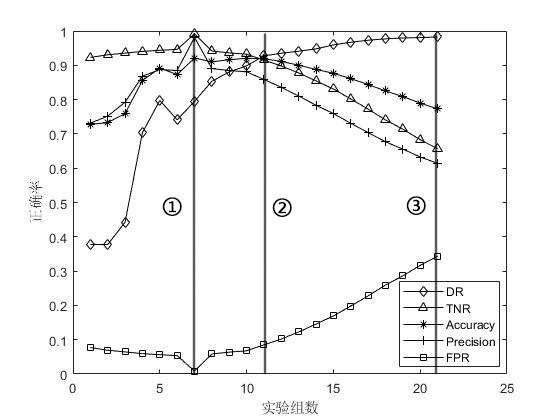
3.4.3 实验结果与分析

在实验过程中，根据实际情况，在取上阈值时，本实验将限定在1到2之间(以0.05作为增量区间)，在取下阈值时，本实验将限定在0.4到0.9之间(以0.025作为增量区间)。本实验在10%的数据集中，随机挑选出正常流量8166条用于训练，并挑选出4542条异常流量辅助测试，最终实验结果如表3.2所示：

表3.2 实验结果统计表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 下阈值(10-5) | 上阈值(10-5) | TP | FP | TN | FN |
| 0.9 | 1 | 0.02943 | 3.7008 | 1714 | 629 | 7537 | 2828 |
| 0.875 | 1.05 | 0.0777375 | 3.797415 | 1714 | 566 | 7600 | 2828 |
| 0.85 | 1.1 | 0.126045 | 3.89403 | 2010 | 526 | 7640 | 2532 |
| 0.825 | 1.15 | 0.1743525 | 3.990645 | 3196 | 485 | 7681 | 1346 |
| 0.8 | 1.2 | 0.22266 | 4.08726 | 3623 | 460 | 7706 | 919 |
| 0.775 | 1.25 | 0.2709675 | 4.183875 | 3366 | 437 | 7729 | 1176 |
| 0.75 | 1.3 | 0.319275 | 4.28049 | 3610 | 69 | 8097 | 932 |
| 0.725 | 1.35 | 0.3675825 | 4.377105 | 3876 | 477 | 7689 | 666 |
| 0.7 | 1.4 | 0.41589 | 4.47372 | 4001 | 521 | 7645 | 541 |
| 0.675 | 1.45 | 0.4641975 | 4.570335 | 4081 | 552 | 7614 | 461 |
| 0.65 | 1.5 | 0.512505 | 4.66695 | 4213 | 694 | 7472 | 329 |
| 0.625 | 1.55 | 0.5608125 | 4.763565 | 4247 | 834 | 7332 | 295 |
| 0.6 | 1.6 | 0.60912 | 4.86018 | 4275 | 1000 | 7167 | 267 |
| 0.575 | 1.65 | 0.6574275 | 4.956795 | 4305 | 1189 | 6977 | 237 |
| 0.55 | 1.7 | 0.705735 | 5.05341 | 4358 | 1382 | 6784 | 184 |
| 0.525 | 1.75 | 0.7540425 | 5.150025 | 4393 | 1618 | 6548 | 149 |
| 0.5 | 1.8 | 0.80235 | 5.24664 | 4420 | 1860 | 6306 | 122 |
| 0.475 | 1.85 | 0.8506575 | 5.343255 | 4438 | 2118 | 6048 | 104 |
| 0.45 | 1.9 | 0.898965 | 5.43987 | 4450 | 2338 | 5828 | 92 |
| 0.425 | 1.95 | 0.9472725 | 5.536485 | 4455 | 2587 | 5579 | 87 |
| 0.4 | 2 | 0.99558 | 5.6331 | 4463 | 2805 | 5361 | 79 |

根据本节所提出的PMM算法，在对正常阈值上下设定不同的限定参数，之后，得到上表基本信息，根据上表，本实验可以计算出五个经典指标信息，并且通过matlab平台得到图3.1：



­­­图3.1 评价指标图

可以清楚观测到，有三个关键位置，分别标注为①②③，对于关键位置1而言，表明测试模型对于正常流量的识别达到最高，并且对所有攻击样本中真正攻击所占比例达到最高，且被错误识别的正常流量少，整体准确率达到最高，对于关键位置2来说，作为过渡数据，弥补了1处攻击样本识别差，相对于3来说，弥补了除去攻击识别样本以外其他指标缺陷，对于关键位置3来说，攻击样本识别能力最强，但是，错误识别为攻击流量的正常流量比例太高，综上所述，选取关键位置1作为最终阈值。即正常距离应该落入范围[0.319275e-05,4.28049e-05]中。此时监测系统整体准确率到达最高。但是模型对于攻击识别能力稍有欠缺，也是今后检测模型体要提升的地方。

4 基于BP\_Adaboost强分类器的DDoS攻击检测算法

在上一章节中，检测模型仅仅完成了检测DDoS攻击，将正常流量与异常流量区分开来，属于简单二分类问题，但是在数据集中，Dos攻击共包含六大类，如何对异常流量进行进一步的分类，则是本节需要探讨的问题。BP神经网络是一种效率高的多层神经网络，利用“信号向前传，误差向后传”原理，为了降低整体网络误差率不断调整网络状态，最终使得结果与预期尽可能接近。本实验将BP神经网络看作为一个弱分类器，利用Adaboost算法将若干弱分类器集合起来构成一个更强的分类器。实验结果证明，分类器更高效地将异常数据进一步细分类，并且强分类器相对于弱分类器来讲，可将正确率从78%提高至96%以上。

4.1 弱分类器——BP神经网络

随着人工智能的发展，人工神经网络得到发展，这种网络是通过模拟大量生物神经元之间传递信息的机器学习模型。目前使用最广泛的是BP神经网络，即误差逆向传播(Back Propagation, BP)算法，对于一个多层神经网络，一般由输入层，输出层，及隐藏层组成。假设一个多层神经网络共有L层，其中第一层为输入层最后一层为输出层，除此以外，中间的神经网络层均为隐藏层。

4.1.1 BP神经网络结构及正向传播

假设一个多层神经网络总共有L层，则其结构如图4.1所示：

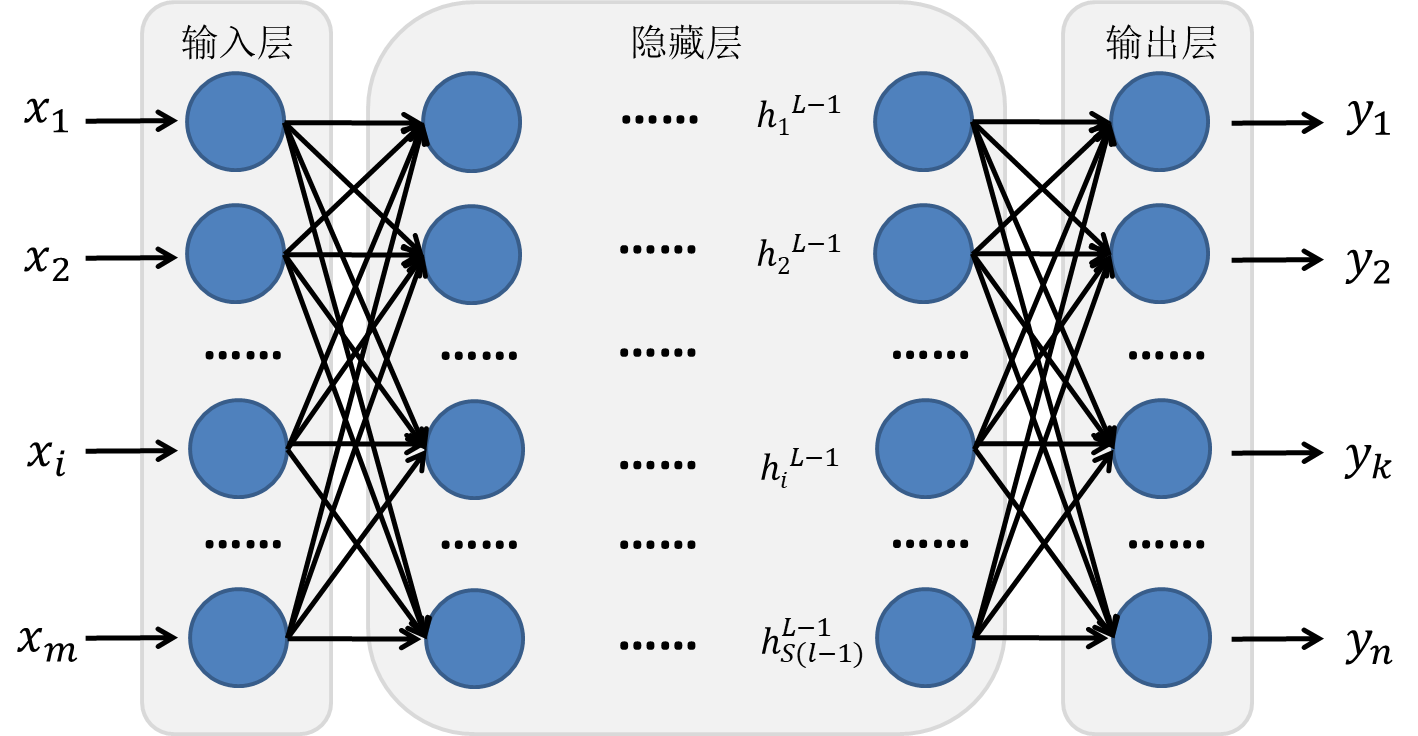


图4.1 多层(L层)神经网络结构图­­­

假设对于第一层共输入*m*个样本，要求分成*n*类，则输出层需要输出*n*个结果。令输入向量为，对应输出向量为，对于隐藏层来说，假设第*l*层隐藏层共有*S(l)*个神经元，第*l-*1层隐藏层共有*S(l-*1*)*个神经元，则每一个神经元的输出向量为。对于BP算法来讲，每一次的迭代过程都需要对权重与偏置进行更新。即设表示*l-*1层的第*j*个神经元与第*l*层第*i*个神经元之间的连接权重并且设为偏置。那么对于第*l*层的第*i*个神经元，其输入值公式则为：

(公式4.1)

设*F(x)*为神经元的激活函数，根据上式所得，其输出则为。

对于激活函数而言，BP神经网络通常使用非线性激活函数，而不是用线性函数，道理显而易见，如果使用线性激活函数，那么经过多层神经网络之后，仍然是之前线性计算的叠加，神经网络层数变得没有意义，因此选择非线性激活函数完成线性函数所不能完成的任务。

激活函数包括很多但是其中常用的激活函数共有两种，分别为sigmod函数与双曲正切函数，其公式分别如公式4.2，公式4.3所示：

(公式4.2)

(公式4.3)

本实验使用matlab软件对以上激活函数画图可看出，其函数走势分别如图4.2所示：

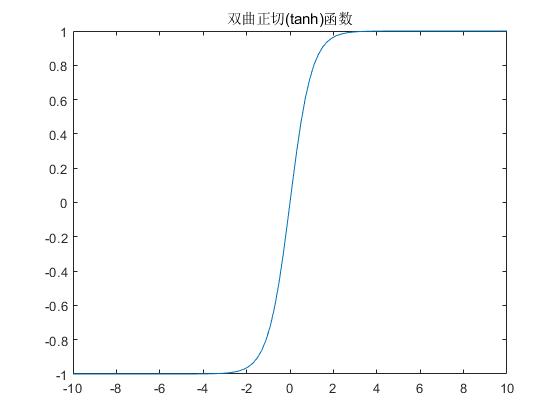
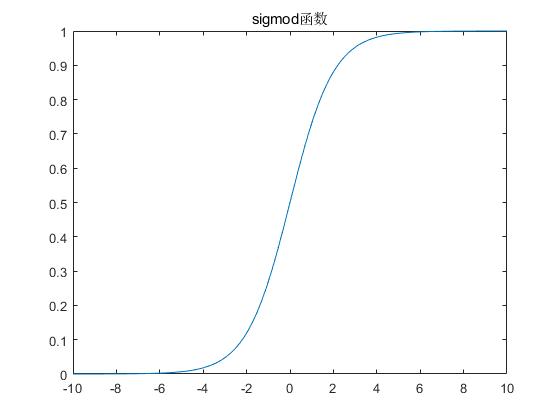


图4.2 神经网络激活函数走势图

从图中可以看出对于sigmod函数来说，其取值范围为(0,1)，并且求得其导数：，但是对于双曲正切函数，其函数取值范围为(-1，1)，其导数为：。

4.1.2 误差反向传播

对于BP神经网络来说，将误差从输出层沿着向输入层方向反向传播也就是利用误差反向传播原理，使得网络中每一层权值更新。对于这个网络本实验首先可以定义网络总误差函数为，其中单个*E(i)*是指单个样本的误差，对于最终输出的*n*个结果，假设*d(i)*为对于输入*x(i)*的期望输出，则平方和误差函数如公式4.4所示:

(公式4.4)

对于每一次的迭代，BP算法对于权值以及偏置的更新是其算法关键之处，更新公式分别如公式4.5，公式4.6所示：

(公式4.5)

(公式4.6)

对于上述公式中为学习速率，其取值范围为(0,1)，对于每一次的迭代，BP算法对于权值以及偏置的更新都是通过梯度下降法最终分别求得其偏导数如公式4.7，公式4.8所示：

(公式4.7)

(公式4.8)

4.2 强分类器——BP\_Adaboost分类器

在讲Adaboost算法之前，首先要了解boosting提升算法，对于boosting算法中 ，其主要思想为对于每一个弱分类器，首先采用初始权重去训练弱分类器，然后每一轮重新计算弱分类器的误差率，根据求得的误差率在重新更新下一轮分类器权重，例如在第一轮中样本误差率较高的分类器，在下一轮训练中，加大权重，使其得到重视。

Adaboost属于其中一种boosting算法，是指将若干弱分类算法反复学习训练组合成一个强的分类算法，对于提升算法来讲需要涉及到两方面，一个是加法模型，另一个是前向分步算法。加法模型指一个强分类器是由若干个弱分类器线性组合而成，前向分布是指下一轮迭代产生的分类器是在上一轮训练基础上训练而成。

4.2.1 Adaboost算法分类问题

在上一节的实验中，本实验已经采用距离算法实现二分类将正常流量与异常流量分开，在本次实验中本实验将继续对于六种DDoS攻击加以区分，加上对正常流量的区分相当于进行七分类，由于多元分类问题是对二元分类的推广，本小节介绍简单二元分类，首先计算器误差率*e*，假设二元分类的输出为{-1,1}，由于采用权值思想误差率则可使用加权平均计算，那么第*k*个弱分类器*Gi(x)*的加权误差如公式4.9所示：

(公式4.9)

其中表示训练数据集，则表示第*i*个弱学习器的输出权重。初始对于第一个弱学习器来说，则每一个样本权重都为1*/m*，对于二元分类来说，根据上一步求得的误差率，第*k*个弱分类器的权重系数如公式4.10所示：

(公式4.10)

根据公式可以看出，对于分类误差越大的弱分类器，对应的权重将会减少。使得在下一轮训练中削弱其权重大小。对于样本权重的更新，首先假设第*k*个弱分类器其中*m*个样本中，样本集权重系数为，则下一次训练的弱分类器样本集权重为公式4.11：

(公式4.11)

按照公式不难理解，计算第*k*个弱分类器的每一个样本权重对应所占比例。此时，再看本节一开始介绍的boosting算法，如果在分类器中，某一个样本分类错误，使得，使得权重变大，因此在下一轮的考虑值变大。最后通过采用加权平均求和，将若干弱分类器集合成一个强分类器，即为公式4.12：

(公式4.12)

至此，将若干BP神经网络弱分类器集合成一个强分类器BP\_Adaboost完成。在构建完成强分类器之后，接下来将进行分类实验。

4.3 数据集介绍：

本次实验数据仍采用与上一章节同样的数据集， KDD99仍是当前网络安全领域最有效的数据集，本节实验采用10%测试数据集共有488736条。

4.3.1 数据集处理

在每一条数据记录中共有41项数据特征，有三种是非数值型，首先本实验需要将其转化为数值类型，详细的数据转化类型详见表4.1：

表4.1 数据转化类型表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 顺序 | 特征名称 | 类型转化1 | 类型转化2 | 类型转化3 | 类型转化4 |
| 2 | protocol\_type | TCP=1 | UDP=2 | ICMP=3 | 无 |
| 3 | service | ecr\_i=1 | private=2 | http=3 | 其他=0 |
| 4 | flag | SF=1 | 其他=0 | 无 | 无 |

为了能够保留所有数据特征，本次实验不做降维处理，因此，数据中总量为：41\*488736。由于实验数据分类平台依托于软件matlab，所以.mat文件中，存放有C\_label,C\_data矩阵，一个用于存放数据集，一个用于存放数据标记，表明数据分类情况，为以后训练以及判断分类正确率奠定基础。

4.4 实验结果与分析

本实验将BP神经网络看作弱分类器，共输入488736条流量41项数据特征。考虑到时间代价等问题，本实验将中间隐藏层设为1，将BP分类器进行弱化，此时神经网络结构如图4.3所示：

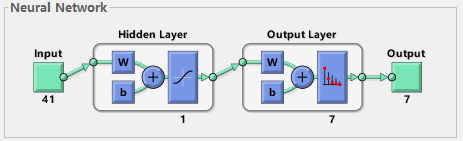


图4.3 分类器训练图

对于BP神经网络，本次研究所做共30个实验，，隐藏层层数分别从1-6，训练数据与测试数据，从1:1到1:5共30组实验，并将此30个实验正确率做出对比，如图4.4所示：

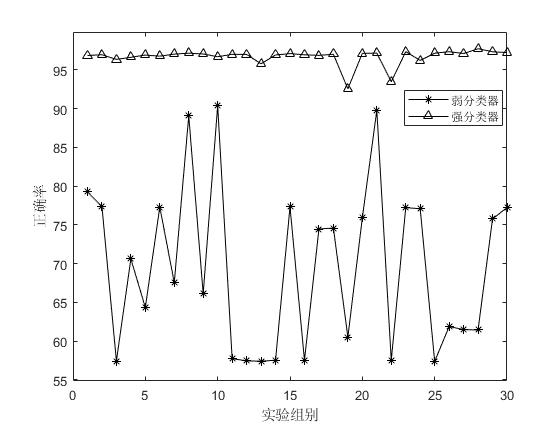


图4.4 强弱分类器正确率对比图

通过实验数据以及上图可以明显看出本实验将隐藏层数量选定在1-6内，反复进行实验，改变隐藏层数量以及测试训练数据，但最终都发现相较于弱分类器而言，强分类器明显将正确率提升。

6 简易检测系统的设计与实现

在上一节中，在完成最后一个实验之后，对于三个算法做出比较与判断，发现实验二有监督学习效果最为理想，并且可以对于DDoS六种攻击进一步做出判断，基于此，本章利用matlab软件设计GUI界面，实现加上正常流量，对于七种类型流量的区分。对于简易系统设计来说，主要针对实验二开展，利用强分类器BP\_Adaboost原理实现，系统展示了采用弱分类器与强分类器所得的正确率，并且实现了对于输入的数据进行检测判断。

6.1 系统设计

在系统设计过程中，主要分为四大步骤，首先是对于弱分类器与强分类器正确率的展示，其次是对数据的选择，用户可以选择自己输入41维数据特征，也可以选择系统提供的待测数据进行检测。在完成数据输入之后，利用强分类器进行模型训练，完成训练之后，最终查看判断类型，并将结果呈现。具体来说为：

1. 正确率的显示：首先为了突出强分类器优于弱分类器，模型设计中，用户可以首先进行两者正确率的测试，点击对应按钮之后，此时matlab选取9264条数据，将数据随机抽取一半作为训练数据，其余一半则为测试数据，然后将测试数据将最终分类准确性显示到对应显示框中。
2. 输入数据：对于此部分，系统会给出七种数据流量的待测数据，但是在测试时，系统不知道这七种流量的类别，或者用户可以自己选择输入，此时，点击确认按钮后，系统会自动将输入数据放入测试数据中，并且将此条流量类别设置为0，即系统事先不知此流量种类。为下一步测试做好数据准备。
3. 模型训练：此时matlab仍将9264条数据作为基础，加上刚才输入的一条数据，在训练了4632条数据后，将剩余4633条数据作为测试数据，最终得到预测数据类别。
4. 结果判断与显示：最后一步中，将上一步测试预测分类类别做出判断，将其代表的攻击类型输出到系统最终的检测框中，完成系统检测功能。

6.2 系统功能实现

基于上一节系统设计原理，最终采用matlab语言在Windows10下利用matlab(R2018b)进行系统编写。图6.1则为系统的原型界面设计图：



图6.1 系统原型界面设计图

对于正确率的显示来说，点击系统按钮选择不同分类器即可完成不同分类器的准确性检测。图6.2与图6.3分别为弱分类器与强分类器的准确性显示图：



图6.2 弱分类器准确性检测图

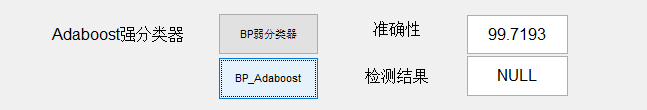


图6.3 强分类器准确性检测图

对于数据流量检测来说，数据分为待测数据与用户自行输入，比如，当选择数据为Normal的时候，在检测结果中，则会显示强分类器的分类结果，在下面对应的41个数据框中，则会一一对应显示数据41维特征所对应数据，测试结果如图6.4所示：



图6.4 系统检测正常流量图

不仅仅是对于正常流量的检测，在系统检测中，对于其他六种DDoS攻击类型均检测成功。至此，系统功能检测完毕。

7 总结与展望

7.1 论文实验工作量

本文主要针对DDoS检测问题，运用监督学习方法以及无监督学习方法，将正常异常流量区分，并且对于DDoS六种异常流量进行进一步分类(聚类)。三个实验都在matlab平台上完成，并且所用的数据集都是KDDCup99，对于实验一来说，训练8166条正常流量数据，并将原数据根据各自特点进行降维处理，在计算得到两两特征值之间的三角形面积之后，计算其马氏距离，将4542条异常数据进行相同处理，划分不同的阈值对于流量进行区分。实现简单的二分类正常异常判断。对于实验二来说，实验效果最好并且更加适合数据量大的情况，将多个BP神经网络集合成为一个新的强分类器。在不断调整隐藏层层数或者训练以及测试数据个数，实验结果表明集合生成的强分类器都要比单个弱分类器效果要好得多。因此对于实验二来说，数据量大并且实验效果较为稳定。对于实验三，首先实验中发现，对于数据样本之间，如果样本数量差距较大的情况下，数量较小的样本不易聚为一类，并且不易发现，针对利用Kohonen网络聚类时，小样本数据不易发现的特点，实验三对于输入数据进行降维去噪处理，但是存在实验效果不稳定的情况。

7.2 实验中出现的问题以及展望

本节我将针对实验中出现的问题以及反思详细写下：

对于实验一来说，我认为实验的不足是数据量小，原数据量将近50万，但是实验一只选取了其中8166条正常数据进行训练，测试集有4542条异常流量。对于实验数据量没有达到理想效果。实验一本文实验参考北京邮电大学贾斌博士第一节算法，相较于原作者，本文仅仅提高了一项指标，由于仅仅是对流量进行二分类，我认为此算法在今后发展中，根据不同的阈值范围可以将异常也进行区分。特别是在数据量大的情况下，算法消耗时间大，因此算法比较适合于简单二分类，如果将算法进一步提升为精确分类的算法，需要改进算法速度。降低时间复杂度。

对于实验二来说，实验的数据量比较理想，结果也比较理想，在这三个算法中，正确率比较稳定，但是对实验二来说，再近期读到的论文中，北京邮电大学博士将多个不同种类的分类器组合起来，构成混合异构多分类器学习模型，利用Rotation Forest的分类集成策略，实现将特征提取用到特征子集，并且可以在集成的系统中实现对每个分类器重新构建全特征集合。因此在以后的发展中，实验应该使用更为高效的集成学习策略，并且不只是针对BP神经网络，应该集成多种类的分类器，这样不仅可以对于攻击检测进行分类，将来可以扩展到对于其他种类数据进行监督学习，针对不同的数据，发挥不同优势分类器，不断提高其权重，扬长避短使得整体分类器准确率更加精准。

对于实验三来说，与实验一相同，存在数据量少的问题。在实验中，有几组数据不太理想，正确率两级分化率较高。如下表所示，将数据降到七维时，虽然整体准确率较高，但是却不能忽视对于teardrop分类准确率只有不到30%，但是前几个类别分类的准确度高达99%以上，因此最终的准确率判定不能仅仅只加权平均，应该有所改进。对于实验中，失败的数据种类不能忽视。在这次实验中体会到了总体数据的准确性并不能代表成功的概念。应该在以后的研究加强重视。本实验仅仅对输入数据进行了去噪处理，而对于本身的Kohonen网络并没有进行改进，在本文大量的实验基础上，发现，对于输入数据的去噪可以在一定程度提高准确率，但是如果对于网络本身加以改造，在论文中，河北大学硕士对网络进行改造使得正确率提高，但改造后的网络是监督学习，不属于无监督学习范畴，因此本文只对Kohonen输入数据进行处理。

以上是我对今后实验研究方向的展望，希望进一步解决实验一遇到数据量的问题，以及对于实验三来说如何对于小数据量的数据在聚类中，减少误差。希望在以后的研究中继续深造。

参考文献

1. 贾斌. 基于机器学习和统计分析的DDoS攻击检测技术研究[D].北京邮电大学,2017.
2. 电信云堤. 2018年度DDoS攻击态势报告 [EB/OL]．(2019-02-28) [2019-4-16]．<https://mp.weixin.qq.com/s/Ra5JsR6dOjos81OyxBBXVg>．
3. 张圆园. 基于无监督聚类的网络入侵检测方法研究[D].北京理工大学,2016.
4. 董宁. 基于S-Kohonen神经网络的拒绝服务攻击检测算法的研究[D].河北大学,2017.
5. 于冰洁. 基于高斯模型的异常检测算法[D].中国矿业大学,2017.周志华.机器学习[M].清华大学出版社.2016
6. 李秋霞. 基于地图的网络攻击可视化系统设计与实现[D].电子科技大学,2018.
7. 陆胜雷. ACO-BP算法优化及其在入侵检测中的应用[D].重庆师范大学,2018.
8. 库凯琳. 基于改进遗传算法的贝叶斯入侵检测研究[D].湖北工业大学,2017.
9. 徐温雅. 基于机器学习的网络入侵检测研究[D].北京交通大学,2018.
10. 王明. 基于卷积神经网络的网络入侵检测系统[D].北京邮电大学,2018.
11. 孔令爽. 基于深度学习和迁移学习的入侵检测研究[D].山东大学,2018.
12. 张浩. 一种新型分类算法及其在网络入侵检测中的应用研究[D].北京邮电大学,2018.
13. Alan Saied, Richard E. Overill, Tomasz Radzik, Detection of known and unknown DDoS attacks using Artificial Neural Networks[J].Neurocomputing,2016,172:385-393.
14. Abdulaziz A , Shahrulniza M . Cloud-Based DDoS HTTP Attack Detection Using Covariance Matrix Approach[J]. Journal of Computer Networks and Communications, 2017, 2017:1-8.
15. Da Y , Lianming Z , Kun Y . A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework[J]. IEEE Access, 2018, 6:24694-24705.
16. Z. Tan, A. Jamdagni, X. He, P. Nanda and R. P. Liu. A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis[J]. *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 447-456, Feb. 2014. doi: 10.1109/TPDS.2013.146

致谢

通过几个月的努力，我如期完成了我的本科毕设工作，在做毕设的过程中，首先我重新学习了matlab语言，并且对于神经网络得到了一些了解，在选题过程中要特别感谢自己未来研究生导师的意见，在阅读了一些国内外的论文中提高了自己的阅读与理解能力，并且在阅读过程中，选定了自己的实验所用数据集以及实验方法。特别在神经网络方面，阅读到了许多现在广泛使用的神经网络并且了解到了一些优化技术。感谢张少强老师在我写论文期间，对于算法上的指导，在我开题报告时，对我的文章审核，耐心指导我得写作论文的思路，张少强老师对我的帮助从转专业刚来开始，使我在新班级新的学习专业种给了我莫大的鼓励与帮助。在此特别感谢。

大学四年时光匆匆而过，回想起来自己的四年，仍有许多感慨。从大一初来乍到感觉迷茫，到大一下学期鼓起勇气我选择了转专业。在这里我要感谢我的论文指导老师张少强老师，在我从大一转专业来到新班级新专业的时候，感谢我的班主任张少强老师一路上的帮助，感谢胡燕翔胡院长在转专业面试中给了我学习计算机的机会。让我对自己今后的发展方向更加明确目标。感谢来到班级里所有帮助过我的同学。大二的不安，让我在磕磕绊绊中越来越看清楚计算机学习的本质，我记得张立新老师的数据结构是我大二初来计算机遇到最大的门槛，现在想起感谢张立新老师在上机课上耐心为我调试程序，在博理楼的二层每周五下午的上机课，每一次张立新老师都会为我调试程序，对于插入节点问题给我讲得很详细。我还记得大二上课到了中午，下课之后仍不辞劳苦给我讲解课后题的李燕荣老师，对于数字电路这门课，我仍记得李老师中午给我讲组合逻辑电路的设计，我也记得下了课遇到操作系统的问题，怎么也算不对的预测死锁问题，孙华志院长的耐心指导。杨嵘老师在301大机房给我调试C++，在C++语言中有一章写数据结构栈，杨嵘老师耐心给我讲解。匆忙而又充实的大二过去了，在这一年中，有了老师们的一路帮助，我逐渐对于计算机有了自己的见解。

到了最关键的大三，这一年里，从上学期到下学期我要感谢胡燕翔老师从计算机组成原理讲到了计算机系统结构，我也记得中午下课，胡老师给我们继续讲cache的三种映射方式，我记得胡燕翔老师上课讲完系统结构之后，下课曾问我听懂没有，感谢胡老师大三一年学习上的关心。感谢张占营老师，在大三的时候带我做项目，让我了解到了目前流行的前端技术，还有springboot的后台技术，加上刘洋老师讲授的数据库知识，让我搞明白了前端，数据库，后台的技术并且亲自进行尝试。感谢于洋老师，从大二的单片机的实验课到大三的编译课程，感谢于洋老师耐心地讲解，有很多书本中体会不来的知识，在硬件实际操作中让我有了后知后觉的觉悟。

大学四年，匆匆而过，我到现在清清楚楚的记得转专业的时候胡燕翔院长在面试的时候问我的三个问题，1：什么是大数据，2：什么是云计算，3：什么是指针，大一的我对于前两个问题没有任何概念，第三个问题也只是磕磕绊绊回答了上来，到了现在，即将结束我得本科生活的时候，我想我现在想明白了前两个问题，感谢所有计信学院帮助过我的老师，让我在这条道路上越走越踏实。感谢高考让我选择天津师范大学，道阻且长，未来可期，希望自己在这条路上越走越远。太多的感谢送给天津师范大学计算机与信息工程学院全体老师！

附录：代码

%BP\_Adaboost强分类器算法

clear; clc; %清除工作区

load matlab2.mat C\_data C\_label; % 加载自己所建工作区以及相关数据

Data = C\_data; % 将所使用数据挑选

Label = C\_label; % 将所使用标签提取

indices = crossvalind('Kfold',488736,6);

% 在48万多条数据中按照1:6的比例，随机选出训练数据以及测试数据

trainindices = (indices == 1);%指代为训练数据

testindices = ~trainindices; %指代为测试数据

traindata = Data(:, trainindices);

testdata = Data(:, testindices);

trainlabel = Label(:, trainindices);

testlabel = Label(:, testindices);

%根据上两个式子，创建四个变量，分别为训练数据，测试数据，以及训练测试数据

%对应的分类类别

%数据归一化（可选）

[traindata,PS]=mapminmax(traindata,0,1);

testdata=mapminmax('apply',testdata,PS);

% setdemorandstream(pi); % 固定随机值,

% 训练i个弱分类器

i =10;

adaboost\_model = Ada\_train(@Threshold\_tr, @Threshold\_te, traindata, trainlabel, i);

% 用adaboost法训练10个弱分类器

% 训练样本测试

% [L\_tr,hits\_tr] = AdaBoost\_te(adaboost\_model, @Threshold\_te, traindata, trainlabel); %此函数输出第一项为训练分类结果，第二项为正确分类的样本个数

% tr\_n = size(traindata,2);%表示训练数据的规模

% tr\_error = (tr\_n-hits\_tr)/tr\_n;%计算错误

[L\_te,hits\_te] = AdaBoost\_te(adaboost\_model, @Threshold\_te, testdata, testlabel);

te\_n = size(testdata, 2); % 得到测试数据的数量

te\_error = (te\_n - hits\_te) / te\_n \* 100; % 错误率

Accuracy = 100 - te\_error; % 正确率

%kohonen神经网络聚类

sample=9264; %本实验共 9264个样本点

trace=41; %每个数据共有41维特征

load data6.mat %下载工作区

A=data1(:,1:41); %定义变量，选取工作区中数据及其41维特征

min0=min(size(A)); %得到数据维度，本实验中min0=41为下面循环操作奠基础

[U,S,V]=svd(A); %进行SVD去噪计算

for i=1:min0 %循环41次，得到一维数组x，记录41个特征向量

x(i)=S(i,i);

end

for i=1:min0 %循环41次，将特征值平方处理，下一步则按照数平方计算贡献率

k(i)=x(i).^2;

end

SumValueEnergy=sum(k(1:min0)); %总体贡献率相加求和

for i=1:min0

kk(i)=k(i)./SumValueEnergy; %每一个奇异值的在总的奇异值中的贡献率

end

for i=1:min0 %累积能量分布;

kkk(i)=sum(k(1:i))/SumValueEnergy;

end %kkk(i)指代前i个贡献率之和

for i=1:min0

if sum(kk(1:i))>=0.9 %前i个奇异值的累积贡献率之和，第一个到达0.9值的奇异值。即可代表整个数据

END=i;

break;

end

end

partvalue=END;

S1= (sample,trace); %生成大小为9264\*41的零矩阵

for i=1:partvalue %部分特征值的拐点

S1(i,i)=x(i); %改变原S矩阵

end

f=U\*S1\*V';%部分特征值构建出的矩阵

input=f(:,1:41); %将去噪后的矩阵当作新的输入

[row,column]=size(input); %计算输入矩阵的规模column=41,row=9264

Inputnum=41;

M=6;

N=5;

K=M\*N;%即本实验Kohonen采用总节点数为30

%Kohonen层节点排序

k=1;

for i=1:M

for j=1:N

jdpx(k,:)=[i,j];

k=k+1;

end

end

maxstudyrate1=0.1; %表示最大学习率

minstudyrate1=0.01; %表示最小学习率

maxstudyr1=1.9; %最大学习半径

minstudyr1=0.2; %最小学习半径

w1=rand(Inum,K); %随机第一层权值

%本实验设置迭代求解次数5000次

maxgen=5000;

for i=1:maxgen

%自适应学习率rate1与相应半径r

rate1=maxstudyrate1-i/maxgen\*(maxstudyrate1- minstudyrate1);

r= maxstudyr1-i/maxgen\*( maxstudyr1- minstudyr1);

%从数据中随机抽取

k=unidrnd(9264); %matlab产生9264数字内的一个随机数，假设此时k=6089

x=input(k,:); %在输入数据中选取第6089行数据，作为初始节点

[mindistance,index]=min(dist(x,w1)); %计算距离最小的最优节点

d1=ceil(index/5); %求整数部分

d2=mod(index,5); %求余数部分

nodeindex=find(dist([d1 d2],jdpx')<r);

%计算其周围节点，即距离小于相应半径r

for j=1:K

if sum(nodeindex==j)

w1(:,j)=w1(:,j)+rate1\*(x'-w1(:,j));

end

end

end

Result=[]; %存放实验最终聚类结果

for i=1:9264

%循环9264次，最终在Index数组中，共9264列，代表数据最终归属为哪一类

[mindistance,index]=min(dist(input(i,:),w1));

Result=[Index,index];

end