

学士学位论文

基于频繁模式挖掘的网络异常事件检测系统

计算机科学与技术

学生 李盛乐 指导导师 周颖杰

SiChuan University

Network abnormal event detection system based on frequent pattern mining

Computer science and technology

Student’s Name Shengle Li Supervisor Yingjie Zhou

**摘要**：随着互联网时代的到来，特别是移动互联网的高速发展，web服务和微服务已经深入到人们生活的衣食住行，人们使用互联网进行搜索、购物、支付、娱乐等等。因此，保障web服务的稳定运行变得十分重要。Web服务的稳定运行主要靠运维人员来保障，运维人员通过监控各个关键性能指标（KPI）来判断web服务是否发生异常，KPI 异常的发生往往意味着相关的应用发生了异常。面对大量的KPI数据，亟需通过高效的网络异常事件检测系统帮助运维人员检测网络异常事件。围绕着如何通过频繁模式挖掘的算法建立一个网络异常事件检测系统，辅助运维人员对网络异常的检测，对频繁模式挖掘算法进行了研究和对网络异常事件检测系统进行了设计研究，主要研究的内容和贡献包括：（1）对频繁模式挖掘算法的研究和使用：对频繁模式挖掘算法中的经典算法Apriori算法进行了研究，Apriori算法的核心思想是通过候选集生成和情节的向下检测两个阶段来挖掘产生频繁项集，在算法执行的过程中伴随着剪枝的操作，最后通过最小支持度得到频繁项集；（2）网络异常事件检测系统的设计和实现：设计和实现了一个网络异常事件检测系统，该系统的主要功能包括：a.选择训练集和测试集数据；b.绘出指定KPI的曲线图和异常点，以及异常的占比；c.对训练集进行训练后，可选择不同类型的KPI进行检测；d.计算并展示检测的精确率、召回率和得分。

**关键词：**频繁模式；Apriori；数据挖掘；网络异常；关键性能指标（KPI）

**Abstract:** With the advent of the Internet age, especially the rapid development of the mobile Internet, web services and microservices have gone deep into people's lives. People use the Internet for search, shopping, payment, entertainment and so on. Therefore, it is very important to ensure the stable operation of web services. The stable operation of web services is mainly guaranteed by the operation and maintenance personnel. The operation and maintenance personnel determine whether the web service is abnormal by monitoring each key performance indicator (KPI). The occurrence of an abnormal KPI often means that the related application has an exception. Faced with a large number of KPI data, it is urgent to use an efficient network anomaly event detection system to help operation and maintenance personnel detect network anomalies. A network anomaly event detection system was built around how to use frequent pattern mining algorithms to assist network operators in detecting network anomalies, and frequent pattern mining algorithms were studied and network anomaly event detection systems were designed and researched. The contents and contributions include: (1) Research and use of frequent pattern mining algorithms: The Apriori algorithm, a classical algorithm in frequent pattern mining algorithms, is studied. The core idea of ​​Apriori algorithm is to generate candidate sets and detect the two downwards. Stages to mine frequent itemsets, with pruning during the execution of the algorithm, and finally obtain frequent itemsets with minimal support; (2) Design and implementation of network anomaly event detection systems: design and implementation Network anomaly event detection system, the main functions of the system include: a. Select training set and test set data; b. draw the specified KPI graph and abnormal points, and the proportion of abnormal; c. After training the training set Different types of KPIs can be selected for detection; d. Calculate and display the accuracy of detection, summon Rate and score.

**Key words:** Frequent mode; Apriori; Data mining; Network abnormal; Key performance indicators

目录

[第1章 绪论 7](#_Toc513622482)

[1.1 研究背景及意义 7](#_Toc513622483)

[1.2 研究现状 8](#_Toc513622484)

[1.1.1 异常检测算法现状 8](#_Toc513622485)

[1.1.2 频繁模式挖掘算法现状 12](#_Toc513622486)

[1.3 本文研究内容 13](#_Toc513622487)

[1.4 论文结构安排 13](#_Toc513622488)

[第2章 频繁模式挖掘算法研究 14](#_Toc513622489)

[2.1 频繁模式挖掘算法介绍 14](#_Toc513622490)

[2.1.1 概念 14](#_Toc513622491)

[2.1.2 Apriori算法 14](#_Toc513622492)

[2.1.3 FP-Growth算法 15](#_Toc513622493)

[2.1.4 Eclat算法 15](#_Toc513622494)

[2.1.5 三种算法对比 15](#_Toc513622495)

[2.2 Apriori算法 16](#_Toc513622496)

[2.2.1 基本概念 16](#_Toc513622497)

[2.2.2 算法步骤 16](#_Toc513622498)

[2.2.3 算法优缺点 17](#_Toc513622499)

[2.3 研究方法 18](#_Toc513622500)

[2.3.1 数据集下载及预处理 18](#_Toc513622501)

[2.3.2 Apriori算法设计与实现 18](#_Toc513622502)

[2.4 实验结果及结论 19](#_Toc513622503)

[2.5 本章小结 20](#_Toc513622504)

[第3章 网络异常事件检测系统的设计和实现 20](#_Toc513622505)

[3.1 需求分析 20](#_Toc513622506)

[3.2 系统功能设计 21](#_Toc513622507)

[3.2.1 读取文件 21](#_Toc513622508)

[3.2.2 绘画KPI曲线图 21](#_Toc513622509)

[3.2.3 异常频繁模式挖掘 21](#_Toc513622510)

[3.2.4 测试及评估算法 22](#_Toc513622511)

[3.2.5 导出测试集 22](#_Toc513622512)

[3.3 数据流图 22](#_Toc513622513)

[3.4 系统功能实现 22](#_Toc513622514)

[3.4.1 系统界面设计 22](#_Toc513622515)

[3.4.2 系统后台实现 23](#_Toc513622516)

[3.5 设计结果 24](#_Toc513622517)

[3.6 本章小结 25](#_Toc513622518)

[第4章 总结与展望 25](#_Toc513622519)

[4.1 总结 26](#_Toc513622520)

[4.2 展望 26](#_Toc513622521)

[参考文献 27](#_Toc513622522)

# 

# 绪论

## 研究背景及意义

随着计算机网络的飞速发展[1]，网络服务逐步发展到人们生活的每个方面，特别是移动互联网时代的到来，web服务更是和我们的生活紧密关联在了一起。在现今的中国，利用手机支付，使用共享自行车，在线进行购物，预定酒店、机票等，都必须有web服务的支持。如今的社会没有web服务是难以想象的。由此可见，web服务对于现在社会已经不可或缺，没有web服务社会是难以想象的，确保其平稳性、能动性和高效性十分的重要。与此同时，很多不法分子利用一些技术对网络和服务进行攻击，此类针对服务的流量攻击威胁问题十分严重，造成服务的瘫痪、网络的拥堵和信息泄露等很多严重的问题，如SYN洪水、特洛伊木马、DNS高速缓存污染等网络攻击频繁发生，给网络的安全性和web服务的稳定性带来了严重的威胁和不良的影响。如何在高速网络环境下对网络异常行为进行及时感知和快速处理，对于保证网络有效运行和提高服务提供能力的稳健性具有非常重要的意义[2] 。

现有的网络异常检测手段，大部分是依靠人力查看关键性能指标（KPI）来实现的。这种方法需要监控者持续不断的查看各类线上数据，而这些数据的状态也代表着关联应用的状态。指标数据包含服务数据和机器数据，如今应用服务每天需要处理海量的服务请求，由此产生的KPI数据十分庞大，面对大量的网络KPI数据，仅靠人工监控以无法满足对服务异常检测的需求。而如今大数据挖掘和分析技术以十分成熟，在交通、金融、天气等方面均有成功的应用，结合大数据技术设计网络异常检测算法有很大的研究意义。

通过运维人员指出的异常数据的特征，使用数据挖掘分析，以及深度学习的技术，不断学习异常的模式和规则，不断优化检测的算法，找出最有效的检测方法，从而通过学习所得的最优模式对测试数据进行检测，这就是网络异常检测的定义。异常检测在生产环境中还有大量的困难和缺陷，导致其不能运用于实际场景中，主要的困难如下：

1.相对于正常数据，异常事件很少发生。现实的web服务经过各种测试和使用后改进，已经很少发生异常情况了。相对于海量的指标数据，其中异常的数据只是很少的部分，有些类型的数据甚至难以找到异常，可用于解析的数据很少。但是异常发生时又严重影响用户的使用，异常的检测不得不进行。

2.异常的类型丰富。网络异常行为通常具有不同的模式，且隐藏在复杂的背景流量中[3]。不同类型的服务也意味着不同类型的异常，某些算法在一种类型的异常中表现很优秀，但是却无法检测另一种类型的异常。所以大量类型的异常也给检测算法的设计带来了困难。

3.关键性能指标种类繁多。关键性能指标有各种不同的模式，它们所呈现的规律不一样，所有很难设计通用的算法。

4.界限难以区分。在一些服务中，判断数据是否异常没用明确的标准，仅仅依靠工作人员的经验，间于正常与异常之间的数据更是难以判断。

目前存在的困难，造成了检测的结果并不理想，不仅没能真正的帮助运维人员检测异常，还给他们的工作造成了干扰，浪费了人力和物力在不准确检测结果上。因此，研究和设计一个能够有效检测异常事件的系统，辅助运维人员在现实场景中找出web服务数据中的异常，对于减少运维人员的工作量和保证服务的稳定性十分有意义。

## 研究现状

### 异常检测算法现状

#### 基于分类的检测算法

由标签的种类可以将此种算法归纳为两类：多分类问题和单分类问题。基于分类的检测算法种类十分多，以下介绍一些常见的算法：

1. 基于神经网络

神经网络是指包含多个引入和引出节点，所有的节点相互连接，并且所有的连接采用权重进行权衡。基神经网络一般有三层即输入层、隐含层和输出层[4]。其在单分类和多分类事件中均可使用。该异常检测算法的主要步骤分为两步：

1. 首先需要得到一个神经网络，此神经网络利用用户输入的样例文件，对样例进行训练获得，通过它获得各种正常分类。在学习的过程中不断修正网络节点连接的节点阈值和权重。
2. 利用学习所得的神经网络，检测测试数据每个样例的，判断其是否异常。
3. 基于贝叶斯网络

该算法主要在多分类异常检测中使用，它是基于概率统计的方法。贝叶斯网络进行分类预测的主要原理是：

1. 朴素贝叶斯网络能够预测提供的样例的后验概率，而这个预测的结果能够作为判断样例标签的缘由。
2. 对于测试数据，后验概率最高的标签类就是它的标签类。零概率问题是使用贝叶斯网络经常出现的问题，此问题可以结合拉普拉斯平滑解决。
3. 基于支持向量机

该算法是通过建立一种二分类模型对数据进行检测，经常使用在单分类的问题中。它的基本模型是定义在特征空间上的间隔最大的线性分类器；支持向量机还包括核技巧，这使它成为实质上的非线性分类器[5]。在检测样例中，定义一个点为异常点的方法是判断该点是否在产生的范围之内，不在则为非正常点。

1. 基于规则

基于规则的方法需要使用训练数据训练获得一些规则，如学习定义一个样例数据为异常点的规则。对于得到的规则，使用这些规则可以判断测试样例是否为异常。通常的步骤分为两步：

1. 通过训练集挖掘异常模式的频繁项集，且这些频繁项集在正常模式中不频繁出现。
2. 得到频繁项集后，测试集中符合频繁项集的样例被判断为异常数据。

基于分类的检测算法种类很多。此类算法的适用场景和优劣势也各有不同，其优势和不足如下：

（1）此类算法不仅能解决单分类问题，也适用于多分类问题，在测试建立模型时算法较快。

（2）多分类问题需要精确的多分类标签，直接将一个确定的标签赋给测试数据有时候会不恰当[6]。

#### 基于最近邻的异常检测算法

该算法的原理是检验所有的样例，获得不同样例之间的间隔大小，得到所有样例数据间距结果后，对比不同样例的间距，将远离大部分样例数据较远的点判断为异常。此类算法有以下几种：

1. 基于到K个近邻的距离

此算法的原理是找出某个样例N个最邻近的样例，获得这些样例到该样例的间距总和。比较所有的样例的间距总和，非正常的点通常是间距总和比大部分样例点间距总和都较大的点。

1. 基于相对密度

算法的原理是每次以一个样例为基准，获得这些样例的涵盖N个附近点的最小超平面，计算它们的密度。如果一个样例的密度与其周围的样例的密度差别较大，则判断此样例为异常点。

此类算法的优势和不足之处：

（1）适应于多种数据类型，只需要定义适合的距离即可。（2）算法的复杂度较大，不容易得到适合的间距。

#### 基于聚类的异常检测技术

基于聚类的算法使用设置的预定值来遴选异常对象。方法是计算每个聚类簇的大小，利用设定的小聚类簇的大小判断预定值来判定一个聚类簇的大小归类。然后获得每个聚类簇与其他聚类簇的间距，利用设定的聚类簇间距预定值判定是否舍弃一个聚类簇，将离其他聚类簇较远的小聚类簇舍弃，与其他样例非强相关的对象为非正常样例。经常使用的算法包括层次聚类、K-means以及EM算法等。

此类算法的优势和不足之处：

（1）使用聚类的挖掘算法，在样本充足的情况下准确度相对较高。

（2）训练过程中的计算量较大，原理复杂。

（3）想要得到合理的模型，需要大量的数据对模型进行训练[7]。

（4）如果异常数据单独成簇，则难以发现异常。

#### 统计异常检测技术

该算法的思想是在由一系列数据构成的随机模型中，正常点存在于高频率模块的可能性较大，非正常点存在于低频率模块的可能性较大。此种算法可以归纳为以下两类：

第一类为参数化方法，此类方法有：

1. 基于混合参数分布。
2. 基于回归模型。
3. 基于高斯模型。

第二类为非参数化方法，此类方法有：

1. 基于密度估计函数。
2. 基于核函数。
3. 基于直方图。

该类算法也有其适用的场景，以及自己的不足和优势，其特点如下：

1. 需要做出数据分布的假定，然而难以确保这个预想正确，直方图难以发现属性间的关系。
2. 若是估计分布阶段对异常足够鲁棒，可以用无监督的方式[6]。
3. 此算法还有另外一种归属于统计学的检测方法，前提是数据分布的预想正确。

#### 信息理论异常检测技术

信息理论异常检测技术的基本思想是数据中的异常会引起数据集信息内容的不规则性，通过一个数据的存在与否是否影响数据的规则性进行判断数据是否属于异常。

该技术主要的优势和不足如下：

1. 其不必像神经网络等算法要建立模型，对分布不必做出假设。
2. 效果的好坏取决于评估方法，难以得到数据的异常程度。

#### 谱异常检测技术

谱异常检测技术的基本思想是将数据嵌入到一个较低维的空间中，在此空间中正常情况和异常情况显得不同。判断异常样例的方法是寻找此空间的子空间。

谱异常检测技术的优势和不足如下：

1. 对于高维度的数据样例，可以降低数据的维度，使其在此类数据中也能使用。
2. 时间复杂度较大。

#### 上下文异常检测技术

此类算法的思想是识别样例数据的上下文，然后计算测试样例的异常得分。算法的步骤是：

1. 通过样例数据的上下文属性识别其上下文。
2. 通过已知点的异常检测算法计算每个测试样例的异常得分。

该算法的主要优势和不足有：

1. 相关技术能够生效的前提是，上下文相关的属性得到界说。
2. 得到实际异常的定义后，可利用这些定义来识别测试样例的异常。

上述中的一些算法在网络异常事件检测中已有一定的研究和应用，但由于网络异常的种类多种多样且变化十分迅速，面对不同类型的网络异常，其中的某种算法难以检测各种异常，没有令人满意的精确率和召回率，所以需要研究更加高效准确的算法用于网络异常检测系统中。

### 频繁模式挖掘算法现状

频繁模式挖掘是指通过大数据技术，寻找高频度出现在输入集中的模式，获得数据集中每一项的支持度，将不小于给定最小支持度的项目组成频繁项集[8]。利用挖掘得到的频繁项集，找出元素间的关联关系，算出每个关系的置信度，不小于预定值的关系为强关联。频繁模式挖掘算法多种多样，该挖掘算法已在文本分类、网络入侵检测技术、信息检索等领域中有所使用。

在网络异常领域中频繁模式挖掘算法研究和应用还较少，没有很好的研究成果，频繁模式挖掘算法通过挖掘数据中的频繁模式，并且这些模式在正常标签中不频繁出现，通过这些频繁模式对测试集进行预测，理论上能达到较好的效果，所以研究出频繁模式挖掘算法在网络异常检测中的应用可行度很高，具有很好的应用前途。

## 本文研究内容

基于上述的原因和意义，本文的重点工作是学习频繁模式挖掘算法的思想和实现方法，并使用这些技术设计一个网络异常事件监测工具，对网络KPI数据实行检测，辅助运维人员对应用服务进行监控。

具体的研究内容为基于频繁模式挖掘的网络异常事件检测系统，具体工作分为：

1. 数据的采集和预处理：发掘样例集中的频繁项集，然后找出样例集中的规则，要训练大量的数据，通过训练数据获得频繁项集。从竞赛官网下载的关于各种类型KPI的数据存在一定的缺失，需要对数据进行预处理补全缺失值。
2. Apriori算法的学习和设计：本研究使用经典的频繁模式挖掘算法Apriori算法。Apriori 算法是用一种称为逐层搜索的迭代方法。在进行算法实现之前，需要事先定义最小支持度与最小置信度[9]。该算法在简单易学，实现上也不存在很大的困难，而且能保证较快的速度和较好的较高的准确率。
3. Matlab集成开发环境的学习：MATLAB是一个方便、高效、功能强大的开发集成环境。其在矩阵计算、多维图像显示等方面有很好的变现，并且其具有界面系统开发的功能，使用该工具能够快速开发一个良好的检测系统。
4. 系统功能和界面的设计和实现：本设计最终需要完成一个网络异常事件检测系统，需要设计系统的功能、数据流和界面等，并实现其中的功能、处理数据的流向并将界面和后台实现连接起来，完成一个易于使用且简洁优美的系统。

## 论文结构安排

本文分为四个章节，每个章节注重介绍的内容均不同，本文的大纲以及内容简介如下：

第一章是绪论，此部分的核心为论述网络异常事件检测学习的原因以及学习的作用。本章首先讲述当前与网络异常事件监测相关的探研的状态，简述重点的算法知识和使用场景，以及每个算法特点、优势和不足，表明本研究核心的研究和工作内容，结尾阐明论文的组成、构造以及大纲。

第二章核心内容是阐述论文设计相关的算法。本章讲述的首要内容是频繁模式挖掘算法，过程中会详尽讲解本设计使用的Apriori方法，描述研究的方法和过程，阐述研究和实验的结果和结论。

第三章主要阐述设计网络异常事件检测系统的工作过程，其中包括对设计系统进行的需求分析、系统的功能设计、系统数据处理流程和系统界面的设计和后台实现，展示最终完成的系统。

第四章是对本次毕业设计和研究工作的总结和展望，叙述本次毕业设计的收获、遇到的问题、解决的方法，以及对研究内容未来的展望。

# 频繁模式挖掘算法研究

## 频繁模式挖掘算法介绍

### 概念

频繁项集是频繁模式中最重要的研究内容，是生成关联规则的重要一步[10]。

频繁模式挖掘算法有很多种类，以及很多与其他算法相结合所得的算法变种。此中3种经典的频繁模式挖掘算法分别是Eclat算法、Apriori算法和FP-Growth算法。下面将具体讲解这三种算法的原理知识和实现过程，并对它们的优势、不足和适用场景进行分析。

### Apriori算法

Apriori算法是对现在影响较大的数据挖掘算法，利用迭代形式，依次循环处理，直到生成最终的频繁K项集，整个过程需要多次扫描数据库，最后根据频繁K项集分析总结出强关联规则，供用户决策分析[11]。

Apriori算法执行的2个过程：

1. 频繁项集的产生：第一步中的两个核心方法是连接和剪枝。遍历输入集中的所有样例，获得所有的1-项集（C1），接着算出它们的support，舍弃不大于预定值的C1，删除不符合规则的项集的步骤叫做剪枝步。然后1-项集自身连接得到2-项集，此步骤成为连接步。反复进行上述两步，当没有新的频繁项集产生，则算法结束[12]。
2. 关联规则的产生：第一步获得频繁项集后，利用这些频繁项集即可找出关联规则。获得这些规则的置信度，找出满足频繁项集和置信度阈值的规则，我们把这些规则称为强规则[13]。

### FP-Growth算法

FP-Growth算法主要通过FP-Tree来构造频繁集[14]。FP-Tree包含频繁项头表和项前缀树，能加快算法的速度。

FP-Growth有3个运行过程：

1. 组建一棵FP树。FP树按照支持度递减的规则排序，在构造FP树时，算法共2次读取遍历数据文件，第一次遍历的效用是计算发生的频度，第二次遍历不考虑非高频度的对象。由于FP树的特点，其节点构成的频繁项集能够更多地一起使用相同的首部。
2. 找出（1）中得到的tree中的频繁项集。（1）中得到tree后，即可利用这些tree找出频繁项集，方法是：（1）通过tree得到数据中的条件模式基。（2）通过（1）的结果，生成一棵条件tree。
3. 重复执行以上两个步骤，到树只囊括单个对象时终止。

### Eclat算法

Eclat算法的原理是：任何一个非1-项集S-项集，可以由2个S-1-项集进行或计算产生，而1-项集在读取数据文件时获得。

依据上述理论，Eclat算法只需要遍历一次样例集文件，利用这次遍历获得所有的1-项集，利用这次遍历所得的1-项集能够迭代产生其他数量的项集，不需要第二次遍历样例文件。实现该算法有两个阶段：

1. 第1个步骤为输入，此步骤的作用是遍历一次样例文件，把所有的样例存储在一个类中，该类记录了项中存在的元素，还有项的支持度。
2. 第2个步骤为连接，此步骤使用上一个步骤产生的项集连接得到下一个维度的项集。

### 三种算法对比

1. 3种算法中，Apriori算法为了计算项集的支持度和置信度，算法必须数次遍历样例文件，遍历的过程花费很多的性能和时间，所以此算法最慢。
2. 在3种算法中，FP-Growth的速度会随着元素的变多而变慢。由于元素变多会导致tree的高度变大，因此产生的子问题就也越多，所以会导致其速度变慢。
3. 在3中算法中，Eclat算法的速度是最快的，但是它不适用于样例太多的场景。因为此算法是通过递归实现的，如果样例太多，会导致存储递归方法出口和方法变量的栈帧数量膨胀，从而导致内存膨胀，使得算法速度变慢。

## Apriori算法

上一个小节中已经介绍了频繁模式挖掘算法的基本概念和其三种经典算法，基于本文的研究方向为使用Apriori的算法进行网络异常模式挖掘，本小节将讲解与Apriori算法相关的概念和算法原理。

### 基本概念

为了更好的学习这个方法，必须先学习一些相关的知识，例如支持度等一些知识。具体知识讲解如下：

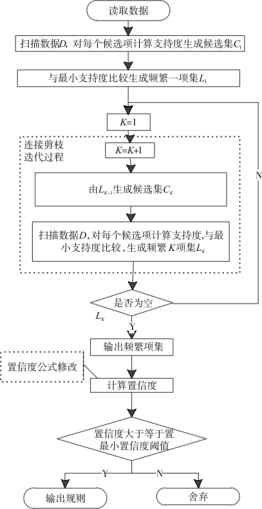
1. 支持度：两个事件S和T，关于S->T，这个规则的支持度意指事件S发生时，事件T也发生的几率。计算方法是S和T同时发生的次数除以总得样例的次数。
2. 置信度：事件S和T，置信度(S|T)的意指事件S发生时，事件T也发生的几率。计算的方法是事件S和T同时发生的次数除以事件S发生的总的次数。
3. 项集：项集是指样例中对象与对象的组合，将含有S个对象的样例成为S-项集。
4. 关联规则：关联规则是映射某个元素与其他元素内部的联系关系的一种方法。对大量的样例进行发掘，可以得到难以察觉的元素与元素内在的联系。

### 算法步骤

以下为其操作的过程，主要分为连接和剪枝。具体如下：

1. 遍历样例文件，由此得到所有的1-项集，算出所有1-项集的支持度，其中有一部分项集的支持度不大于预定值，去掉这部分项集，剩下的即是频繁1-项集。
2. 利用上一步产生的N-项集，运算后能够获得N+1-项集。方法是任何两个N-项集相互组合，生成N+1项集，去除重复的项集，根据最小支持度大小，去掉频次小于最小支持度的项集[9]，剩下的即是频繁N+1项集。
3. 反复进行（2），当不再出现未出现过的项集时终止。
4. 获得关联规则：使用上一步找到频集产生期望的规则[15]。取以上获得的所有频繁项集的非空子集，这些子集与其补集形成一种关联关系，如频繁项集{S,T}有S->T和T->S两种关联关系，算出每种规则的置信度，将置信度高的改造成为强关联。

图一 Apriori算法流程图



### 算法优缺点

Apriori方法的优势有：基本思想简单，实现容易。

该算法的劣势为：耗费的时间和性能较大。获得置信度和支持度必须遍历所有的样例数据，而且遍历的次数较多，时间复杂度较大。这些缺点导致Apriori算法的执行效率极低[11]。

## 研究方法

### 数据集下载及预处理

本毕业设计最后完成一个软件，上述的算法是软件的核心实现算法，该软件的功能是检测网络异常数据。系统核心算法的编写测试以及系统的功能测试都需要大量的数据进行训练和测试。本研究数据均从网上下载，数据来源于多个互联网公司，经过脱敏处理后供研究者使用。

由于数据均为实时记录的线上数据，因为数据存在一些缺失值需要进行预处理方能使用，本研究采用matlab的插值函数进行数据补全，其补全数据的方法是利用缺值点前后的点所连接成的曲线来估计缺失的值。

### Apriori算法设计与实现

其中数据训练集每个样例包含四个属性， timestamp、value、label和KPI ID，其中需要挖掘出KPI ID、value和label之间的关联关系，即找寻异常标签的频繁模式，且这些标签在正常模式中不频繁出现。

详细的算法过程为：

1. 读入训练集文件的数据，将数据存储在系统的工作空间中。
2. 遍历所有的样例，识别所有样例的所属的KPI ID，并得到数据中包含的所有KPI ID，按照ID对数据进行分组。
3. 用户选择需要检测的KPI后，获得该组KPI所有的样例数据进行训练。先得到该组KPI所有的异常点数据，然后在正常数据中找出与异常点数据相近的数据，计算每种异常数据在正常数据中占有的比例，删除比例较高的异常点，剩下的就是频繁出现的异常模式。
4. 利用上一步产生的异常模式，遍历需要检测的数据。如果某个或者某段数据模式与异常模式匹配，则判断其为异常点。
5. 计算检测的结果。

## 实验结果及结论

通过算法检测得到的每种KPI的数据结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| KPI ID | 精确度 | 召回率 | 得分 |
| 02e99bd4f6cfb33f |  |  |  |
| 046ec29ddf80d62e |  |  |  |
| 07927a9a18fa19ae |  |  |  |
| 09513ae3e75778a3 |  |  |  |
| 18fbb1d5a5dc099d |  |  |  |
| 1c35dbf57f55f5e4 |  |  |  |
| 40e25005ff8992bd |  |  |  |
| 54e8a140f6237526 |  |  |  |
| 71595dd7171f4540 |  |  |  |
| 769894baefea4e9e |  |  |  |
| 76f4550c43334374 |  |  |  |
| 7c189dd36f048a6c |  |  |  |
| 88cf3a776ba00e7c |  |  |  |
| 8a20c229e9860d0c |  |  |  |
| 8bef9af9a922e0b3 |  |  |  |
| 8c892e5525f3e491 |  |  |  |
| 9bd90500bfd11edb |  |  |  |
| 9ee5879409dccef9 |  |  |  |
| a40b1df87e3f1c87 |  |  |  |
| a5bf5d65261d859a |  |  |  |
| affb01ca2b4f0b45 |  |  |  |
| b3b2e6d1a791d63a |  |  |  |
| c58bfcbacb2822d1 |  |  |  |
| cff6d3c01e6a6bfa |  |  |  |
| da403e4e3f87c9e0 |  |  |  |
| e0770391decc44ce |  |  |  |

表 1 检测结果

如上表所示，使用此算法能检测出大部分类型的KPI，并且得分均在60以上，但是仍有少部分类型的KPI数据无法检测出结果，或者检测结得分很低。由此得出，Apriori算法能够挖掘出大部分类型的网络异常数据，但是在某些类型的异常数据则难以检测其异常。所以，如若需要使用的生产环境中，必须解决算法目前的缺陷，只有与其他方法相结合，才能解决目前算法的不足。

## 本章小结

本章阐述了与设计相关的数学算法和软件实现过程。开头是阐述算法的原理、相关知识、不同种类的实现算法的简介、适用场景和优劣势的表明。而后阐述了软件设计及实现的过程，包括数据的下载和预处理、算法的完成过程。最后阐述算法的运行的结果，总结实验的过程和收获，给出结论。

# 网络异常事件检测系统的设计和实现

## 需求分析

本研究为设计一个基于频繁模式挖掘的网络异常事件检测系统，通过分析和调研，发现系统最好可以数据可视化、结果可视化和选择不同类型的KPI进行检测等功能，按照系统的定位和调研结果，得出了以下的需求分析。

首先用户可以通过系统选择需要检测的KPI数据对应的训练集和测试集。为满足用户的需求，本系统对KPI的检测不限定于某些类型，而是可以选择多种不同类型的KPI进行检测。

选择了数据文件后，系统会读取文件的内容，并识别出其中包含的KPI种类，以列表的形式将所有的KPI ID展示给用户，用户可以通过绘图功能选择指定的KPI，将其以时间戳为横坐标，value为纵坐标绘出曲线图，并标出其中的异常点，并计算该KPI中异常所占的比例，以饼图形式展示。此功能可以让用户更加直观的查看KPI的模式，以及查看异常发生的情况和比例。

最后还有一个十分关键的功能，即对不同种类的KPI实行测试。用户可以通过检测功能对读取的测试数据的KPI进行检测，用户可以选择不同类型的KPI进行检测。系统检测完成后，同样会将测试数据的KPI曲线图绘画出来，同时展示原始异常和算法预测异常的对比，计算系统预测的精确度、召回率和得分，以可视化的形式展示在用户面前。

## 系统功能设计

系统功能图如下所示：

用户界面

功能接口

读文件功能

绘图功能

检测功能

导出文件功能

文件读取服务

数据解析服务

数据存储服务

绘图服务

异常计算服务

异常模式挖掘服务

测试集检测服务

文件导出服务

计算得分服务

图2 系统功能设计图

### 读取文件

用户点击文件按钮即可选择测试集和训练集，待系统读取文件完毕后，系统将用户选择的文件显示在界面。

### 绘画KPI曲线图

系统读取用户数据后，会从训练集中找出KPI ID的种类，并将ID的列表展示在界面，用户可以选择想要查看的KPI，点击绘图功能，即可将该KPI的曲线图绘制在界面上，并且会标出其中的异常点，同时以饼图的方式展示异常所占的比例。

### 异常频繁模式挖掘

选择文件之后，用户即可使用系统对训练集训练，只需选择训练功能，系统会自动开始挖掘异常标签中的频繁模式。

### 测试及评估算法

进过训练数据后，得到异常标签的频繁模式，用户即可使用挖掘的模式对测试集进行测试，系统测试完毕后，同样会将测试的异常点以曲线图的形式展示给用户，同时展示原始的数据异常图便于用户对比。并算出测试成果的精度、召回率和得分。

### 导出测试集

对测试集检测结束后，系统展示检测的结果和得分，用户可将测试的结果导出到文件中，以便后续观结果时可以直接观察文件。

## 数据流图

用户导入的数据流转图如下所示：

用户

训练集和测试集

KPI ID

KPI数据文件

KPI ID

KPI数据

数据图形

KPI ID

KPI数据

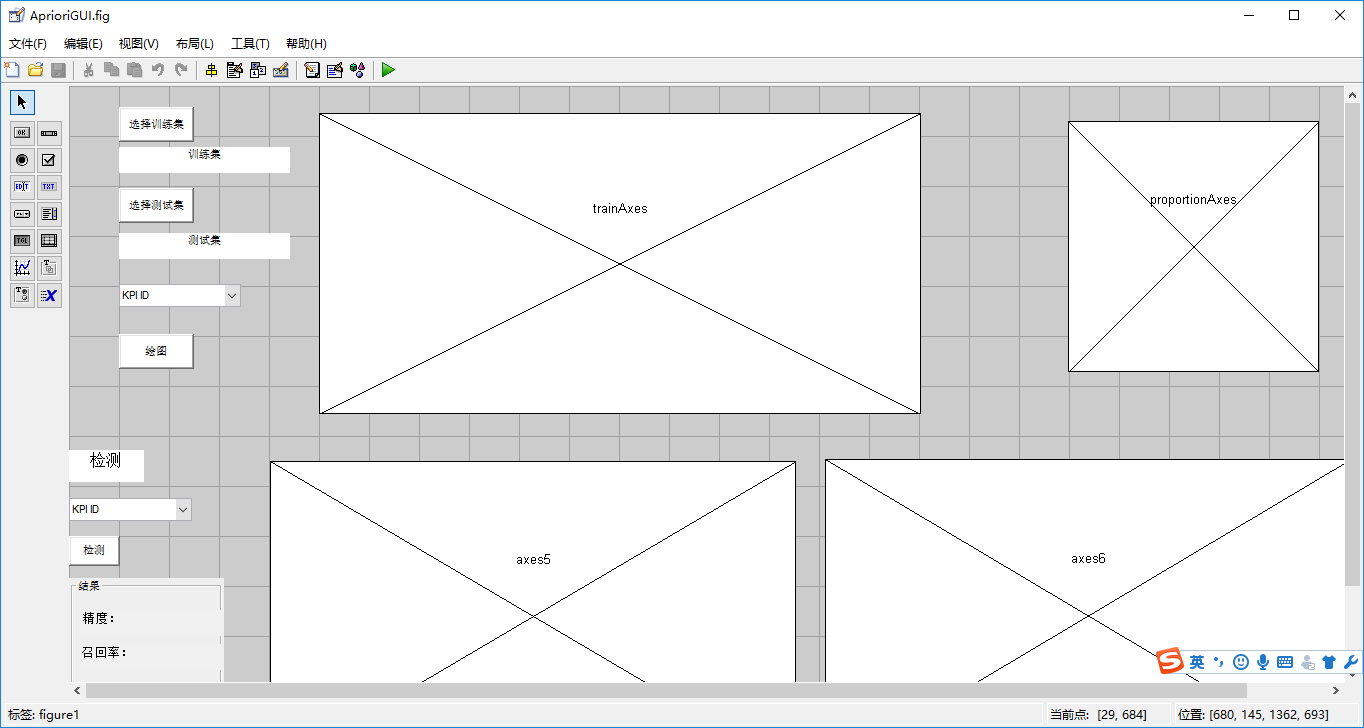
检测结果

图3 系统数据流图

## 系统功能实现

### 系统界面设计

按照系统的需求分析和功能设计，系统界面以简洁实用为原则，在保证所有需求实现的同时，给用户带来良好的实用体验。

图 4 系统原始界面

上图为系统的外观界面图。该界面分为三个模块，分别为顶端的菜单栏、中部的文件选择和绘图模块以及底部的KPI检测模块。

菜单栏为用户展示的的操作包括文件、编辑、工具和帮助四个操作。每个操作的功能如下：

1. 文件菜单提供了打开文件、保存文件和退出程序的功能；
2. 编辑菜单提供了清除数据、选择曲线图颜色和缩放图片的功能；
3. 工具菜单栏提供了图片下载、测试结果导出功能；
4. 帮助菜单栏提供了软件介绍和使用帮助的功能。

中部文件选择和绘图模块主要的功能为：

1. 可用通过菜单栏或者文件选择按钮选择需要读取的训练集和测试集；
2. 文件选择完毕后，系统会展示训练集和测试集的文件名以便用户查看；用户选选择文件结束后，软件会读入文件中的数字和文本，并将这些数字和文本存储到软件的工作空间中，为其他功能提供数据。

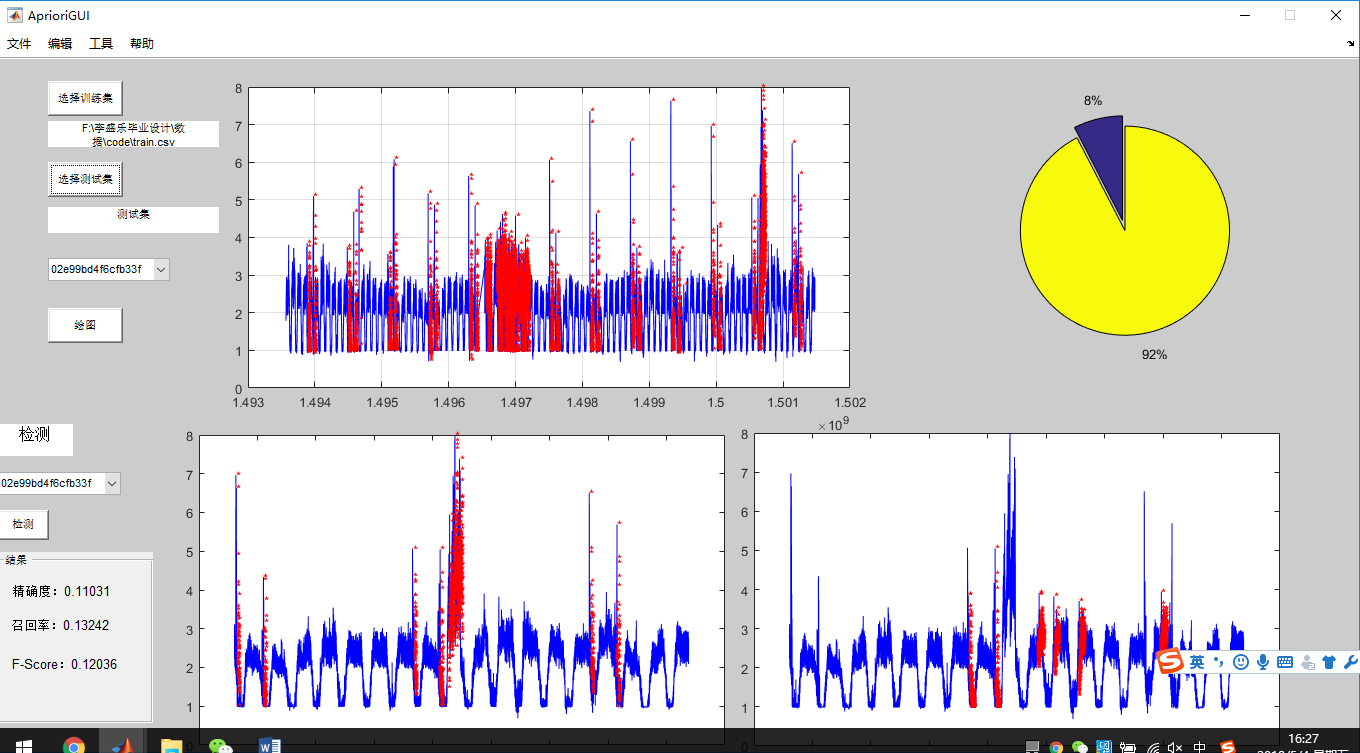
### 系统后台实现

上述章节已经对系统的需求分析、功能设计和界面设计进行了阐述，接下来将分别讲述各个功能的后台实现方式。

1. 读取文件：读取文件的过程是用户点击文件选择按钮，然后弹出一个文件选择对话框，用户利用对话框找到并选择需要读入的文件，此处使用的库方法是uigetfile。获得用户选择的文件名后，即可使用textscan()函数读取文件，并将文件数据存储到矩阵中。
2. KPI种类解析：读取文件完毕后，对数据进行解析，将其中的数据和文本分别存储，数据主要是测试集和训练集的KPI 数据，文本则是KPI ID。将KPI ID种类解析后展示在用户面前，解析方法是遍历所有的ID，然后取出重复ID，将ID的种类解析出来。
3. KPI绘图：用户选择某个KPI后，系统查找该KPI所有的数据，然后通过matlab的plot()函数将这些数据绘出曲线图展示给用户，并计算其中的异常比例，使用pie()函数绘出饼图。
4. 异常模式挖掘：通过前面阐述的2.3.2章节中Apriori算法的设计与实现中的方法，挖掘KPI数据中的异常标签的频繁模式。
5. 测试数据异常检测：用户选择需要检测的KPI ID后，通过（4）中挖掘出来的频繁模式判断测试样例是否异常。
6. 召回率和精确率的计算：精确率=TP / (TP + FP)，召回率=TP / (TP + FN)。其中符号的含义是：TP表示被成功识别的异常点；FP表示被错误识别为异常的正常点；FN表示未被正确识别为异常点的异常点。

## 设计结果

本设计的结果为成功设计一个基于频繁模式挖掘的网络异常事件检测系统，其中对系统的需求分析的功能全部实现，系统的界面如下：

图 5 系统运行时界面

## 本章小结

本章主要介绍了网络异常事件检测系统的设计和实现过程。

首先介绍了软件的需求分析，从用户的角度和软件功能的角度对软件的需求进行了说明，叙述了软件功能和界面设计的理念和思想。

紧接着介绍系统的功能设计，包括系统的主要功能和功能的作用，帮助读者更好的熟悉系统的功能要点。

然后给出了系统的数据流图，描述系统数据的流动方向，帮助读者更好的理解系统处理数据的流程。

接着叙述了软件的实现历程，囊括软件外观的制定和软件后端实现的重要技术和历程，为用户现实界面的重要模块，以及界面与后端共同工作的的方式，让读者了解软件的全部设计心路历程和重要模块。

最后为读者呈现系统的最终形态，即本设计工作最终的结果。

# 总结与展望

## 总结

在互联网时代背景下，确保一个网络服务的平稳性、能动性和高效性十分的重要，仅靠人力的监控和检查是很难远远不够的，如今数据挖掘和分析技术发展的十分迅猛，已在很多领域得到使用，所以结合数据挖掘技术设计更加高效的网络异常检测算法很有必要。本研究的首要任务包罗如下两个方面：

1. 频繁模式挖掘算法

频繁模式挖掘算法是本设计研究学习的核心算法，主要内容是探究频繁模式挖掘算法在网络异常事件检测中的应用。通过挖掘异常标签的频繁模式，用这些频繁模式对数据进行检测。本论文拟采取Apriori算法，进行实验得出结论该算法能挖掘出大部分类型的KPI的异常频繁模式，并能对测试数据进行检验。

1. 网络异常检测系统设计

本文的另一个重要的模块为制定和完成一个网络异常检测系统。文章中对软件的需求阐明、重要功能、界面模块和后台实现历程进行了叙述，将制定和实现的全部历程归纳综合的显现在读者面前。

## 展望

本毕业设计结合频繁模式挖掘算法最终设计了一个功能完善的网络异常事件检测系统，为更多的研究者提供了参考。但是由于能力和时间的局限性，系统的精确度还待提高，其他学者可结合其他算法对Apriori算法进行优化，或者研究更加高效优秀的算法用于异常检测中。

# 参考文献

1. 宋歌, 闫巧, 喻建平. 神经网络在异常检测中的应用[J]. 计算机工程与应用, 2002, 38(18):146-148.
2. 郭通. 基于自适应流抽样测量的网络异常检测技术研究[D]. 中国人民解放军信息工程大学, 解放军信息工程大学, 2013.
3. 钱叶魁, 陈鸣, 叶立新,等. 基于多尺度主成分分析的全网络异常检测方法[J]. 软件学报, 2012, 23(2):361-377.
4. 赵京晶. 三种神经网络对上证指数预测的研究[J]. 商, 2013(17):190-191.
5. 张妤. 支持向量机集成学习方法研究[D]. 山西大学, 2008.
6. 胡元洪. 网络流量异常检测算法[D]. 西安电子科技大学, 2010.
7. 柴洪峰, 李锐, 王兴建,等. 基于数据挖掘的异常交易检测方法[J]. 计算机应用与软件, 2013, 30(1):165-170.
8. 刘双跃, 彭丽. 基于Apriori改进算法的煤矿隐患关联性分析[J]. 内蒙古煤炭经济, 2013(11):149-151.
9. 董阳光. 基于Apriori算法的关联规则超市购物推荐算法研究[J]. 中国战略新兴产业, 2017(48).
10. 马青霞, 李广水, 孙梅. 频繁模式挖掘进展及典型应用[J]. 计算机工程与应用, 2011, 47(15):138-144.
11. 刘丽娟. 改进的Apriori算法的研究及应用[J]. 计算机工程与设计, 2017(12):3324-3328.
12. 王娟. 基于关联规则的数据挖掘算法研究[J]. 科技信息, 2011(33):56-56.
13. 邹金花. 关联规则在图书馆中的应用[J]. 现代情报, 2013, 33(5):17-20.
14. 田庆, 朱俊岭, 刘永梅. FP-Growth算法在电子商务中的应用[J]. 科技与企业, 2014(14):148-149.
15. 姚旭升, 杨静, 谢颖夫,等. 关联规则算法在临床医疗诊断中的应用[J]. 软件导刊, 2018(3).