单位代码



学 号

分 类 号



毕业设计(论文)

**船舶数据获取和处理系统的设计与实现**

|  |  |
| --- | --- |
| 院（系）名称 | 软件学院 |
| 专业名称 | 软件工程 |
| 学生姓名 | 邓经纬 |
| 指导教师 | 马平 |

2018年 月

北京航空航天大学

**本科生毕业设计（论文）任务书**

Ⅰ、毕业设计（论文）题目：

船舶数据获取和处理系统的设计与实现

Ⅱ、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

原始数据主要来源于两部分。第一部分为船舶机舱综合检测报警系统（AMS）；第二部分是船舶自动识别系统（AIS）。

设计技术要求主要使用了数据库管理技术，分布式文件管理技术，数据挖掘技术

Ⅲ、毕业设计（论文）工作内容：

1.研究数据来源，设计并实现可靠的数据库，提供外部接口

2.对数据库中的数据进行挖掘，得出有价值的信息和结论

3.将成果可视化展现给用户

Ⅳ、主要参考资料：

软 件 学院 软件工程 专业 班

学生 邓经纬

毕业设计（论文）时间： 年 月 日至 年 月 日

答辩时间： 年 月 日

成 绩：

指导教师：

兼职教师或答疑教师（并指出所负责部分）：

系（教研室） 主任（签字）：

注：任务书应该附在已完成的毕业设计（论文）的首页。

**本人声明**

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：

签字：

时间： 2018 年 月 日

**船舶数据获取和处理系统的设计与实现**

作 者：邓经纬

指导教师：马平

摘 要

本课题综合获取了船舶航行状态数据，模拟的船舶设备数据，完成了对船舶数据的管理。在稳定安全存储的基础上，对各种数据进行数据挖掘，获得了各种有价值的信息。最终通过Web应用的形式将挖掘成果可视化展现给用户。

在大数据的背景下，为了完成目标，首先使用了Hadoop分布式存储和Mongodb数据库存储数据；对于船舶的设备数据主要使用将MySQL数据库导入Mongodb数据库，而船舶运行状态数据则使用爬虫技术获得并存入数据库；在数据挖掘方面，则使用Spark计算框架库编写代码计算获得价值信息，最终根据实际需求确定存入数据库或是直接展示给用户；可视化问则使用Node.JS、Express框架搭建网页、使用高德地图API在地图上展现各种位置和挖掘成果，主要包括船舶分布信息、单艘船舶航行热点、单艘船舶设备数据分析结果、速度分析结果等；使用ECHART框架构建数据图表，显示数据和挖掘结果。最终形式为用户通过网页浏览信息，以搜索的方式查看船舶数据挖掘成果。

最终本系统完成了上述的目标，最终拥有了稳定安全的数据库，并且系统接口稳定，响应较快，用户体验良好，并展示了各项有价值的信息。

**关键词：**Web应用，大数据，数据库，数据挖掘，可视化

**Design and Implementation of Ship Data Acquisition and Processing System**

Author : DENG Jingwei

Tutor : MA Ping

Abstract

This project has comprehensively acquired ship navigation status data, simulated ship equipment data, and completed management of ship data. On the basis of stable and secure storage, data mining of various data has obtained various valuable information. Finally, the mining results are visualized to the user through the form of web application.

In order to accomplish the objectives, the Hadoop distributed storage and the Mongodb database are first used to store data; for the ship’s device data, the MySQL database is mainly imported into the Mongodb database, and the ship running state data is obtained by using the crawler technology and stored in the database; In terms of mining, the Spark computing framework library is used to write code calculations to obtain value information. Finally, it is determined to be stored in the database or directly displayed to the user according to actual needs. Visualization uses Node.JS and Express framework to build web pages and use the high-tech map API. The map shows various locations and excavation results, including ship distribution information, single ship navigation hotspot, single ship equipment data analysis results, speed analysis results, etc.; using ECHART framework to build data charts, display data and mining results. The final form is that the user browses the information through the webpage and searches the ship data mining results in a search manner.

In the end, the system has achieved the above objectives, and finally has a stable and secure database, and the system interface is stable, the response is faster, the user experience is good, and various valuable information is displayed.

**Keywords:** web application, database, big data, data mining, visualization

**目 录**

[**1** **绪论** 1](#_Toc5370625)

[1.1 课题背景与意义 1](#_Toc5370626)

[1.2 国内外研究现状 1](#_Toc5370627)

[1.3 研究目标 2](#_Toc5370628)

[1.4 论文组织结构 2](#_Toc5370629)

[**2** **需求分析** 4](#_Toc5370630)

[2.1 需求分析 4](#_Toc5370631)

[2.2 数据来源分析 4](#_Toc5370632)

[2.3 用例图和用例设计 5](#_Toc5370633)

[2.4 系统业务流程 6](#_Toc5370634)

[**3** **技术简介** 9](#_Toc5370635)

[3.1 MongoDB 9](#_Toc5370636)

[3.2 Hadoop分布式框架 9](#_Toc5370637)

[3.3 Spark计算引擎和Scala编程语言 10](#_Toc5370638)

[3.4 Node.js开发平台和Express框架 11](#_Toc5370639)

[3.5 高德地图API 11](#_Toc5370640)

[3.6 Echart图表框架 12](#_Toc5370641)

[**4** **总体设计** 13](#_Toc5370642)

[4.1 系统总体设计 13](#_Toc5370643)

[4.2 系统体系结构设计 14](#_Toc5370644)

[**5** **详细设计与实现** 15](#_Toc5370645)

[5.1 数据库层模块设计与实现 15](#_Toc5370646)

[5.1.1 船舶数据库详细设计 15](#_Toc5370647)

[5.1.2 Hadoop环境搭建 16](#_Toc5370648)

[5.1.3 Mongodb与Hadoop的连接配置 17](#_Toc5370649)

[5.1.4 数据备份的设计与实现 18](#_Toc5370650)

[5.1.5 爬虫的设计与实现 18](#_Toc5370651)

[5.1.6 设备数据引入的设计与实现 19](#_Toc5370652)

[5.2 数据挖掘层设计与实现 20](#_Toc5370653)

[5.2.1 Spark框架的配置和使用 20](#_Toc5370654)

[5.3 数据展示层的设计与实现 20](#_Toc5370655)

[4.3.1 直接数据展示的设计与实现 20](#_Toc5370656)

[4.3.2 数据挖掘结果可视化的设计与实现 21](#_Toc5370657)

[5.4 系统的整合和搜索功能的实现 22](#_Toc5370658)

[5.5 系统服务器配置 22](#_Toc5370659)

[**总结与展望** 24](#_Toc5370660)

[**论文总结** 24](#_Toc5370661)

[**展望** 24](#_Toc5370662)

[**参考文献** 26](#_Toc5370663)

[**致谢** 27](#_Toc5370664)

# 

# **绪论**

## 课题背景与意义

随着信息化应用水平的进一步深入和各个信息子系统多年的应用，船只和船企都有大量历史数据的积累。而大数据技术为解决此问题提供了很好的解决思路和方案。除此以外，大数据还可以帮助航运公司做很多优化，进行良好的资产配置， 将航运大数据通过数据融合、数据分析转化为具有价值的信息，并以数据产品的形式提供给行业。

大数据和“互联网+”的兴起使得“打破行业壁垒，压缩中间环节，提高效率”成为可能。这既是航运业的痛点，也是航运业的机会。在大数据时代背景下，船舶智能化已经成为当今船舶制造与航运领域发展的必然趋势。船舶本身基本都设有船舶机舱综合监测报警系统（AMS）和船舶自动识别系统（AIS）等系统，能够将这些设备和系统采集到的数据进行传输，整理和存储，就是加强船舶管理，提高行业效率，实现船舶智能化的基础。

该项目需要将数据采集系统所收集到的数据进行进一步处理，存储，分析；从而为船东，船务公司管理人员提供关于船体和航行的各种更为直接的实际数据和航行情况，例如船舶设备信息，油耗信息，位置信息，动向与轨迹等。在出现紧急情况时，也可以最先提供救援支持。这也是本项目相对直接的作用和意义。

## 国内外研究现状

大数据处理的关键技术主要包括：数据采集、数据预处理（数据清理、数据集成、数据变换）、大数据存储、数据分析和挖掘、数据的呈现与应用（数据可视化，数据安全与隐私等）[1]。其中数据的集成需要将数量庞大的数据集成到集中的大型分布式数据库（例如Hadoop。同时也是MapReduce应用的经典实例。在实现原理上采用主从控制模式，主节点存储元数据，接受应用请求并且做出相应应答，而从节点负责存储数据）；而对于重要的数据分析技术也有很大突破，对结构化和半结构化数据通过各种有效算法可以深度分析，对非结构化数据也能够转化为机器可识别的信息从而提取有用信息。在互联网中，还可利用可视化技术将复杂庞大的技术清晰的展现在人们眼前。

在船舶领域，大数据也得到了广泛的应用。随着船舶智能化发展，船舶各种类型的数据急剧增多。很多企业和政府部门都已经利用大数据开始对船舶的数据进行获取和分析。其中国内就有上海意码软件公司利用北斗卫星，对船舶数据进行获取和显示；船讯网和中国港口网则是基于已经相对成熟的AIS系统来获取数据并展示给用户。这是由岸基（基站）设施和船载设备共同组成，是一种新型的集网络技术、现代通讯技术、计算机技术、电子信息显示技术为一体的数字助航系统和设备，对船舶安全有很大帮助，也为船舶和海事部门甚至社会提供服务。

## 研究目标

本课题的目标是通过设计并实现一个船舶智能数据处理系统来实现船舶数据的存储、处理和分析。为查看信息的用户提供可视接口。也在一定程度上为之后可能的分析提供数据支持。

本课题旨在解决一下几个关键问题：

1. 将已经收集到的大量数据进行有效，安全的存储。
2. 对数据进行进一步优化处理，挖掘。
3. 对挖掘成果和数据的可视化展现。

## 论文组织结构

本论文在第一章绪论部分介绍分析了课题背景和研究现状，给出了系统目标。

第二章主要对需要完成的系统进行需求分析，确定系统用户对象以及用例。并且对所有用户和用户进行分析之后，明确系统数据来源，画出各类用户对本系统的操作流程，为接下来的技术选择和进一步设计提供基础。

第三章对本系统所用技术进行简单介绍和比较，为之后的设计提供设计思路、概念模型和技术支持。

第四章对系统进行总体设计并给出系统功能模块图和系统体系结构图。

第五章根据系统功能模块图和体系结构度，进行系统的详细设计。按序给出了系统的数据库层，数据挖掘层和数据展示层的设计思路和详细设计。

之后对本系统和论文进行分析和总结。给出了本系统的成果和优点，在展望中分析了系统存在的问题并给出了一定的解决方案。

最后列出所有参考文献，并对所有帮助本系统的完成的人员表达了深刻的感谢。

# **需求分析**

## 需求分析

针对当前集成大量船舶数据的平台进行调研发现，一些盈利平台主要的问题是面向船企或者船东，以相对精确的数据进行私人定制的服务来获取一定的利润，但在数据挖掘方向和可视化成果不足。数据量太大，功能接口太多反而使得地图显示比较杂乱，页面嵌入太多广告造成了用户体验不佳。

而有些平台虽然免费提供信息，但是存在数据有限，无法集成船舶的设备数据，信息价值有限，信息准确度不高的问题；而且也存在网页不够人性化，技术较为老旧的问题。

正是为了解决这些问题，基于已经存在的船舶数据管理平台的基础和相应数据支持，针对上述存在的一些问题和一些实际需求，才有了这一项目的最终需求。

## 数据来源分析

船舶数据来源大致有2类，来源方式如图2.1所示。

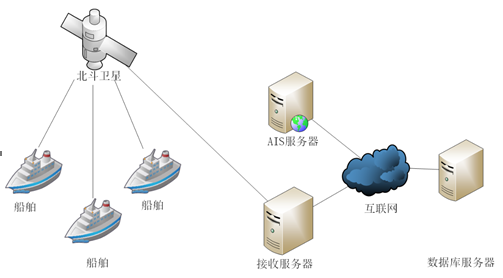


图2.1船舶数据来源方式图

（1）船舶机舱检测报警系统(AMS) [2]：主要是利用了可编程序控制器（PLC）系统可以获得包括主机、辅机及其他动力装置的各种运行参数，包括温度、压力、液位、流量、转速等数据。主要是设备的状态信息以及耗费的资源信息。实际通过另一同学的数据采集子系统获得。

（2）船舶自动识别系统（AIS）[3]：船舶静态数据，包含船名、呼号、MMSI(水上移动通信业务标识码)、IMO（国际海事组织的识别码）、船舶类型、船长、船宽等；船舶动态数据，包含经度、纬度、船首向、航迹向、航速等；船舶航程数据，包含船舶状态，吃水，目的地等。这些信息均可以利用爬虫技术获取。

## 用例图和用例设计

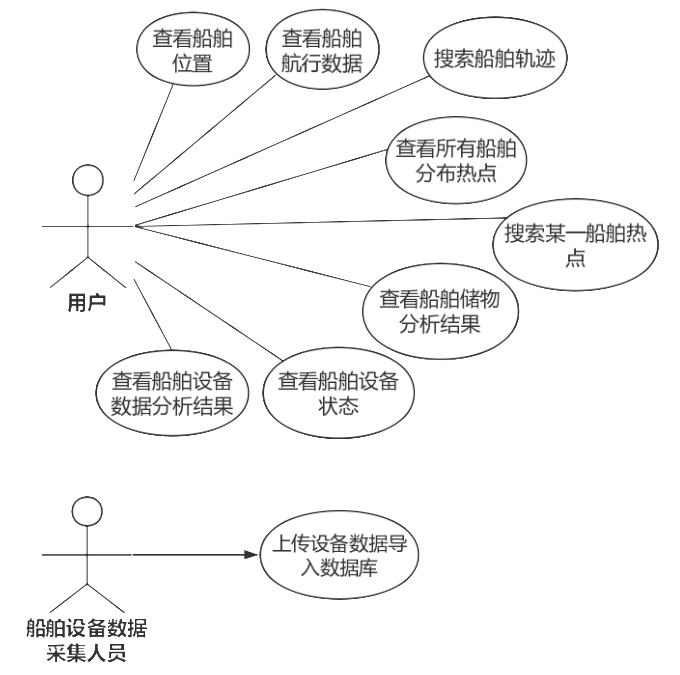


图2.2系统用例图

对于系统的用户，需要能够通过浏览器查看所有船舶的位置，查看船舶信息，查看所有船舶的分布热点；搜索某船舶的轨迹，搜索某船舶的航行热点，搜索某船舶速度分析结果；查看船舶设备数据及分析结果，查看船舶储物量数据和分析结果。这样就可以分析得出如图3.2的系统用户用例图。

对于船舶设备数据主要来源于船舶设备数据采集人员，需要能够将一定格式的数据文件上传导入数据库进行存储。

针对图2.1，进一步设计得出系统用例设计简表，如表3.1所示。

**表2.1系统用例设计简表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用户输入和动作 | 系统响应 | 输出 |
| 1.1 | 查看船舶位置 | 进入网站首页 | 数据库查询所有船舶信息 | 在网站海图上显示所有船舶位置 |
| 1.2 | 选择船舶分布模式 | 点击选择船舶分布模式 | 网页根据选择转换模式 | 以聚集图、海量图、天气海图、热力图等形式展示传播位置 |
| 1.3 | 查看船舶详细信息 | 点击船舶标记 | 根据用户选择展现船舶详细信息 | 在海图上弹出显示详细信息的窗体信息 |
| 2.1 | 搜索船舶 | 输入船舶MMSI或者中文名，点击搜索按钮 | 系统后端查询相应船舶最近两个星期历史纪录 | 跳转进入船舶航迹显示界面并动态展示航迹，并给出数据分析的跳转按钮 |
| 2.2 | 查看船舶航行热力图 | 点击显示该船舶  的航行热点 | 后端查询相应船舶的所有历史纪录 | 跳转进入船舶航迹热点显示界面并以热力图的形式展示 |
| 2.3 | 查看船舶速度分析结果 | 点击显示该船舶的速度分析 | 后端查询相应船舶最近两周的历史纪录 | 跳转进入速度分析界面，以折线图的形式展示，并给出相应结果 |
| 3.1 | 查看船舶设备数据分析结果 | 点击进入船舶设备数据分析结果界面 | 后端查询该船舶旗下船舶最近两周的船舶设备历史纪录并获取数据库中分析文字 | 跳转到船舶设备分析界面，生成折线图、柱状图等分析结果，并展现文字结果 |
| 3.2 | 查看船舶储物数据分析结果 | 点击进入船舶储物数据分析结果界面 | 后端查询该船舶旗下船舶最近两周的船舶储物历史纪录并获取数据库中分析文字 | 跳转到船舶设备分析界面，生成折线图、柱状图等分析结果，并展现文字结果 |
| 4.1 | 上传设备数据 | 数据采集人员点击进入数据上传界面 | 系统返回数据上传界面并等待数据采集人员上传后确定 | 数据采集人员能选择多个CSV格式数据文件并点击上传按钮，系统将数据存入数据库 |

## 系统业务流程

船舶设备数据采集人员业务流程图如图2.3所示，主要描述用户的业务流程。用户业务流程图如图2.3所示，主要描述船舶数据采集人员的业务流程。其中对于选择的文件，必须是数据按一定格式排序的CSV文件，且不含标题。具体数据内容见4.1.1船舶数据库详细设计；具体数据顺序略有不同，详见4.1.5设备数据引入的设计与实现。

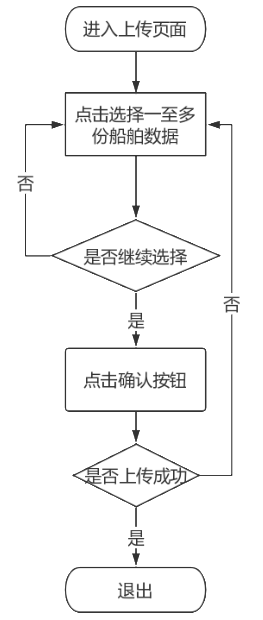


图2.2船舶设备数据采集人员业务流程图

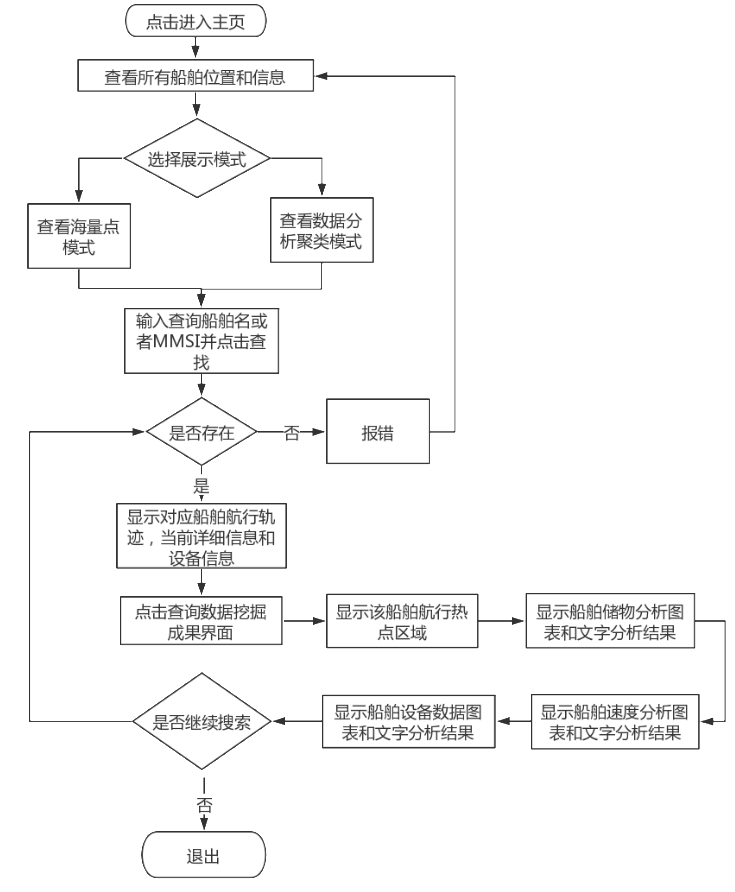


图2.2用户业务流程图

# **技术简介**

## MongoDB

MongoDB由C++语言编写，是基于分布式存储的数据库系统。可以在满足高负载要求的同时，添加节点来保证服务器性能[16]。

其面向文档测存储方式，使得操作更加简单。而文档的数据结构使用类似于JSON的BSON对象，由键值对组成，便于理解；不仅如此，可以设置任何属性的索引，例如存储地理信息的索引来实现更快的查找和排序；对于文档的插入、删除、更新等操作支持丰富的表达式，极为便利；对于数据库的操作也支持Ruby，Python，Java，C++等多种语言。

其最大的特性包括如果需要更大的空间或者计算能力，可以利用分布在网络中的其他节点；可以使用Map/Reduce对数据进行批量和聚合操作。

MongoDB基本概念包括：

（1）文档。文档是数据的基本单位，多个键与其相关联的值的有序结合组成了文档。其中文档允许嵌套。

（2）集合。集合简单来讲就是一组文档，即文档的集合。也就是说可以将不同模式的文档放在一个集合之中。但是为了便于管理，往往有多个集合，集合中一般也存储一种模式的文档。

（3）数据库。多个集合组成了数据库。而一个MongoDB实例可拥有多个数据库，数据库拥有不同的权限。一般存在权限数据库，单台服务器数据库，分片信息数据库。

正是由于MongoDB介于关系型和非关系型数据库之间，所以集中了两者优势。适用于网站数据、大尺寸、高伸缩的业务场景。

## Hadoop分布式框架

Hadoop是Apache基金会开发的分布式系统基础架构[6]。可以通过集群进行高速运算和存储。Hadoop的核心设计包括HDFS和MapReduce，其中HDFS提供了海量数据存储；MapReduce则为数据提供了强大的计算能力。

原本的Hadoop框架和生态圈主要包括以下几个概念：

（1）HDFS。即为Hadoop Distributed File System，用来存储所有节点的文件。对外为一个传统的文件系统，但是其结构如图3.1所示，是基于特殊节点构建的。其中NameNode仅有一个，提供元数据服务，负责管理名称空间和外部访问；DataNode提供存储服。

（2）MapReduce。它是处理大量半结构化数据集合的编程模型，用于大规模数据集的并行运算[16] 。分为Map和Reduce，Map把一组键值对映射成新的键值对，Reduce则保证所有新的键值对共享一个键组。最终实现了数据划分、任务调度、数据和代码定位、优化系统等功能。

（3）Hive是一个数据仓库工具，将结构化数据映射成数据库表，分析海量的日志数据。

（4）Hbase是一个分布式的开源数据库，和传统数据库不同的是，它是基于列模式的数据库。

（5）Yarn是2.0版本新增的系统，负责对集群的调度和管理。具有良好的可用性和扩展性。

（6）还有一些数据收集、机器学习和数据挖掘的框架。如Flume，Mahout等。

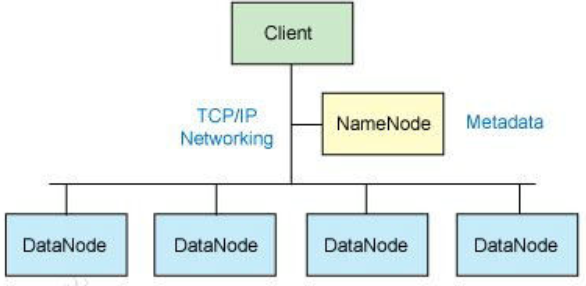


图3.1HDFS结构图

## Spark计算引擎和Scala编程语言

Spark是用于解决大规模数据处理问题的通用计算引擎，具有快速、通用、易用的特点。类似于MapReduce，但将中间结果保存在内存中，无需操作HDFS，所以更适用于数据挖掘中。

Spark有多种运行模式，主要包括local模式开发调试，Standlone模式利用资源管理器和调度器管理集群，Mesos模式运行在Mesos管理框架之上和Yarn运行在Yarn资源管理器上；Spark也有不同的组件进行如图3.2的功能交互。

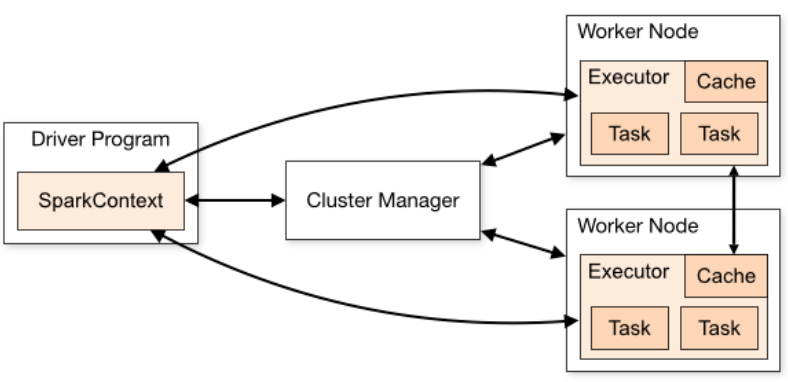


图3.2 Spark组件图

此外Spark能和Scala语言紧密集成，利用Scala即可轻松操作分布式数据。而Scala是类似Java的面向对象编程语言，支持高阶函数，局部套用等多种语言概念。能够更好的支持组件，具有高层的并发模型，可伸缩性强。

## Node.js开发平台和Express框架

Node.js能够使JavaScript运行在服务端，用于构建高速、可伸缩的网络程序，使用了事件驱动和非组赛I/O模型[18] 。它构建在JavaScript之上，与很多NoSQL数据库（包括Mongodb）包括浏览器能紧密结合。

而Express则是官方推荐的简介的Web应用框架。它不对Node.js特性进一步进行抽象，用丰富的工具，方便处理的表单来拓展所需功能，使用任何模板引擎进行渲染，简单快速地创建强大友好的API，。而且与其他框架相比，Express十分轻量，不会引入不需要的东西。

## 高德地图API

地图API是可以将地图嵌入网页的API,并且提供了实用工具对数据和地图进行处理，向地图添加丰富的内容，此外还有更多可以运行的插件可供选择。

其中高德地图API提供了丰富的功能，包括地图的创建和显示、地图图层、坐标转换及标记、信息窗体、绘制图形、3D地图、数据的计算等各种功能。使用方便，极易上手。

## Echart图表框架

Echart是由百度提供的纯JavaScript的图表显示框架。可以流畅运行，兼容绝大部分浏览器。提供直观生动、可交互的可视化图表。

主要提供了常规的折线图、柱状图、散点图、热力图等，还有用于BI（Business Intelligence）的漏斗图等，并且支持图与图之间的混搭。

整个框架非常便于学习和使用，个性化十足，动态效果优秀，易于拓展、支持多维数据。

# **总体设计**

## 系统总体设计

系统功能结构图如图3.1所示，系统总共分为四大功能——船舶数据获取功能、船舶数据挖掘功能、船舶数据管理功能、船舶数据展示功能。对于船舶数据获取功能，首先需要利用网络爬虫技术获取一部分船舶数据，再通过船舶设备数据采集人员的上传上传采集的设备数据。对于数据管理功能，主要拥有Mongodb数据库管理功能，也拥有数据库自带的地理位置索引等索引功能加快查找速度，和数据库的备份功能保证数据库的稳定和安全。对于数据挖掘功能，需要在数据量足够的情况下，针对不同方向进行数据挖掘，并且与数据库管理和数据展示功能进行交互。展示功能则是利用成熟和完善的高德地图API展示；对于挖掘成果也需要利用一些图表生成、展示框架来展示各个方面的数据和结果。

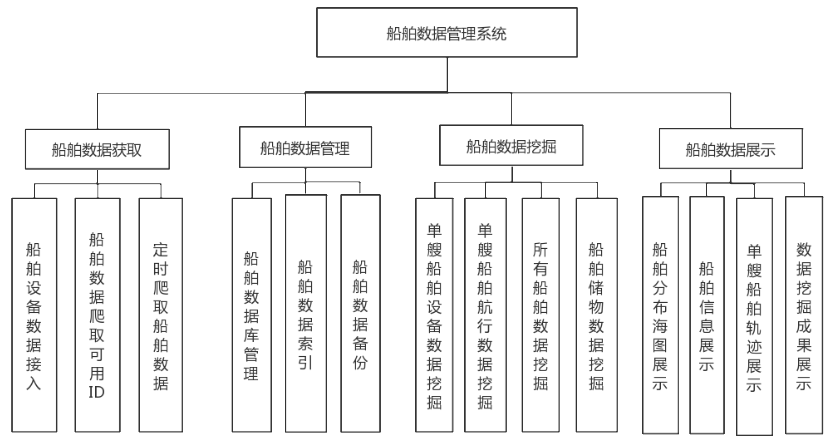


图3.1系统功能结构图

## 系统体系结构设计

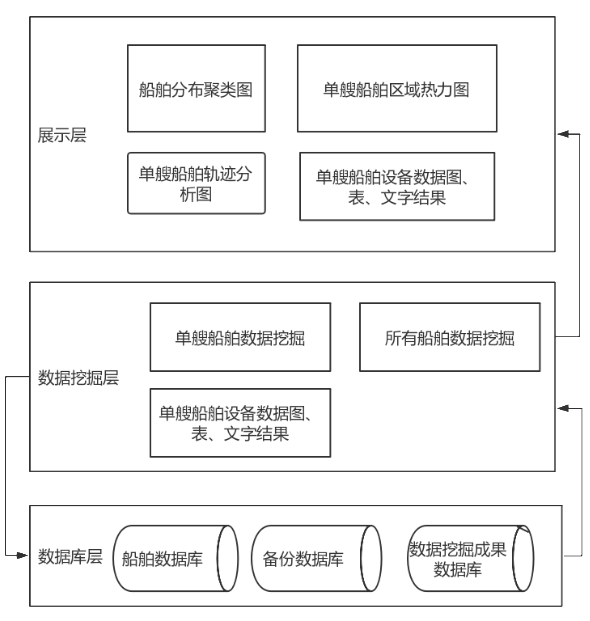


图3.2系统体系结构设计图

系统体系结构图如图3.2所示，主要分为数据库层、数据挖掘层和展示层。层次之间的关系用箭头表示。其含义为数据库层的数据可以为数据挖掘层和展示层提供数据支持。对于数据挖掘层的结果，其中数据量大，计算复杂的结果可以以数据的形式存入数据库便于操作和进一步分析；对于数据少，计算简单快速的结果可以直接进入展示层展示。

# **详细设计与实现**

## 数据库层模块设计与实现

### 船舶数据库详细设计

因为使用了Mongodb以及Hadoop技术，所以船舶数据库设计使用比ER图更加清晰的数据库设计图，如图4.1所示。数据库的详细设计分为三大部分。首先shipinfo主要存储MMSI(水上移动通信业务标识码) 、IMO（国际海事组织的识别码）、船名、船长、载重等基本不会变动的信息。此外由于使用的是Mongodb的数据库技术，所以只需使用shipdyna（是ObjectID类型）就能获得对应的船舶动态信息，而不像关系型数据库需要建立主键的对应关系或声明外键；而shipdyna主要存储船舶位置（GPS）、船首向（朝向）、船舶状态（航行/系泊）等变动信息。而且考虑到在查询等功能场景中经常需要以名称、MMSI等字段查询。为了便于操作，添加了相应的字段；对于shipdynahis，首先是考虑到对于数据挖掘的任务要求设计的数据库，其字段绝大部分与shipdyna这个Collection相同，又因为所有动态信息历史纪录均对应一艘船舶，所以需添加shipinfo来建立对应关系。

对于船舶设备，可以获取的数据主要有发电机功率，主机平均转数，增压器转速，气缸排气量，气缸冷却水温度，燃料油量，柴油量，活塞冷却水温度，机油温度，减速箱油压，增压器压强，海水温度，机舱温度。同理也需要与对应的船舶建立关系。

Mongodb是非关系型数据库，所以只需要在最初配置时进行仔细设计和配置，就可以大大方便之后的使用。其中需要重点就是对于地理索引的建立和使用[14]。在建立地理位置索引时，就需要将经纬度以数组的形式存到GPS 字段中。使用命令行即可建立地理索引。在查找某条船最近的船舶，或者搜索一定范围的船只都可以使用参考文献[13]的方法进行查询。大大加大了查找速率，增强了系统性能。



图4.1数据库详细设计图

### Hadoop环境搭建

由于船舶数量众多，每艘船对应设备产生设备数据更加庞大。所以必须采用Hadoop进行数据的分布式存储。采用两台阿里云云服务器进行环境配置，一台为Centos操作系统，另一台为Ubuntu 16.04操作系统。

Hadoop环境搭建主要步骤分为以下几步：

（1）在两台服务器上通过vim使用命令行查看并修改服务器域名和对应IP。添加用户和用组，赋予相应的操作权限。

（2）设置ssh免密登录。因为Hadoop是采用ssh协议进行Master节点和Slave节点信息交互。所以需要在两台服务器上拷贝对方通过RSA算法加密的密钥至本地相应文件夹，才能实现ssh免密登录。完成后可使用ssh命令进行测试。

（3）主从服务器均需在官网下载Hadoop和JDK文件，解压至合适的文件夹。通过命令行vim ~/.bashrc添加系统环境变量，配置JAVA\_HOME和HADOOP\_HOME等相关环境。完成后可以通过java -version和hadoop version测试。

（4）在hadoop对应文件夹下新建文件夹tmp，在tmp文件夹下再次新建dfs文件夹，dfs下在新建data和name文件夹并赋予相应的权限。

（5）两台服务器均需进入hadoop目录下etc/hadoop文件夹，通过vim修改hadfs-site.xml，core-site.xml，yarn-site.xtml，并新建mapred-site.xml进行相应的修改。主要修改的内容是根据域名、从节点个数、端口以及步骤（4）建立文件夹的位置等信息来进行配置。

（6）完成后均使用hadoop namenode -format格式化HDFS，在主服务上启动hadoop，从服务器也会启动。在hadoop/sbin文件夹下为start-all.sh赋予操作权限并执行即可启动。可以使用jps查看是否启动成功。成果结果如图4.2所示.



图4.2启动成功示意图

### Mongodb与Hadoop的连接配置

正如3.2 Hadoop分布式框架所介绍，Hadoop原本的框架是基于Hbase数据库的框架。于是需要用Mongodb数据库代替Hbase。参考文献[16] 给出了四种不同的结合方式以及结合效果比较。本系统对于二者的连接配置主要需要进行以下几个步骤:

（1）利用git版本控制工具下载mongo-hadoop的开源项目。此项目主要针对Mongodb连接Hadoop提供了解决方案和优化方案。

（2）进入下载的文件夹中，利用./gradlew jar 运行自动配置程序。在此过程中可能会遇见Gradle版本不兼容的问题，需要根据服务器中Java版本选择合适的Gradle版本。

（3）经过较长时间的编译之后，需要在mongodb.github.io/mongo-java-driver/下载合适版本的驱动程序。

（4）之后即可编写Java代码，其中main函数注意设置mongo.input.uri以及output.uri。编译之后利用hadoop jar命令即可。

### 数据备份的设计与实现

此外还进行了数据库备份确保数据库安全。主要包括云服务器快照，脚本自动备份与恢复，本地手动备份。

其中脚本自动备份数据库是利用脚本进行增量备份和全量备份。基本步骤为：

（1）创建备份角色用户或者admin用户。根据使用的备份方式创建不同权限等级的用户，编辑脚本里的config.properties，修改Mongodb连接等配置信息

（2）通过连接配置对相应数据库、备份方式、备份时间等进行选择和设置。

（3）对于数据库的恢复同样根据备份方式设置。增量备份需要创建具有applyOps权限的角色，修改脚本中restore\_inc.py里面的配置；全量备份只需修改restore\_full.py里的配置。最后执行相应的脚本。

脚本的主要通过oplog文件记录数据库操作信息进行备份。但是增量备份只能在集群架构使用，全量备份会删除之前的文件，恢复脚本导入oplog文件。

### 爬虫的设计与实现

经过调研分析发现几乎所有网站都暴露了对应的数据接口，而且经过对比分析发现数据几乎没有太大的差距。所以最终决定直接编写爬虫代码对中国港口网(www.chinaport.com)进行数据爬取。

对爬取的数据进行分析和对比，确定每个字段的物理含义，同时需要将直接获得的经纬度进行转换，为保证精确度保留五位小数并存在GPS数组中进行保存。在按照系统功能结构设计的要求下，先爬取可用的ID存入id.txt，在每隔2个小时读取ID爬取数据。分成两步既可以增加爬虫模块稳定性，也可以分离操作，便于管理。

利用python3的urllib.request和requests模块，填写必要的传输数据，并添加随机User\_Agent，在代理IP池中选择随机IP对目标url以post方式传送数据，对获得的数据进行解析并存入相应的数据库。 这样大大提高了爬虫的稳定性。

在实际运行的过程中发现对于不同的ID爬取结果出现了少部分的重复，对之后数据挖掘造成影响。经过分析是由于不同ID对应的船舶实际是相同的船舶。所以需要进行爬虫的优化和初步的数据清洗。具体思路为对将不同的ID进行验证，将对应的船舶MMSI号记录下来，如果在记录中不存在此MMSI号，则可以记录下ID号；如果存在则舍弃此ID号。

### 设备数据引入的设计与实现

对于设备数据，主要采取模拟数据的方式。通过python脚本每五分钟自动生成所有船舶的设备数据并生成CSV格式文件。之所以设计为五分钟是因为为了保证系统的实时性，需要以较快的速度传入船舶设备数据；有考虑到对于服务器对于文件的处理时间和间隔，最后比较发现五分钟最为合理。其中CSV文件格式满足数据库设计规范，为了尽量模拟真实情况，对各项数据设定了合理范围；并且存储的船舶对应ID来自于爬虫设计所爬取的船舶ID。

CSV文件内数据需要满足一定的顺序，以“shipinfo”，“发电机功率”，“主机平均转速”，“增压器转速”，“气缸排气量”，“气缸冷却水温度”，“燃料油量”，“柴油量”，“活塞冷却水温度”，“机油温度”，“减速箱油压”，“增压器压强”，“海水温度”，“机舱温度”，“更新时间”为对应的顺序。否则存储会发生错误。

对于这些CSV文件，通过设计简单的上传CSV数据的网页，为船舶数据采集系统设计人员提供接口。主要利用form表单样式并设置为file类型来上传多个文件。文件经过Web端和Node.js的处理存入相应文件。之后在服务器编写python脚本定时将所有文件存入数据库。为了保证服务器性能和容量，设置等待时长为一分钟，并在读取存入数据库之后删除该文件。

最终效果如图4.3所示

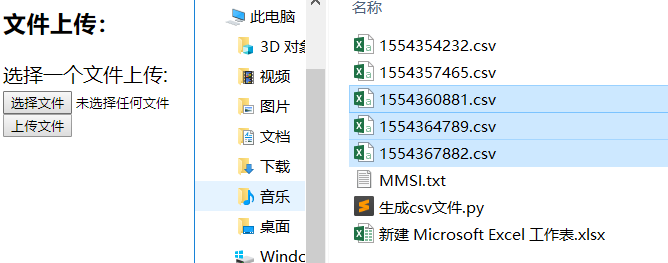


图4.3上传文件功能示意图

## 数据挖掘层设计与实现

### Spark框架的配置和使用

在服务器上配置Spark环境需要进行步骤

（1）在主从服务器均需要首先需要配置java环境，此步骤已在4.1.2 Hadoop环境搭建中完成。

（2）在主从服务器上通过wget命令下载并安装scala安装包，并配置scala环境其中scala详细介绍见3.3。

（3）此外考虑到已搭建分布式环境，所以需要将Spark与Hadoop结合。在主从服务器上均要在安装文件夹下通过wget下载spark-hadoop安装包；利用vim查看并修改/etc/profile添加SPARK\_HOME环境变量；修改SPARK\_HOME下的config/spark\_env.sh文件引入Hadoop，Scala，Java的相关环境；再在config/slaves文件中添加从节点信息。

（4）最后启动脚本查看是否成功。结果与Hadoop环境配置相似。

## 数据展示层的设计与实现

Node.js是非常稳定的JavaScript运行在服务端的开发平台，而且Express是官方推荐的开发框架。所以决定使用相应的技术开发。最终选用WebStorm的集成开发环境，进行了服务端的代码编写。并且通过高德地图的官方API[3]和ECHART）[3]的图表显示框架来进行可视化的展现。

### 4.3.1 直接数据展示的设计与实现

考虑到此系统无需通过客户端向数据库中存入数据，而是查找数据的操作占据绝大多数，所以需要在数据库各种查询操作的接口，利用Mongodb的地理位置索引方便查询。为了满足基于Restful[10]风格的接口，不仅有GET的HTTP请求，也有查询特定船只的POST请求。通过前端HTML网页嵌入的JavaScript脚本，以ajax异步请求方式对接口进行请求并获得数据。

接口主要有查询所有船舶位置，根据名字和MMSI查询特定船舶的详细信息（包括基本信息、速度和当前设备数据），查询某船舶的历史位置， 某船舶的设备历史数据等。

船舶位置的展现主要包括点标记地图，如图4.6所示；考虑到在大数据的背景下，船舶数量巨大的情况，系统需要进行海量点的显示，显示效果如图4.7所示；船舶轨迹和回放效果则如图4.8所示。以上展示效果是基于高德地图点标记API，海量点展示API，信息窗体API等技术进行优化和展示。

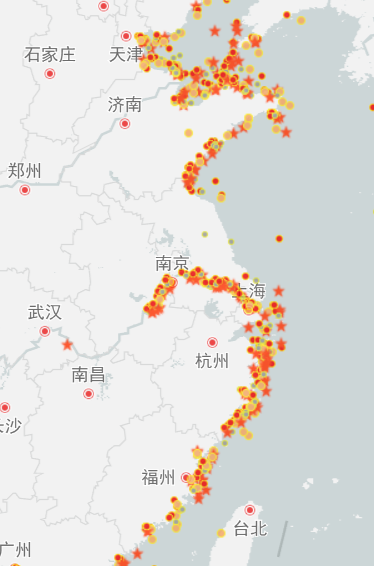
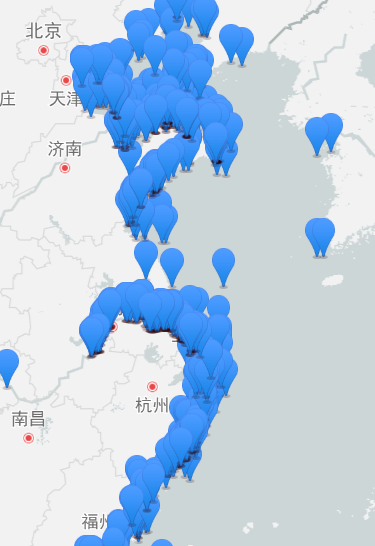


图4.6点标记地图 图4.7 海量点地图

### 4.3.2 数据挖掘结果可视化的设计与实现

通过高德地图的官方API将数据可视化展示在地图上。所有船舶位置界面，所有船舶详细信息显示，某船舶的历史轨迹和回放，所有船舶聚合位置，所有船舶海量位置显示，所有船舶热点区域界面，某船舶航行热点界面。

首先船舶聚类信息挖掘采用高德地图的点聚集API，自动实现船舶位置聚集效果，并且能够在不同的地图比例尺下自动适配地图大小，都很好的展示了不同区域船舶的数量。如图4.9所示。

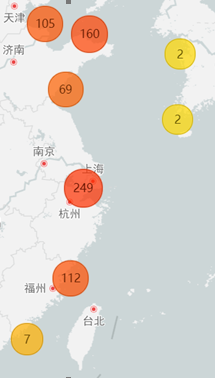
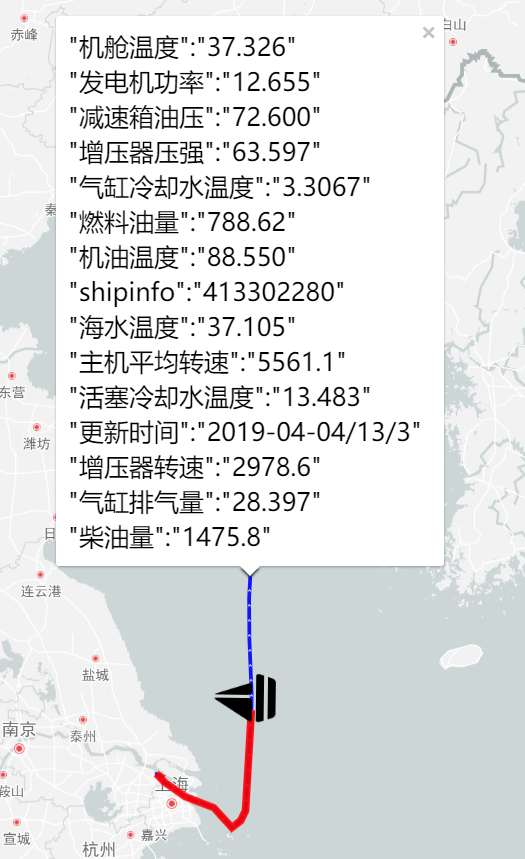


图4.8船舶轨迹及设备信息图 图4.9点聚合图

对于船舶航行热点显示，

对于船舶设备数据，以船舶柴油量为例，

## 系统的整合和搜索功能的实现

首先根据2.1需求分析的结果，将各个功能的可视化界面进行整合，添加跳转链。

对于搜索功能，划分为按MMSI进行搜索和按英文名进行搜索。选择英文名是考虑到在爬虫获得的数据中发现很大部分的船舶信息还不够完善，导致中文名可能为空，但英文名基本完整。实际运行过程中，保持着均以船舶MMSI为主的思想，最终选择先按英文名查找对应的MMSI,再统一利用MMSI进行搜索和查找。另外在查找过程中尽可能使用Mongodb索引加快查找效率，再次不再赘述。效果如图4.10所示。



图4.10 搜索功能和模式选择示意图

## 系统服务器配置

最终需要将整个系统部署在Web端进行展示。所以需要在服务器上配置基本工具，例如Mongodb，Node.js；实用工具，如Screen来运行爬虫脚本，备份脚本，Node服务器等；Hadoop环境，Spark框架，Scala环境等配置不作赘述；申请域名并配置；另外需要解决服务器端传输文件可能存在的问题。最终整个Web应用成功上线。

# **总结与展望**

## **论文总结**

本系统最终完成了项目目标和需求，获取了船舶航行状态数据和模拟的船舶设备数据，完成了对船舶数据的管理。在稳定安全存储的基础上，对各种数据进行数据挖掘，获得了各种有价值的信息。最终通过Web应用的形式可视化展现给用户。

本系统使用的是技术几乎均是在相应领域最为经典或者最新的技术，为用户提供了良好的体验和相对完整的船舶信息。论文也是尽可能详细的介绍设计思想和实现方式，也将所用的详细技术和使用步骤列出。最终形成了相对完整的系统。也为以后改进提供了更多的可能。

## **展望**

总结整个船舶信息获取和处理系统，虽然很好的完成了目标，实现了需求，解决了一些已有系统相应的问题。但是本系统在很多方面都有亟待改善的地方。

（1）最明显也是最重要的问题，就是对船舶设备数据的模拟。本系统是自动随机生成的设备数据。主要是船舶设备数据获取来源有限，且对于一些老旧船舶和船用设备，有些船舶的设备数据甚至进行了加密，所以无法获取大量的真实数据。只能通过模拟。这样的弊端就是对于数据挖掘成果在本项目必然是毫无意义的，只能提供简单的数据导入接口，为之后的真实设备数据的导入提供可能性和方便。

（2）其次，就是对上传CSV文件的处理比较粗糙，必须按照一定的顺序存储。而且会上传至服务器，虽然会在较短的时间内通过脚本分析处理存入数据库，但是这对服务器同样会造成一定的压力，脚本的运行也会占据一定的资源。难以在真正意义上支持特别庞大的数据，而且对上传的吞吐量是一个巨大的挑战。

（3）再次，此系统虽然在数据库设计时存储了时间信息，但是在已有功能上并未对针对时间信息进行太多的比较和操作。但是对于整个船舶数据，时间信息也是连接各种数据之间的纽带。所以还需要加强对时间信息的分析和处理，建立与其他数据之间的联系，这样也可以更好地完善数据挖掘方面的功能。

（4）还有通过AIS系统获得的船舶位置等数据与来源于AMS的船舶设备数据时间间隔差距较大，难以实现数据的匹配。但是又存在着船舶位置等数据在短时间内变化不大的现象，所以需要在以后对部分数据的实行性要求进行更仔细的分析和取舍。

（5）最后，对系统对于数据挖掘研究不够深入。大部分对数据停留在表面的可视化工作上，对于隐藏在其中最深的信息无法挖掘。主要信息主要基于可视化的基础上系统用户对于数据的理解。针对此问题除了对能获得的数据进行更加深入的分析，编写更完善和富有创新性的挖掘算法之外，还可以从人的角度出发，调研和咨询资深的经验丰富的船员，获得更多的思考方式、挖掘方向和理论知识。

# **参考文献**

[1]谢芳.基于互联网的大数据挖掘关键技术分析[J].信息记.

[2]陈意惠,蒲小莲.基于PLC的船舶机舱实时监测报警系统[J].上海海事大学学报,2005,26(2):67-70.

[3]李永攀,刘正江,郑中义.基于时空密度的船载AIS数据聚类分析方法研究[J].重庆交通大学学报（自然科学版）,2018,37(10):117-122.

[4]钱晓江,宿巧丽.船舶航海信息采集系统(MIGS)的设计与实现[J].微型电脑应用,2003,19(8):15-19.

[5]陈玺,马修军,吕欣.Hadoop生态体系安全框架综述[J].信息安全研究,2016,2(8):684-698.

[6]陈丽,黄晋,王锐.Hadoop大数据平台安全问题和解决方案的综述[J].计算机系统应用,2018,27(1):1-9.

[7]李建江,陈玮,李明, 等.基于网格热度值的船舶规律路径提取算法[J].计算机研究与发展,2018,55(5):908-919.

[8]徐思路. 基于船舶轨迹大数据的交通环境时空分析研究[D].大连海事大学,2018.

[9]向玉云,高爽,陈云红,黄嘉成,许新华.百度、高德及Google地图API比较研究[J].软件导刊,2017,16(09):19-21+25.

[10]徐欣威.基于ECharts的科技统计数据可视化设计与实现[J].天津科技,2019(03):66-70.

[11]李晓英,何首武,苏树海.基于NoSQL的海量出租车GPS数据Web服务研究[J].广西科学院学报,2016,32(03):231-236.

[12]高立佳. 基于MongoDB与WebGL的船舶大数据可视化研究[D].大连海事大学,2016.

[13]宋平亮. 基于MongoDB的航道数据Web服务研究[D].大连海事大学,2015.

[14]郑睿博,潘雅静.基于Mongodb以及GIS的海量探地雷达数据存取研究[J].城市勘测,2018(S1):277-280.

[15]周尧,刘超,徐树楠,曹振宇,耿丽丽,刘建川,廖一铧.基于Spark与MongoDB的地理空间大数据应用分析系统设计与实现[J].测绘与空间地理信息,2018,41(09):71-74.

[16]曾强,缪力,秦拯.面向大数据处理的Hadoop与MongoDB整合技术研究[J].计算机应用与软件,2016,33(02):21-24+37.

# **致谢**

不知不觉大四时光已经接近尾声，毕业论文也已完成。回首大学四年在软件学院度过的生活，获益良多。一想到自己是从来自贫困山区，大学之前从未接触软件工程和代码的人变成现在已经能够相对独立地完成毕设这样一个大型的项目，不由得感到非常的高兴与自豪。

在整个毕业论文的完成过程中，首先要感谢我的指导教师——马平老师。在最初接触这个实际项目时，不觉得自己能够担当起如此大型的项目。但是马平老师一直以来的信任给了我足够的信心，并且在设计的过程以及实际编码过程中都得到了很大的帮助。不得不指出的是，在自己开发过程中略微偏离项目主题时，马平老师的及时提醒和批评也促进了自己能够顺利完成此次毕设项目。

其次，我要感谢我的室友。毕设过程中大家都在积极完成自己的项目，也激励了自己的能动性。不仅是他们，还包括同级的同学，我认识的本科毕业参加工作的同学，马平老师实验室的学长学姐们。他们都是我项目开发遇到瓶颈时的咨询对象，正式他们所有人的帮助，我才能最终解决难题，顺利完成毕设。

最后我还要感谢自己的父母，在毕设过程中虽然并不能咨询到技术相关的问题，但是遇到问题时心情沮丧，甚至一个问题长时间没有进展时的心理崩溃，都是他们及时的支持才帮助我重拾信心，面对难关。