基于地铁出行行程的大数据实时分析预测系统详细方案

团队编号:2002066　二仙桥老大爷

目录

1. 项目背景

在新一轮的科技革命和产业变革的浪潮推动下，近些年我国的城轨交通行业信息化建设步入了快速的发展阶段，信息化建设的成果初具规模，改变了传统的建设模式、服务手段和经营方式。为各个相关部门提供科学的数据，并且能够有效的分配资源和人力，提高整个交通系统的安全性、舒适性和经济效益。能够为有关部门处理紧急突发事件提供有效的数据支持和决策依据，尤其是在组织大型活动时、客流量的预测能够帮助轨道交通运营单位做好相应乘客运输能力的调整匹配，既能够保证活动的顺利进行也能够减少对其他居民的影响。

2. 项目目标

（1）挖掘地铁出行数据的数据中的潜在价值和规律；

（2）利用算法模型，合理实现短期和长期的客流分析；

（3）利用分析模型，实现客流高峰的预警；

（4）了解区域的拥挤程度、高峰时段、是否有异常聚集等现象。

（5）对于地铁人流量的实时监控，实现人员的调度后台管理。

（6）将分析结果转换为直观的图形、文字、数字等可视化信息。

3. 问题分析

数据来源巨大，高吞吐，为避免数据丢失，可以采用 kafka。出

行方式分时，应该离线分析，客流分析应该实时在线。正好 kafka 支持实时和离线两种解决方案，一部分数据通过 Flink 做实时计算处理，一部分到 hadoop 做离线分析。

对于客流的预警分析，可以采用 LSTM_Prophet 组合模型先采集地铁出行数据，洗去不规范数据并制作训练集。可以参考一些真实数据或气象相关的数据进行训练，将两者的预测结果使用 BP 神经网络进行组合预测最终可以得出短期和长期的预测结果，并实现预警。

前端展示可以用 spring 框架，echarts 做数据展示。前端展示应该考虑大屏展示。

4. 解决方案

4.1 数据处理解决方案

4.1.1 数据说明

数据主要包括用户进出站点的出站、用户进出站点的入站（此处出站入站统称为更新站点）、：

示例数据包含以下几部分：

1. 约 80W 条行程数据 trips.csv
2. 约 12W 条用户数据 users.csv
3. 站点名称数据 station.csv
4. 2020 年全年节假日数据 workdays.csv

本示例数据格式为 csv 文件。

csv：逗号分隔表，可以表示一个二维表，文件中的每一行对应表格一行数据，行内各个列使用逗号分隔。（以上数据总称为出行数据）

出行行程数据包含 username,line,sta,userid,intime,outtime 等信息其

中,userid 是用户的唯一标识。。in & out time 则代表了更新站点所在位置的时间。sta 代表站点名称。通过以上三种数据就可以得出同一用户或同一站点相关的数据信息。

	A	B	C	D	E
1	用户ID	进站名称	进站时间	出站时间	出站名称
2	ffffc3d68a559b8b6fc	Sta40	2020/7/15 22:03	2020/7/15 22:25	Sta23
3	ffffc3d68a559b8b6fc	Sta23	2020/6/16 21:14	2020/6/16 21:37	Sta13
4	ffffc3d68a559b8b6fc	Sta20	2020/5/6 18:40	2020/5/6 18:45	Sta23
5	ffffc3d68a559b8b6fc	Sta20	2020/6/18 18:46	2020/6/18 18:53	Sta23
6	ffffc3d68a559b8b6fc	Sta23	2020/5/26 9:10	2020/5/26 9:15	Sta20
7	ffffc3d68a559b8b6fc	Sta56	2020/6/8 21:16	2020/6/8 21:25	Sta23
8	ffffc3d68a559b8b6fc	Sta20	2020/6/15 19:22	2020/6/15 19:27	Sta23
9	ffffc3d68a559b8b6fc	Sta23	2020/7/16 9:11	2020/7/16 9:17	Sta20
10	ffffc3d68a559b8b6fc	Sta23	2020/4/24 9:22	2020/4/24 9:26	Sta20
11	ffffc3d68a559b8b6fc	Sta20	2020/4/29 18:31	2020/4/29 19:01	Sta157
12	ffffc3d68a559b8b6fc	Sta23	2020/5/9 8:54	2020/5/9 9:01	Sta20
13	ffffc3d68a559b8b6fc	Sta56	2020/6/29 21:03	2020/6/29 21:13	Sta23

	A	B	C	D
7	Sta103	10号线	Dist5	1154
8	Sta105	2号线	Dist4	1045
9	Sta106	1号线	Dist3	1018
10	Sta107	1号线	Dist2	1010
11	Sta108	1号线	Dist1	1008
12	Sta109	3号线	Dist5	1082
13	Sta11	3号线	Dist5	1089
14	Sta110	1号线	Dist3	1017
15	Sta111	11号线	Dist5	1191
16	Sta112	3号线	Dist5	1085
17	Sta113	3号线	Dist5	1073

4.1.2 基本数据清洗

将 20200608 当天的数据整理为时间序列流。为了达到模拟的效果,Proucer 模块将会每 5s 向 kafka 中发送一次数据,也就相当于 kafka 每 5s 收到一次日志收集数据. 日志收集数据存放于 kafka 的“timeStream” topic 中。

对于题目要求和信令数据的特点设计了基本的数据清洗算法。

首先将数据以流的形式读入,然后按以下几个步骤对数据进行清洗:

1) 使用 flink 的 filter 算子过滤掉数据中的特殊字符包含‘#’,’*’,’^’的条目

2) 使用 flink 的 filter 算子过滤掉行程出数据中为空的条目

3) 使用 flink 的 map 算子将时间戳进行转换

4) 使用 flink 的 filter 算子过滤掉不是 20200608 的数据

5) 使用 flink 的 filter 算子结合本地的经纬度表处理的数据

6) 最后将数据以特定格式输出

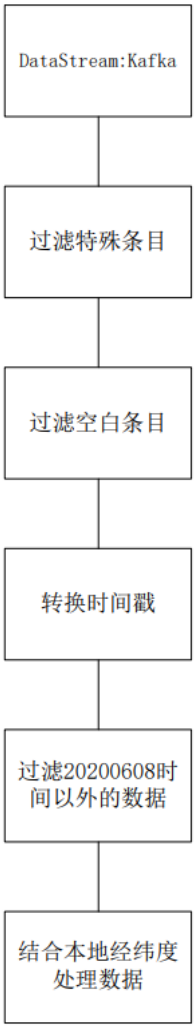


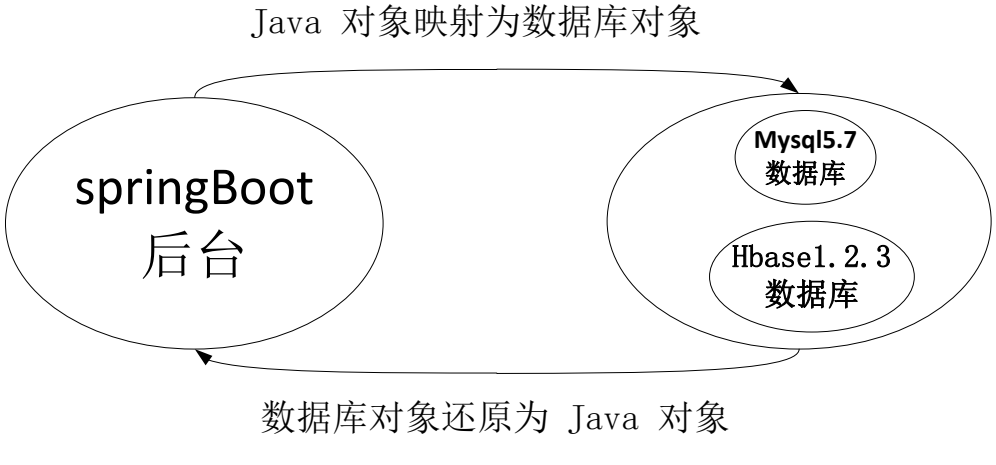
图 1　数据清理图

4.2 数据挖掘分析方法

4.2.1 springBoot 数据存储系统

这种映射机制是双向的,当向数据库存入数据时,是将 java 对象映射为数据库对象,而从数据库取出数据时,却将数据库中的数据还原为 java 对象。关系型数据库都使用了 JPA 的一套执行标准,

他结合 使用 Mybatis 实现了实体的持久化.后续的数据库管理设计都遵循 了 JPA 这一个标准规范,提供了相同的访问数据库的 API。



4.2.2 LSTM_Prophet 组合预测模型

本模型时序数据由某城市地铁乘客行程数据约 80 万条、乘客信息数据约 12 万条、站点名称数据以及节假日数据组成，并将这些数据划分为训练集和测试集，为了提高模型的预测精度并加快收敛速度，需要对训练集和测试集进行归一化处理和反归一化处理。使用 LSTM 模型对数据集进行拟合，设定 batch_size 和 epochs 的可能取值，并由 Grid_Search 对其进行选择，使用均方误差作为损失函数，其值收敛至几乎无变化时，表示使用该组参数的模型最优。使用 Prophet 模型对数据集进行拟合,设置模型的 changepoint_prio_scale 值为 0.2，seasonality_mode 为乘法模型，n_change points 值为 30，yearly——seasonality 为 outo,设置这些参数以后可以提高模型拟合的灵活性，增加曲线的转折点数量，进而提高了模型拟合精度，然后将数据集导入 Prophet 模型中进行拟合，从而得到待预测日期数据的预测结果和训练集的预测结果。

最后使用 BP 神经网络，将训练集的 LSTM 和 Prophet 模型的预测结果作为输入，真实值作为输出，对 BP 神经网络模型进行训练拟合，通过模型训练，由 BP 神经网络确定 2 个模型的预测值在组合中的权重，然后将测试集的 LSTM 和 Prophet 的预测结果作为输入，并加入气象因素，输出通过 BP 神经网络非线性组合后的预测值，并与真实值进行比较，从而判断组合模型的预测效果。

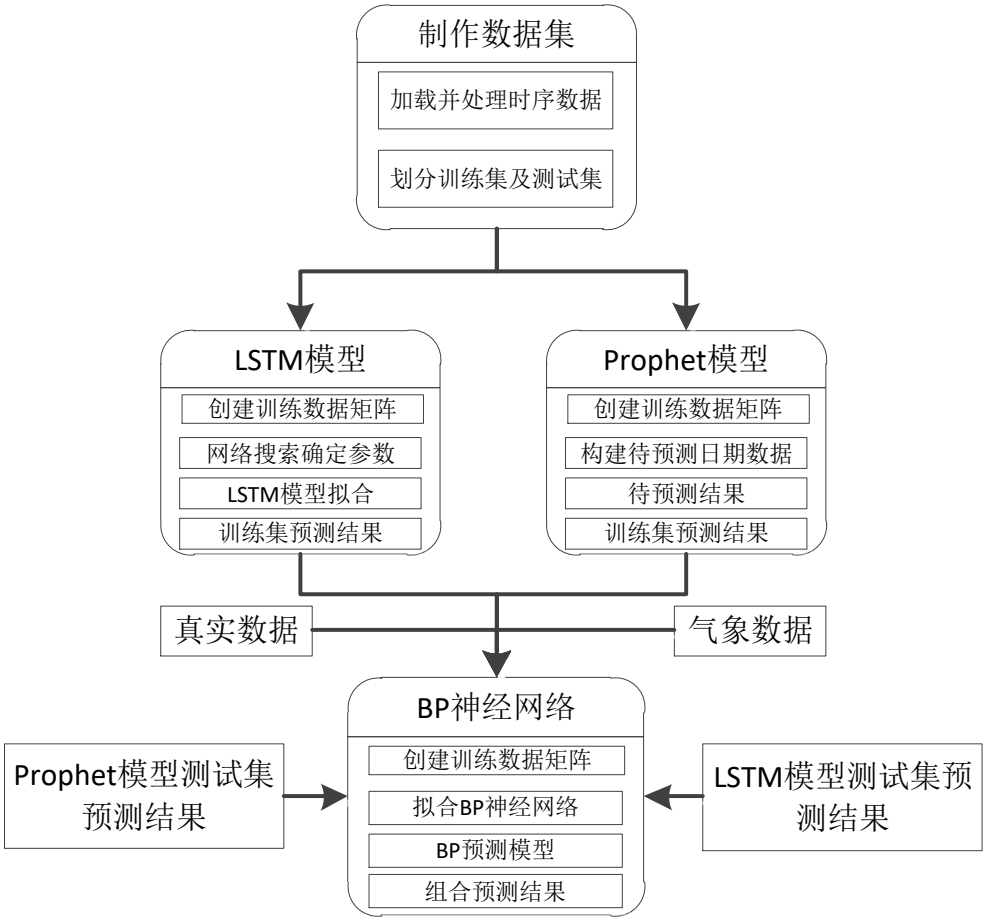


图 1 LSTM_Prophet 组合预测流程

4.3 web 前端展示

SSM 框架整合，即整合 SpringMVC、Spring 和 Mybatis 框架。其

中 SpringMVC 属于 SpringFarmeWork 的后续产品，它提供了构建 Web 应用程序的全功能 MVC 模块，分离了控制器、模型对象、过滤器以及处理程序对象的角色，这种分离让它们更容易进行定制。

Spring 是一个轻量级开源框架，它的主要特点是方便解耦、简化开发、面向切面(AOP)的编程支持和声明式事务支持，其主要优点有低侵入式设计、独立于应用服务器、允许将一些通用任务如日志等进行集中处理。Mybatis 是轻量级 ORM 框架，它消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索，使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJOs 映射成数据库中的记录。其 spring 框架架构图如下：

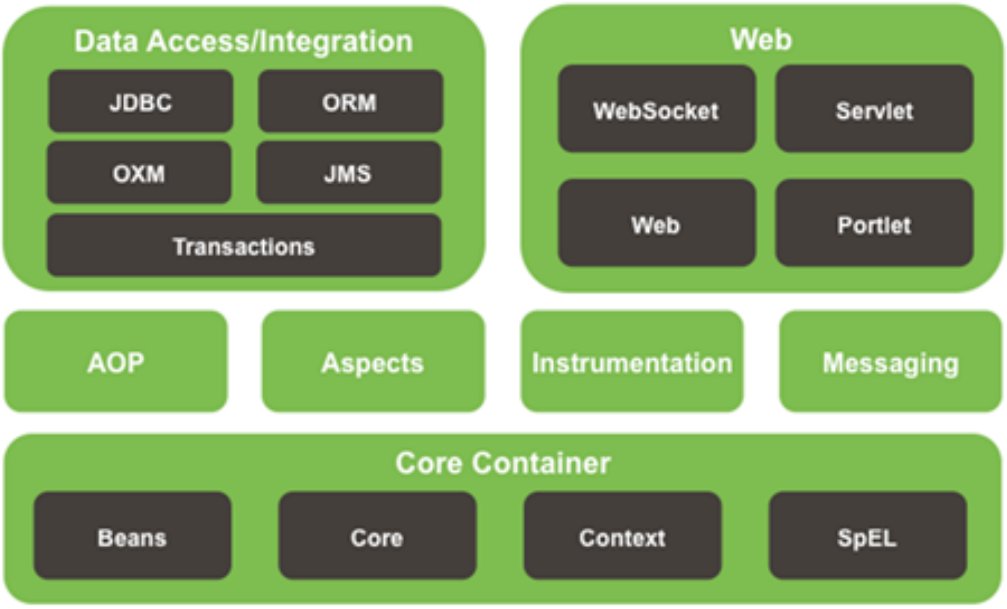


图 2 前端架构总体框架

5. 系统架构和设计

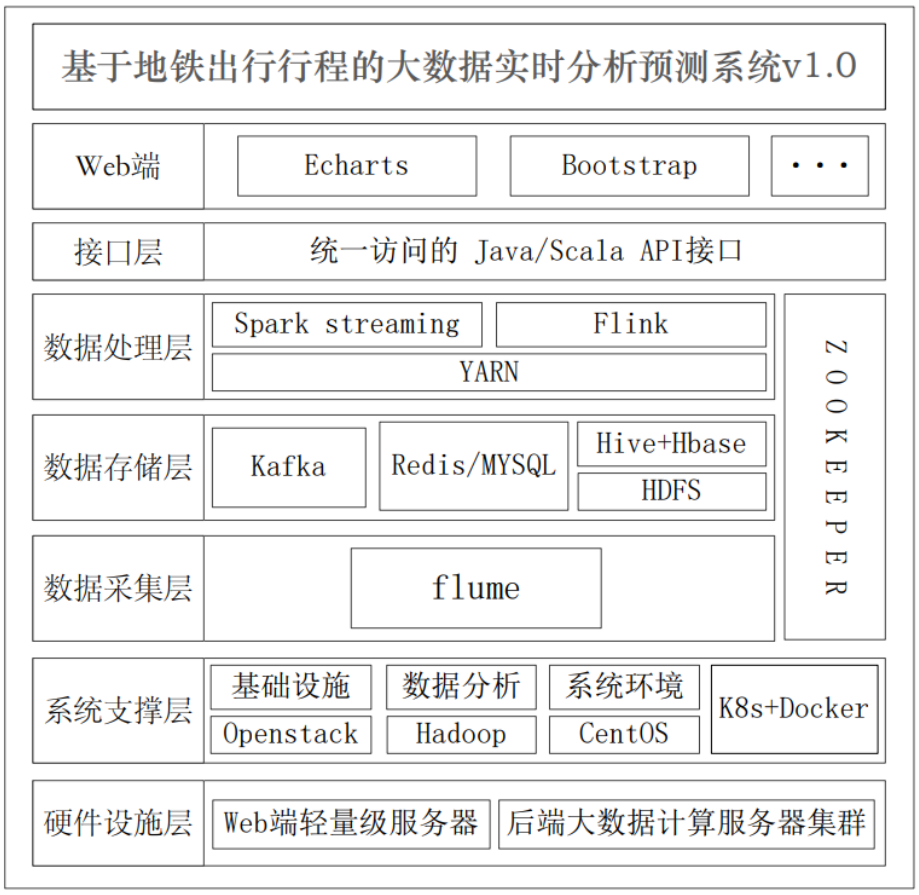


图 3 大数据平台总体架构

6. 项目管理

6.1 项目进度安排

表 6-1 项目进度安排表

序号	任务名称	开始时间	结束时间
1	可行性分析	2021-01-01	2021-01-07
2	需求分析	2021-01-08	2021-01-13
3	技术准备	2021-01-14	2021-01-19
4	概要设计	2021-01-14	2021-01-19
5	详细设计	2021-02-10	2021-02-18
6	系统设计	2021-02-10	2021-02-20
7	系统编码	2021-02-21	2021-03-21
8	系统功能集成测试	2021-03-22	2021-03-23

9	系统测试	2021-03-23	2021-04-14
10	系统修正	2021-03-27	2021-04-18

表 6-2 项目任务分解与进度安排

任务名称	开始时间	结束时间	参与人员	工作量	工作成果
可行性分析	2021-01-01	2021-01-07			可行性分析报告、团队分工
需求分析	2021-01-08	2021-01-13			项目管理计划、需求规格说明书
技术准备	2021-01-14	2021-01-19			项目开发计划、概要说明书
概要设计	2021-01-14	2021-01-19			项目开发计划、概要说明书
详细设计	2021-02-10	2021-02-18			数据库设计、测试设计、详细设计
系统设计	2021-02-10	2021-02-20			概要设计、数据库设计、测试设计
系统编码	2021-02-21	2021-03-21			源代码
系统功能集成测试	2021-03-22	2021-03-23			源代码、功能集成
系统测试	2021-03-23	2021-03-14			功能测试、系统测试、测试文档
系统修正	2021-03-27	2021-04-18			源代码定稿

6.2 项目管理过程

1) 项目组制定项目开发计划，建立人员组织，并进行人员分配。

2) 根据项目开发生命周期启动项目。

3) 召开项目会议，一周一次小会，一月一次大会，并建立会议文档，保证项目进度和保证项目过程出现问题的解决。

4) 小组长扮演项目经理角色，对项目生命周期中的正常运行情况进行监督并对出现的问题进行处理。