

一种新能源客车大面积聚集停放检测系统

杨 凯^{1,2}, 叶 标^{1,2}, 谢金煊^{1,2}, 李 祥^{1,2}, 王江维^{1,2}, 汪昕源^{1,2}

(1. 长沙中车智驭新能源科技有限公司, 长沙 410083; 2. 中车时代电动汽车股份有限公司, 湖南 株洲 412007)

摘 要:为响应政府对企业新能源车辆监管平台建设的要求,本文开发一种新能源客车大面积聚集停放检测系统。先通过 Hive 构建数据仓存储车载终端上传的数据。再筛选离线超过指定天数,且 SOC 大于指定值的车辆信息,并以此作为分析样本集。最后基于 DBSCAN 算法识别车辆样本集,输出超过设定聚集停放车辆数的车辆集群信息。

关键词:新能源客车; 大面积聚集停放; 检测系统; DBSCAN 算法

中图分类号:U469.7; TP274 **文献标志码:**A **DOI:**10.15917/j.cnki.1006-3331.2025.02.003

A Detection System of Large Area Clustered Parking for New Energy Buses

YANG Kai^{1,2}, YE Biao^{1,2}, XIE Jinhuan^{1,2}, LI Xiang^{1,2}, WANG Jiangwei^{1,2}, WANG Xinyuan^{1,2}

(1. Changsha CRRC Intelligent Control and New Energy Technology Co., Ltd., Changsha 410083, China;

2. CRRC Electric Vehicle Co., Ltd., Zhuzhou 412007, China)

Abstract:In response to the government's requirements for constructing enterprise supervision platforms for new energy vehicles, this paper develops a detection system of large area clustered parking for new energy buses. First, building a data warehouse through Hive to store data uploaded by vehicle terminals. Then, screening the filter vehicle information that has been offline for more than a specified number of days and its SOC greater than a specified value, and use this as the analysis sample set. Finally, based on the DBSCAN algorithm, identifying the vehicle sample set and output information on vehicle clusters where the number of parked vehicles exceeds the set aggregation threshold.

Key words:new energy bus; large area clustered parking; detection system; DBSCAN algorithm

随着新能源车辆产量的持续增长^[1],部分营运车企或公交公司存在大面积聚集停放的现象。动力电池一旦出现热失控,可能引发自燃,且火势蔓延迅速,从而造成严重的安全风险和财产损失。因此,应避免大面积聚集停放新能源车辆。为避免大面积聚集停车,各车企现阶段多采用人工巡逻和定期排查的方式。但该方式实时性差,难以及时发现和处理潜在安全隐患。基于此,本文设计一种新能源客车大面积聚集停放集群的检测系统,以代替人工对部分营运车企或公交公司的大面积聚集停放新能源客车进行排查,在节省人力的同时,还提高了隐患车辆排查的准确性。

1 相关技术

本文介绍的检测系统是企业新能源客车数据监管平台的一个功能模块。首先车载终端基于 GB/T 32960^[2]及其扩展协议上传新能源客车数据到企业的新能源客车数据监管平台;再基于 Hive 在指定时间节点对上传到平台的历史数据进行批量分析(下文统称为离线分析),获取满足排查条件的车辆信息。其中排查条件指车辆处于下线状态(即车辆长时间未向监测平台发送数据),且维持该状态超过指定天数,同时车辆最后一次上传的数据中电池剩余容量百分比(SOC)大于设定的阈值。获取满足排查条件的车辆坐标信息,然后通过 DBSCAN 算法对获取的上述

收稿日期:2024-10-21。

第一作者:杨 凯(1989—),男,硕士,工程师,主要从事车联网大数据平台架构设计工作。E-mail:404871394@qq.com。

车辆坐标进行聚合计算。如果在指定半径的圆形区域内存在大于或等于指定数量的车辆坐标点集, 则输出该圆形区域内所有的车辆坐标点集信息。该点集对应的新能源客车则被认为是存在大面积聚停现象的车辆集群。其相关技术如下。

1.1 Hadoop 平台

Hadoop 作为一个分布式文件系统平台, 具有高可靠性、高效存取效率和易于扩展性^[3]。其核心组件 HDFS 将数据以文件的形式分散存储于 Hadoop 集群的多台服务器节点上。集群一般由一个 NameNode 和多个 DataNode 组成, NameNode 负责管理 HDFS 的命名空间和元数据信息, 它将文件划分成数据块, 并将这些数据块分配给 DataNode 进行存储^[4]。本文采用 Hadoop 平台作为新能源客车车载终端上传数据的数据仓库, 同时利用其文件存储、管理和计算等功能, 将清洗处理后的新能源客车车载终端数据划分成多个数据块, 并分配到不同的计算节点上进行并行处理, 从而提升数据处理的效率。

1.2 Hive 系统

Hive 是基于 Hadoop 平台的数据仓库系统, 主要用于对存储在 HDFS 中的大数据进行查询和分析计算。Hive 系统为数据分析人员提供了 HQL 查询语言。HQL 查询语言由 Hive 解析器编译为一组 MapReduce 程序。与关系数据库类似, Hive 系统将这些保留了原始结构和格式的数据以表的形式进行组织并存储到 HDFS 中。由于 Hive 系统将表的元数据信息与 HDFS 上的物理存储位置进行映射, 用户可以通过 Hive 表名来访问 HDFS 上的数据^[5]。本文基于 HQL 对存储在 HDFS 上的车辆数据进行清洗、转换、聚合等操作, 为后续数据分析和挖掘提供数据源。

1.3 Flink 框架

Flink 是一个分布式、高性能的实时数据流处理框架, 主要用于对无界和有界数据流进行状态计算, 数据流中的“状态”可以看作中间计算结果或历史计算结果^[6]。本文使用 Flink 从 Kafka 消息队列中实时采集数据并进行预处理, 然后将预处理后的数据存储在 HDFS 中, 供后续查询分析使用。

1.4 DolphinScheduler 系统

DolphinScheduler 是一个基于 DAG(有向无环图)

的工作流任务调度系统^[7], 其通过任务间的依赖关系直观地编排、调度和管理复杂的任务流程, 实现高效的工作流自动化。同时可实时监控任务的运行状态, 适用于多种任务类型。本文使用 DolphinScheduler 调度 Hive 和 Python 的脚本任务。

1.5 DataX 工具

DataX 是阿里巴巴公司开源^[8]的一个异构数据源(多种不同数据源)离线同步(在指定时间节点将一个数据源的数据转换成其他数据源存储格式, 并存储到其他数据源的过程)工具。基于 DataX 工具能够实现包括关系型数据库(MySQL、Oracle、PostgreSQL 等)、HDFS、Hive、ODPS、HBase 等各种异构数据源之间稳定且高效的数据同步功能。本文通过 DataX 工具将监控平台中的车辆属性数据同步到 Hive 中进行聚合计算。

1.6 Kafka 工具

Kafka 作为一个分布式的发布/订阅信息传递的中间件^[9], 具有可扩展、高吞吐量、有序化、持久化等特性, 常用于处理异步通信、应用解耦以及流量控制等问题。

1.7 DBSCAN 算法

DBSCAN 是基于密度空间的聚类算法^[10], 通过连接高密度区域来形成聚类, 同时识别出密度小于任一簇类的噪点(又称为异常点)。本文设计的 DBSCAN 算法定义如下:

1) 样本集 $D = (x_1, x_2, \dots, x_n, \dots, x_m)$ 为车辆坐标点集, 其中 x_n 为第 n 辆车的坐标点。

2) 车辆坐标 x_j 属于 D 中样本。

3) ε -邻域为车辆坐标样本集 D 中车辆坐标 x_i 与车辆坐标 x_j 的距离不大于 ε 的子样本集, 子样本集车辆坐标集的个数记作 $N\varepsilon(x_j)$, $N\varepsilon(x_j) = \{x_i \in D \mid \text{distance}(x_i, x_j) \leq \varepsilon\}$ 。

①如果存在一个点 $x_j \in D$, 且 $N\varepsilon(x_j) \geq \text{MinPts}$ (一个给定的最小点数值, 本文为设定的密集阈值), 则称该点为核心点。

②如果一个点不是核心点, 但它位于某个核心点的 ε -邻域内, 则该点被称为边界点。

③既不是核心点也不是边界点的点被称为噪声点。

本文基于 DBSCAN 算法计算大面积聚集停放车辆的信息。通过设置 ε -邻域范围和 MinPts 值(阙值)来筛选包含核心点和该核心点 ε -邻域中的所有边界点的车辆坐标点集。这些点集对应的车辆即是本文需要排查的,存在大面积聚集停放隐患的车辆。

2 检测系统架构设计

本文设计的检测系统使用 CentOS 云服务器搭建 Hadoop 平台,并将 Kafka 工具作为数据信息跨平台传递的中间件。首先,车载终端按照 GB/T 32960 规定

的协议格式上传本新能源客车数据到新能源监控平台,然后该监控平台将解析后的实时数据存入 Kafka 中间平台,最后通过实时数据流处理框架 Flink,将 Kafka 中间平台中的数据分别存入 Hbase 数据库和写入 HDFS 对应的 Hive 表中,同时通过 DataX 工具同步监控平台中车辆相关静态信息(如车架号、车型、所属公交公司、运行线路等)并保存到 HDFS 中,Hive 中经过清洗加工的数据也同样存储在 HDFS 中。其中检测系统 Hadoop 平台的集群资源架构如图 1 所示。

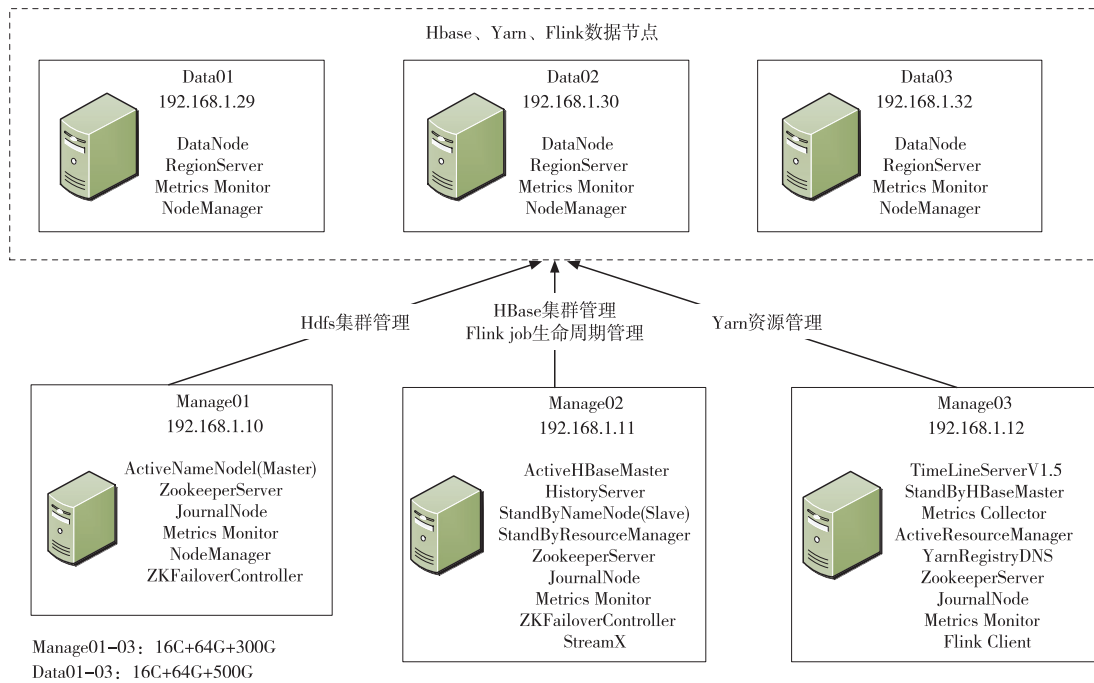


图 1 检测系统 Hadoop 平台的集群资源架构图

3 检测系统数据仓库设计

数据仓库依次分为操作数据处理层(operational data store,简称 ODS 层)、数据仓库明细层(data warehouse detail,简称 DWD 层)、维度数据层(dimension,简称 DIM 层)、数据仓库汇总层(data warehouse summary,简称 DWS 层)和应用数据层(application data store,简称 ADS 层)。

1) ODS 层: Flink 实时解析车辆原始数据,并将解析后的数据存入 ODS 表。

2) DWD 层: 从 ODS 层中抽取车辆定位数据、电池电量数据,并进行清洗和过滤,然后存入 DWD 表。

3) DIM 层: 基于 DataX 从平台数据库同步描述车辆属性的标签数据到 DIM 表。

4) DWS 层: 与 DIM 表聚合获取车辆最新一条数据采集时间,然后记录最后一条数据的电池 SOC 和车辆定位信息。

5) ADS 层: 存储应用 DBSCAN 算法计算出来的结果集,并与 DIM 表数据聚合。

4 检测系统应用

本文设计的新能源客车大面积聚集停放检测系统,首先从新能源客车数据监管平台中获取解析后的车辆数据,并写入到 HDFS 中,然后通过搭建 DB-

SCAN 算法模型对数据进行计算, 最后将得到的结果在监管平台上进行展示。系统处理数据的流程如图2所示。具体处理可分为4步。

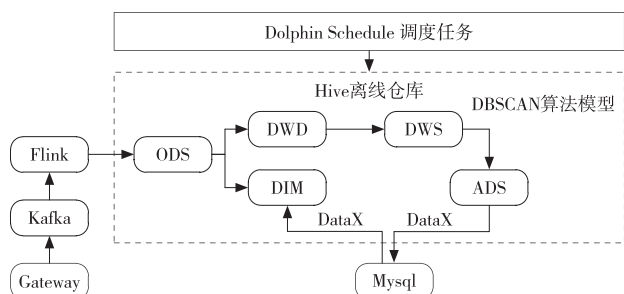


图2 数据处理流程图

4.1 车辆数据获取

车载终端将 GB/T 32960 及其扩展协议数据上传到新能源客车数据监管平台, 包括本文检测所需的 SOC 数据、车辆定位数据、下线天数数据, 其中下线天数数据为平台根据车辆最后一次上传数据时间计算得出。监管平台将采集到的车辆实时数据进行解析并存入 Kafka 消息中间件, 检测系统通过 Flink 获取 Kafka 中间件中存储的数据, 并对数据进行解析后将数据写入 HDFS, 并作为 Hive ODS 表。

4.2 数据加工处理

本文使用到的 Hive ODS、DWD、DIM、DWS、ADS 表均通过 Hive 的元数据与 HDFS 进行关联和交互。通过编写 Hive SQL 语句, 并撰写执行 Hive SQL 语句的脚本, 对 ODS 层数据进行异常数据清洗和相同采集时间数据过滤, 将出现 SOC“异常”或“无效”状态值的数据 (GB/T 32960 规定 SOC 值为 0xFE 时为异常, 0xFF 为无效) 过滤, 同时去除采集时间相同的数据 (车辆补发时可能导致数据相同), 然后加载到 DWD 表中。

通过将 DWD 表中数据与限制条件信息 DIM 表中数据聚合来生成 DWS 表, 本文 DIM 表为 DataX 同步业务平台设置的限制条件, 即 DBSCAN 算法的核心参数和数据筛选条件。其中包含: ①设置 ϵ 值 (即 ϵ -邻域值), 本文设置邻域形状为圆形, 则 ϵ 代表圆的半径; ②设置 MinPts 值, 即满足大面积停放集群条件的邻域包含最小车辆数; ③下线天数阈值; ④电池 SOC 阈值。这些 DBSCAN 算法的核心参数和入

参数据筛选条件可根据停车区域面积变化和领域车辆数的增长趋势进行调整。

DWS 表包含车架号 (vin)、SOC、车辆定位经度 (gps_lon)、定位纬度 (gps_dim)、最近一次车辆发送数据时间 (latest_event_time)、最近一次车辆发送电池数据时间 (latest_battery_event_time)、最近一次车辆发送定位数据时间 (latest_location_event_time)、截止目标日期车辆离线时间 (offline_days)、数据处理时间 (system_time)、目标日期 (process_date)、 ϵ 区域半径值 (eps)、成为核心点最小样本数 (MinPts)。

4.3 DBSCAN 算法实现

本文基于 DolphinScheduler 系统在任务中调度执行 DBSCAN 算法的 Python 脚本对 DWS 表中的数据进行分析判断。算法步骤如下:

1) 初始化。将车企停放的所有新能源车辆的坐标点作为算法的样本集, 并将样本集所有坐标点标记为未访问。

2) 随机选择一个未访问的车辆坐标点 x_j , 将其标记为已访问, 并判断其是否为核心点, 即判断以车辆坐标点 x_j 为圆心, 半径 500 m (ϵ 值为 500 m) 的圆形区域内, 是否存在超过 30 个 (MinPts 值为 30 个) 其他车辆坐标点。

3) 如果 x_j 是核心点: ①创建 2 个新的簇 C、D, 并将 x_j 添加到 C 中。②查找 x_j 的 ϵ -邻域内所有未访问的坐标点, 将它们标记为已访问并存入簇类 D 中。③对于簇类 D 中的每一个点 Q, 如果以 Q 点为圆心, 半径 500 m 的圆形区域内存在超过 30 个其他坐标点, 则 Q 点也是核心点, 将其添加到 C 中, 并继续查找 Q 的 ϵ -邻域内的未访问点, 并将未访问点存入 D 中。重复此过程, 直到没有新的点可以添加到 C 中。

4) 如果 x_j 不是核心点, 则从样本集中另外寻找一个随机点, 标记该点为已访问并判断其是否为核心点。如果是核心点则继续第 3) 步; 如果不是则继续本步骤。

5) 重复步骤 2)、3)、4), 直到所有点都被访问过, 并形成簇类 C (核心点点集) 和簇类 D (边缘点点集)。

6) 处理噪声点。在聚类过程中, 未被纳入任何簇类的点被视为噪声点。

本文主要通过 2 辆车最后一次上线时刻的经纬度坐标,基于 Geodesy 库计算两个地理位置之间的距离。经过 DBSCAN 算法得到核心点和边界点点集,并写入 ADS 表中。

4.4 数据分析结果展示

本文将某客运集团 1 000 辆车作为测试数据集,

如图 3 所示。初步设置 DBSCAN 算法的入参数据筛选条件如下:①下线天数阈值为 100 天;②SOC 阈值为 90%。初步设置 DBSCAN 算法核心参数如下:①eps 为 500 m;②MinPts 为 30 辆。

	vin	SOC	gps_dim	gps_lon	latest_event_time	offline_days	system_time	process_date
1	LHWN431A0R1240641	38.8	28.54699378	111.41740869	2024-09-01 15:59:15.000	8	2024-09-10 09:08:56.109	20240909
2	LHWN413A1R1240566	99.6	35.07191878	118.14970769	2024-09-01 14:12:39.000	8	2024-09-10 09:08:56.109	20240909
3	LHWN431A6R1240644	40	28.54724778	111.41744469	2024-09-01 16:26:07.000	8	2024-09-10 09:08:56.109	20240909
4	LHWCE43D3P1232546	81.6	31.24384878	119.48093769	2024-08-31 17:24:14.000	9	2024-09-10 09:08:56.109	20240909
5	LHWCD33D3P1232768	83.6	31.10463878	119.35064769	2024-08-30 15:28:25.000	10	2024-09-10 09:08:56.109	20240909
6	LHWN413A0R1240574	97.6	27.49849878	113.03729769	2024-08-30 15:28:28.000	10	2024-09-10 09:08:56.109	20240909
7	LHWN413A9R1240573	96.8	27.49849278	113.03718169	2024-08-30 16:06:23.000	10	2024-09-10 09:08:56.109	20240909
8	LHWCD43D5P1232521	76.8	31.23294678	119.48785069	2024-08-30 08:35:12.000	10	2024-09-10 09:08:56.109	20240909
9	AD202309141335679	87.6	24.11160178	125.19180669	2024-08-23 15:35:13.000	17	2024-09-10 09:08:56.109	20240909
10	LHWCL97D5R1240391	44	31.19459978	121.40689369	2024-08-19 05:32:55.000	21	2024-09-10 09:08:56.109	20240909
11	L1K1H8GH9P0231990	97.2	29.44231678	106.29522869	2024-08-16 12:08:57.000	24	2024-09-10 09:08:56.109	20240909
12	LHWCD53D1L1201430	70.8	28.54990978	111.41600269	2024-08-14 08:42:28.000	26	2024-09-10 09:08:56.109	20240909
13	LHWH413A7R1240667	39.2	27.49739978	113.03804969	2024-08-13 17:29:28.000	27	2024-09-10 09:08:56.109	20240909
14	LHWN413A0R1240395	96	26.00901578	119.20674869	2024-08-09 17:47:34.000	31	2024-09-10 09:08:56.109	20240909
15	L1K1D8FF3R0249920	53.2	27.49798478	113.03812269	2024-08-08 11:42:56.000	32	2024-09-10 09:08:56.109	20240909

图 3 部分测试数据集

首先按 4.2 节对图 3 所示数据进行加工处理,得到满足业务平台限制条件的 DWS 数据集,部分数据集如图 4 所示。然后将图 4 所示的 DWS 数据作为 DBSCAN 大面积聚集停放算法的输入参数,经过算法

模型计算得到满足 $\text{eps} = 500\text{ m}$, $\text{MinPts} = 30$ 辆的聚集停放车辆数据,部分结果如图 5 和图 6 所示。图 5、图 6 中 core_vin 代表核心点所属车辆的 VIN(车架)号, range_count 表示核心点区域内所包含的边缘点个数。

	vin	SOC	gps_dim	gps_lon	latest_event_time	offline_days	system_time	process_date
1	LHWC64D1K1195543	90.4	29.59569478	104.09560869	2024-05-31 21:46:25.000	101	2024-09-10 09:08:56.109	20240909
2	LHWCJ85D4H1172438	91.2	29.15040178	104.06715869	2024-05-27 11:24:31.000	105	2024-09-10 09:08:56.109	20240909
3	LHWCJ95D3H1171562	92.8	29.04490978	104.01179069	2024-05-25 06:51:31.000	107	2024-09-10 09:08:56.109	20240909
4	LHWCJ95D9J1180322	90.4	29.08865178	104.55191369	2024-05-17 15:46:51.000	115	2024-09-10 09:08:56.109	20240909
5	LHWCJ95D2J1184292	90.8	29.16803278	104.03517169	2024-04-21 11:27:57.000	141	2024-09-10 09:08:56.109	20240909
6	LHWCJ95D8J1181588	91.6	29.14308878	104.11498369	2024-03-28 15:57:29.000	165	2024-09-10 09:08:56.109	20240909
7	LHWCJ95D0J1180211	92.8	29.56754778	104.36288269	2024-03-28 15:57:22.000	165	2024-09-10 09:08:56.109	20240909
8	LHWCJ95D0J1180337	92.4	29.08945778	104.55311669	2024-03-08 21:46:02.000	185	2024-09-10 09:08:56.109	20240909
9	LHWCJ95D0H1171583	92	29.08815578	104.54701969	2024-03-06 10:53:41.000	187	2024-09-10 09:08:56.109	20240909
10	LHWCJ95D4H1171571	90.8	29.08800378	104.54607469	2024-03-05 19:33:22.000	188	2024-09-10 09:08:56.109	20240909
11	LHWCJ95D4H1171585	92.8	29.08826278	104.54698969	2024-03-05 20:10:57.000	188	2024-09-10 09:08:56.109	20240909
12	LHWCJ95D1H1171608	91.2	29.01081978	104.58080569	2024-03-01 13:11:18.000	192	2024-09-10 09:08:56.109	20240909
13	LHWCJ95D1H1171558	91.6	29.01051978	104.58060469	2024-03-01 11:18:29.000	192	2024-09-10 09:08:56.109	20240909
14	LHWCJ95D0H1171616	93.2	29.01094078	104.58077069	2024-03-01 11:13:39.000	192	2024-09-10 09:08:56.109	20240909
15	LHWCJ95D9J1180353	92.8	29.06009178	104.00550969	2024-02-27 17:30:16.000	195	2024-09-10 09:08:56.109	20240909

图 4 部分 DWS 数据集

	vin	SOC	gps_dim	gps_lon	latest_event_time	offline_days	core_vin	range_count	process_date
1	LHWCJ95D1H1171608	91.2	29.333862	104.780983	2024-03-28 15:57:32.000	165	LHWCJ95D7J1185549	10	20240909
2	LHWCJ95D3H1170797	94	29.345846	104.774490	2024-03-28 15:57:32.000	165	LHWCJ95D7J1185549	3	20240909
3	LHWCJ95DXJ1180264	92.8	29.335321	104.784020	2024-05-11 10:05:32.000	121	LHWCJ95D7J1185549	18	20240909
4	LHWCJ95DXJ2180164	91.3	29.335221	104.783920	2024-05-14 10:05:32.000	118	LHWCJ95D7J1185549	3	20240909
5	LHWCJ95D7J1185541	91	29.335248	104.784720	2024-05-21 12:04:32.000	111	LHWCJ95D7J1185549	24	20240909
6	LHWCJ95D7J1185549	90.2	29.335122	104.783929	2024-05-21 12:04:32.000	111	LHWCJ95D7J1185549	30	20240909

图 5 DWS 部分核心点、边界点数据集

	asc vin	123 SOC	asc gps_dim	asc gps_lon	latest_event_time	123 offline_days	asc core_vin	123 range_count	asc process_date
1	LHWCJ95D9J3125519	90.5	104.793929	29.345122	2024-03-03 21:57:32.000	140		0	20240909
2	LHWCJ95DXJ2109979	91.2	104.796278	29.343471	2024-03-03 21:52:18.000	140		0	20240909
3	LHWCJ96D8H3342812	90.4	104.792385	29.348309	2024-03-03 21:55:41.000	140		0	20240909

图 6 DWS 部分噪声点数据集

下面举例对结果集中有关信息进行说明:

1) 结果集中显示 VIN 为 LHWCJ95D1H1171608 的这辆车为边界点,即以该车辆坐标点为圆心时,半径 500 m 的圆形区域内只存在 10 个车辆坐标点,即以该车辆为圆心时,500 m 半径范围内不存在聚集停放现象。而 VIN 为 LHWCJ95D7J1185549 的这辆车为核心点,以该车辆坐标点为圆心时,半径 500 m 的圆形区域内刚好存在 30 个其他车辆坐标点,即以该车辆为圆心时,500 m 半径范围内存在聚集停放现象。

2) VIN 为 LHWCJ95D9J3125519 的这辆车为噪

声点,即以该车辆坐标点为圆心时,半径 500 m 的圆形区域内不存在其他任何车辆,因此该车辆 core_vin 为空,range_count 为 0。

最后,为便于排查,将图 5 和图 6 所示的结果集数据以核心点为区域进行聚合,聚合后的结果集即为 ADS 数据集,部分 ADS 数据集如图 7 所示。通过 DataX 同步到平台 Mysql 数据库, domain_location_name 对应大面积停车区域核心点车辆位置, domain_location 对应核心点车辆坐标, domain_count 对应区域停放车辆数, vin_list 代表区域停放车辆的 VIN (车架)号集合。

	123 rank	asc domain_location_name	asc domain_location	123 domain_count	asc vin_list
1	1	xx客运停车场	104.783929,29.335122	32	LHWCJ95D7J1185549,LHWCJ85D1H1173014,LHWCJ85D2H1172938,LHWCJ97D3J1183162,...
2	2	xxx东站-停车场	104.827467,29.402022	30	LHWCJ65D1P1231804,LHWCJ85D1H1172798,LHWCJ95D1H1171575,LHWCJ95D6J1180293,...
3	3	xxx西站-停车场	104.729092,29.33653	30	LHWCJ64D3H1170152,LHWCJ64D4K1195553,LHWCJ64D6K1195778,LHWCJ64D9K1194642,...

图 7 部分 ADS 数据集

为验证算法的准确性,在同步到监控平台的结果数据集中随机挑选一组大面积停车区域的车辆,并导出所有车辆与核心点车辆坐标的距离,部分结果如图 8 所示。图中 core_vin 代表核心点车辆 VIN (车架)号、core_location 代表核心点车辆坐标,edge_vin 代表边缘点车辆 VIN (车架)号、edge_location 代表边缘点

车辆坐标、distance_km 代表两车之间的距离(单位 km)。随机抽查结果集中的 10 辆样车发现,500 m 半径范围内均存在离线天数超过 100 天、SOC 值大于 30%的车辆,这也证明了算法的准确性。

以上车辆坐标系的原点在核心点车辆处(即圆心处),坐标值的单位为 m。

	asc core_vin	asc core_location	asc edge_vin	asc edge_location	asc distance_km
1	LHWCJ95D7J1185549	104.783929,29.335122	LHWCJ85D1H1172798	104.788190,29.335846	0.420817
2	LHWCJ95D7J1185549	104.783929,29.335122	LHWCJ95D1H152797	104.764490,29.335846	0.42127
3	LHWCJ95D7J1185549	104.783929,29.335122	LHWCJ95D3H1170797	104.779190,29.335146	0.47892
4	LHWCJ95D7J1185549	104.783929,29.335122	LHWCJ95D1H2918752	104.779190,29.333148	0.37421
5	LHWCJ95D7J1185549	104.783929,29.335122	LHWCJ95D3H29141397	104.764490,29.335442	0.28927
6	LHWCJ95D7J1185549	104.783929,29.335122	LHWCJ95D2H14701990	104.764490,29.334146	0.49811

图 8 部分核心点与边缘点的距离数据集

5 结束语

本文开发了一种新能源客车大面积聚集停放检测系统。基于 Hive 中存储的车辆离线数据进行清洗、加工、筛选、聚合操作,获取维持“下线状态”超过指定天数,且 SOC 大于指定阈值的车辆及其坐标信息,并存于 DWS 表,随后将 DWS 表中数据通过 DB-SCAN 算法模型计算大面积聚集停放的车辆坐标点

集,并将车辆坐标点集推送给相关责任人进行排查,为解决大面积聚集停车带来的安全隐患问题提供有效手段。

参考文献:

[1] 郑雪芹. 2024 年 5 月国产新能源汽车销量 95.5 万辆[J]. 汽车纵横,2024(7):106-107.

客车与座椅性能的要求,对客车前后排共用的双人靠背座椅进行分析及优化,最终实现强度及伤害性能均达标。本文主要研究成果如下:

1) 针对座椅强度和头部伤害的初始性能均未达标的问题,提出了一种分步骤的优化方法,先使强度性能满足要求,然后调整配置类部件,以满足伤害性能要求,提高了解决问题的效率。

2) 采用实验设计方法,有效地平衡了座椅结构弱化或加强的不利影响。系统性的研究方法不仅成功地提升了座椅性能,还有效降低了解决方案的成本。

该座椅优化方法具有代表性,可为类似座椅的改进提供参考,其工程应用价值较高。

参考文献:

- [1] 中华人民共和国工业和信息化部. 客车座椅及其车辆固定件的强度:GB 13057—2023[S]. 北京:中国标准出版社, 2023:3-6.
- [2] 颜长征,王若璜,张科峰. 基于 GB 13057 的客车座椅结构对乘员伤害的分析[J]. 机械研究与应用, 2020, 33(2): 160-163.
- [3] 曾锋,林标华. 基于 GB 13057 讨论稿的客车座椅安全性研究[J]. 客车技术与研究, 2021, 43(5): 53-56.
- [4] 徐海澜,王若璜. GB 13057—2023《客车座椅及其车辆固定件的强度》修订解读[J]. 客车技术与研究, 2023, 45(6): 57-60.
- [5] 徐海澜,王若璜,朱红岩. 客车座椅车辆固定件强度静态试验载荷确定方法研究[J]. 客车技术与研究, 2023, 45(5): 49-52.
- [6] 童劭瑾,苏荣敏,汪俊,等. 汽车座椅安全带固定点强度分析[J]. 机械研究与应用, 2016, 29(6): 57-60.
- [7] 朱宁宇. 不同人体正碰响应分析及基于中国人体的约束系统优化[D]. 长沙:湖南大学, 2010.
- [8] 彭旺,张雅鑫. 客车正面碰撞仿真分析及性能优化[J]. 客车技术与研究, 2019, 41(4): 9-11.
- [9] 颜长征,王欣,赵东旭,等. 客车正面碰撞乘员保护分析[J]. 机械研究与应用, 2017, 30(3): 78-80.
- [10] 罗森,康忠喜,任妍婷. 基于全参模型驱动的某皮卡车架性能带宽研究[J]. 农业装备与车辆工程, 2024, 62(4): 50-55.
- [11] 陈祖兴,王运辉,程一明,等. 基于正面碰撞前防撞梁的DOE轻量化设计[J]. 机械工程师, 2022(9): 88-92.
- [2] 胡艳峰. GB/T 32960 标准分析及应用[J]. 汽车电器, 2018(9): 49-52.
- [3] 陈吉荣,乐嘉锦. 基于 Hadoop 生态系统的大数据解决方案综述[J]. 计算机工程与科学, 2013, 35(10): 25-35.
- [4] 崔杰,李陶深,兰红星. 基于 Hadoop 的海量数据存储平台设计与开发[J]. 计算机研究与发展, 2012, 49(增刊1): 12-18.
- [5] ZHOU H T, YAO Y, TU F. Research on the construction of the “hive big data warehouse” experimental teaching environment[C] //Proceedings of 4th International Conference on Education, Knowledge and Information Management, 2023: 409-416.
- [6] 代明竹,高嵩峰. 基于 Hadoop, Spark 及 Flink 大规模数据分析的性能评价[J]. 中国电子科学研究院学报, 2018, 13(2): 149-155.
- [7] 刘魁刚,张瑜,毕鹏,等. 城市轨道交通工程建设安全数据融合路径及方法[J]. 数字通信世界, 2024(1): 4-6.
- [8] 陶浩然. DataX 在工程数据管理平台中的应用[J]. 中国新通信, 2024, 26(1): 85-88.
- [9] KIM Y K, JEONG C S. Large Scale Image Processing in Real-Time Environments with Kafka[C] //Second International Conference on Computer Science, Information Technology and Applications, 2017: 207-215.
- [10] 王光,林国宇. 改进的自适应参数 DBSCAN 聚类算法[J]. 计算机工程与应用, 2020, 56(14): 45-51.