

# 面向预测性维护的工业设备管理系统



禹鑫燧 施甜峰 唐权瑞 殷慧武 欧林林

浙江工业大学信息工程学院 杭州 310023

(yuxinyinet@163.com)

**摘要** 为了解决制造业中工业设备管理混乱、维护成本高昂等问题,以工业机器人、数控车床等工业设备为管理对象,开发面向预测性维护的设备管理系统。基于 SpringBoot 框架和 Vue 前后端分离模式对系统进行开发,降低其耦合性;根据实际生产需求设计设备管理模块,实现设备基础信息和生产数据的管理;并开发人机交互良好的前端界面,达到设备信息可视化管理的目的;整合多种数据库设计数据存储模块,解决系统不同数据种类的读写问题;基于 Spark 大数据处理框架设计设备维护模块,对设备实时数据进行在线分析,并使用机器学习回归算法对历史数据进行预测模型训练,实现设备状态的实时监控与剩余使用寿命的预测,达到设备预测性维护的目的。最后,通过工业机器人设备实验验证了所设计的管理系统的可行性。

**关键词:** 设备管理;Spark;机器学习;大数据处理;预测性维护

**中图分类号** TP315

## Industrial Equipment Management System for Predictive Maintenance

YU Xin-yi, SHI Tian-feng, TANG Quan-ru, YIN Hui-wu and OU Lin-lin

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

**Abstract** An industrial equipment management system for predictive maintenance is developed to solve the problems of chaotic equipment management and high maintenance costs in the manufacturing industry. The system is developed based on SpringBoot framework and Vue front-end separation mode that the coupling is reduced; the equipment management module is designed according to the actual production that realizes the basic information and production data management of the equipment. A good front-end interface for human-computer interaction is developed to achieve the purpose of visual management of equipment information. The data storage module is designed through integrating multiple databases to solve the problem of reading and writing different types of data in the system. The equipment maintenance module is designed based on the Spark big data processing framework to perform online analysis of equipment real-time data. In order to achieve the goal of predictive maintenance of equipment, machine learning regression algorithms are used to train predictive models on historical data to achieve real-time monitoring of equipment status and prediction of remaining life. Finally, the feasibility of the designed management system is verified by industrial robot equipment experiments.

**Keywords** Equipment management, Spark, Machine learning, Big data processing, Predictive maintenance

## 1 引言

随着现代工业的迅猛发展,各国制造业创新战略都提出了智能制造这一理念<sup>[1]</sup>,它推动制造业的数字化、网络化和智能化发展<sup>[2-3]</sup>。制造执行系统(Manufacturing Execution System, MES)的出现为智能制造企业提供了精细的制造环境和生产过程中设备的可视化监控与数据管理<sup>[4-5]</sup>;同时,通过人工智能技术研发智能制造辅助方案<sup>[6]</sup>,并结合成熟的信息技术将其和制造执行系统融合,预测性维护便是其中的重要应用领域。在预测性维护产生之前,对工业设备的维护通常是常规性维护,主要以时间为单位进行周期性的保养,导致了资源浪费和设备损耗。预测性维护的产生改善了这一现状,其通过融合人工智能、数据挖掘、云计算等一系列技术,能在设备损坏之前合理预估设备部件使用剩余寿命,尽早预测设备

故障隐患,避免故障恶化,从而有效减少设备停机维修时间,降低设备维修成本<sup>[7]</sup>。正因为如此,预测性维护也被 CB insight 评为 2019 年人工智能发展趋势之一。

设备管理是 MES 系统中最基础的一环,主要包括设备的型号、参数等基础信息管理和电流、电压等运转数据管理,对设备的维护保养和生产安排具有重大参考价值。国内外有许多学者对设备管理系统的开发与应用进行研究,文献[8]提出一种通用的工业设备智能化管理系统开发及应用,重点解决设备维修方面的管理,降低维修成本;文献[9]基于 Zigbee 无线组网技术实现工业设备电源的远程管理,能够对设备电源进行实时监测和远距离控制启停;文献[10]开发了一套石油设备管理系统,基于 J2EE 平台和 EJB 多层结构开发思想,解决了石油企业存在的基础数据缺乏和不完善等问题。

从目前的研究成果来看,所设计的设备管理系统都只停

本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家重点研发计划(2018YFB1308400)

This work was supported by the National Key Research and Development Program of China (2018YFB1308400).

通信作者:欧林林(linlinou@zjut.edu.cn)

留在设备管理和监控层面,智能化水平不高,设备维护方式落后,采用传统的故障维修手段无法满足现代智能制造大背景下的制造企业需求。因此,随着智能制造的不断推进,开发一款智能化程度高,能结合人工智能等新兴技术实现预测性维护功能的设备管理系统,具有很大的应用价值。

基于上述背景和研究现状,本文在对现有的工业MES系统和设备管理系统现状分析的基础上,设计面向预测性维护的工业设备管理系统。本系统构建设备管理和设备维护两大模块,并根据系统的功能需求开发人机交互友好的浏览器/服务器(Browser/Server, B/S)模式的设备管理软件,最大程度地实现信息共享<sup>[11]</sup>,达到对工业设备的远程管理监控;同时,为了设备维护的及时性和准确性,系统基于Spark大数据处理框架<sup>[12]</sup>搭建设备维护模块,实现工业机器人、数控机床等多种设备生产过程中的实时状态监测和预测性维护;最后,搭建实验平台,通过实验对该设备管理系统的可行性和预测性维护的预测能力进行验证。

## 2 系统总体架构设计

面向预测性维护的工业设备管理系统,主要解决制造型企业设备管理模式落后、设备智能化维护程度较低等问题。基于模块分离的解耦原则,将本系统设计成部署在工厂的本地服务器和部署在具有公网IP的云端服务器两大块,同时采用配置方便和维护简单的B/S架构,将系统部署在广域网的云端服务器上。系统总体架构设计如图1所示。

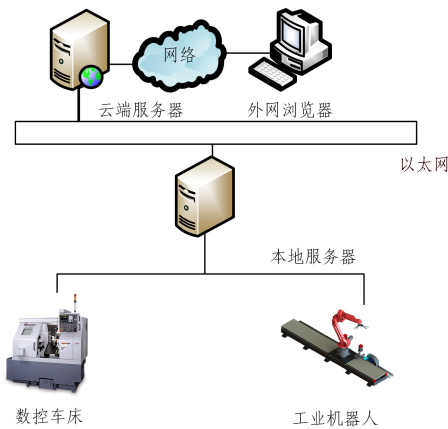


图1 系统总体架构设计  
Fig.1 Overall system design

本地服务器是建立在工厂实地的底层系统,连接远程互联网和工厂本地,主要作用是采集设备的实时运行数据,并通过工业互联网和消息组件转发至云端服务器进行存储;同时还需要接受云端服务器发送的设备控制指令,并将其解析给实际工业设备,达到用户远程控制的目的。系统所设计的云端服务器是该架构的核心,连接用户和底层工业设备,搭载了整个工业设备管理系统,用户通过外网浏览器和公网IP就可以访问云端服务器上的管理系统可视化界面,通过操作界面达到设备管理维护的目的。

以云端服务器为核心,对整个系统进行业务操作,并通过各类协议进行数据的收发,可以实现以下功能:1)对工业设备基础信息和状态信息的读取显示;2)对工业设备的紧急干预控制;3)对工业设备状态数据的智能分析;4)各个层级之间的数据交互。

云端服务器总体软件结构如图2所示,将软件结构设计为处理业务逻辑的SpringBoot框架、存储数据的企业信息系统和控制底层设备的工控系统三大模块。设计的SpringBoot框架内部业务逻辑对象可划分为设备管理模块和设备维护模块,主要与企业信息系统中的数据库模块进行交互,其数据交互体系结构如图3所示。

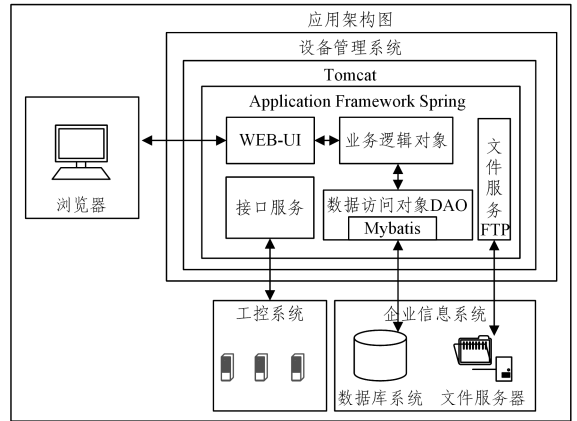


图2 云端服务器软件结构  
Fig.2 Cloud server software structure

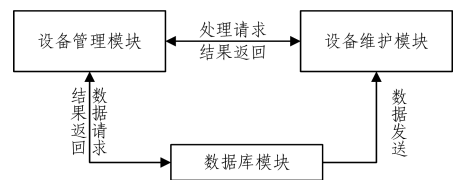


图3 模块数据交互  
Fig.3 Module data interaction

设备管理模块接受前端网页的请求之后对其进行处理,对于数据读取请求,只需要对数据库模块进行操作,数据库模块会返回或写入相应数据;而对于设备维护查询请求,则需要调用设备维护模块,该模块会根据请求内容监测当前设备运行数据,获取数据库模块中的设备生产数据进行模型训练分析,并将分析处理结果同步地返回到设备管理模块中。

## 3 设备管理模块设计

设备管理模块采用SpringBoot架构进行搭建,可以在IntelliJ IDEA 2018.1.5 64位版本平台上进行快速开发,主要编程语言为Java。根据松耦合原则,将架构设计为显示层、控制层、业务层和数据层,如图4所示。

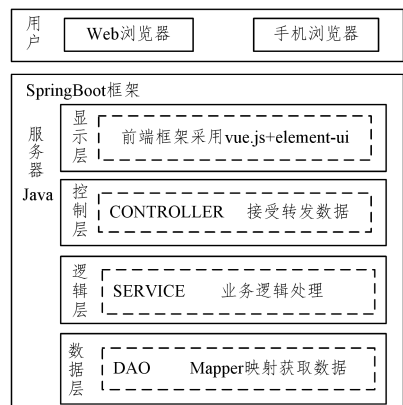


图4 SpringBoot架构  
Fig.4 SpringBoot architecture

为了提升可视化界面的交互性能和前后台的解耦性能,设计了主流的 vue.js + element-ui 前后端分离架构来编写显示层架构;基于 MVC 模式设计后台控制层和逻辑层,将数据与视图相互之间的影响降到最低;基于 MyBatis 持久层框架搭建系统数据层,将代码与数据库查询语句分离,提高系统可维护性。当浏览器发送 HTTP 请求至后台控制层时,控制层将请求处理后转发至逻辑层,逻辑层进行对应的业务处理并通过数据层 DAO 获取数据库中的数据,将获得的数据返回给控制层,控制层再将结果返回至界面显示。

设备管理模块能向系统用户提供一系列的业务功能。

(1)设备状态实时监测

传统的设备实时监控主要采用摄像头观察设备的开关状态,缺乏对设备内部运行状态的展示,可靠性和准确性相对不高。而在本系统的设备状态实时监测当中,本地服务器在获取工业设备数据之后,云端服务器状态实时监测界面直接通过系统搭建的 WebSocket 传输长连接将数据提取至界面进行显示;并且二次开发 Echarts 图表组件,通过图表等方式为用户提供被管理设备的实时运行状况和运行数据。

(2)设备数据报表生成

设备实时生产数据可以按照时序在数据存储模块中进行存储,形成历史数据集。本地服务器将获得的数据利用 MQTT 消息组件存储到云端服务器中的 Elasticsearch 数据库,设备数据报表生成界面通过查询数据库来获得所需数据。为了提升数据的可视化程度,本单元结合百度开源 Echarts 图形组件,用饼状图、折线图的形式直观地展示设备在一段时间内的运行情况。生成的设备数据报表可以为用户提供合理的设备使用改善建议,以提高设备的使用效率。

(3)设备报警干预控制

传统的设备管理虽然系统也存在设备的异常报警,但往往需要人工前往工厂实地解除警报,耗费人力,甚至等到设备真正产生故障停机之后才会触发报警,这样会导致设备不可逆转的损伤。在本系统的设备报警干预控制中,通过用户自行设置的报警阈值参数,或者通过设备维护模块返回的实时监测和维护信息,可以实现及时的设备干预,主要方法是通过互联网和消息组件向工厂本地服务器发送设备工作指令,用户可以在远程对设备进行干预控制。

4 数据库模块设计

本系统中数据格式种类多样,可以分为热数据、普通关系型数据、工业大数据和文件数据四大类:1)热数据是指可视化界面经常需要展示的数据,例如设备维护模块分析获得的处理结果等,都需要先缓存在缓存数据仓库中,然后再持久化到普通关系型数据库中;2)普通关系型数据是指设备的品牌、型号等设备基础信息,还包括缓存数据仓库持久化的数据;3)工业大数据是指本地服务器发送到云端服务器的各类设备运行数据,通常为标准化 JSON 格式,由于数据量庞大,需要存到大数据存储仓库中;4)文件数据是指设备维护过程中产生的照片和音视频文件。

对于热数据和普通关系型数据,数据库采用 MySQL + Redis 的设计来实现数据的读写操作。利用高性能的 NoSQL-Redis 作为 MySQL 的缓存,能够提升系统整体的读取速度。读取数据时,热数据都需要经过 Redis 进行缓存获取,而数据访问不频繁的数据(冷数据)都直接从 MySQL 中获

取,数据库查询数据操作逻辑如图 5(a)所示。写入数据时,传统方法都是先将数据写入 MySQL 后手动同步到 Redis 中,但是在高并发场景下由于数据读写速度不一致会造成数据读写不同步,因此本系统在 Redis 端采用 Canal 解析 MySQL 的 binlog 文件的方法。Canal 能快速地自动进行 MySQL 和 Redis 的写入同步,满足可能产生的高并发应用场景。数据库写入数据的操作逻辑如图 5(b)所示,Canal 解析 MySQL 的增量日志,模拟 MySQL-Slave 的交互协议,向 MySQL-Master 发送 dump 协议,MySQL-Master 收到 dump 请求后开始推送 binlog 给 Canal,Canal 解析 binlog 对象并发给 Redis,Redis 根据 binlog 和 MySQL 同步,这样 Canal 就完成了两者的数据同步操作。

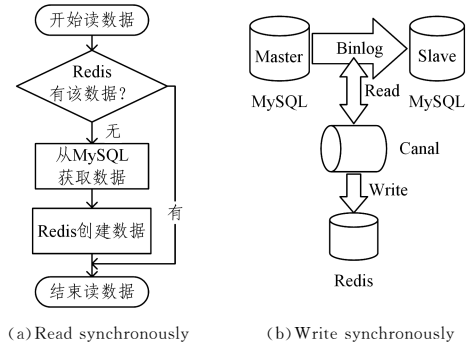


图 5 MySQL 和 Redis 的读写同步

Fig. 5 Sync MySQL and Redis

本系统除了基于 MySQL 数据库存储关系型数据外,还采用 Elasticsearch 分布式存储系统存储海量工业生产数据。Elasticsearch 存储系统可以在极短时间内存储、搜索和分析 PB 级数据,并且基于 RESTful 接口,方便 SpringBoot 程序的接口调用。该存储系统的概念名词和普通的关系型数据库 MySQL 的对比如表 1 所列<sup>[13]</sup>。

表 1 MySQL 和 Elasticsearch 概念对比

Table 1 Comparison of MySQL and Elasticsearch concepts

MySQL	ElasticSearch
DataBase	Indices
Table	Type
Row	Document
Column	Field

设备管理和维护过程中产生的数据文件,例如检修过程中的图片、音频、视频等数据文件,都将在 FTP 服务器端的文件夹仓库进行存储。用户通过前端界面可以对 FTP 服务器进行文件的存储与读取,以备之后的设备保养维护使用。

5 设备维护模块的设计

设备维护模块的实现主要是通过通过对设备生产过程中的电流、电压等数据进行处理与分析来达到设备维护目的,因此本系统结合实际生产设备,搭建了数据实时监控和模型预测的大数据处理框架。本文所设计的大数据处理框架基于 Apache 公司的 Spark 项目,该项目是专为大规模数据处理而设计的快速通用的计算引擎,它启用内存分布式数据集,能更好地适用于类似机器学习等需要迭代的 MapReduce 算法<sup>[14]</sup>。本系统主要用到 Spark 框架中的 Spark-Streaming 和 Spark-MLlib,即流式数据分析和机器模型训练预测。本系统的大数据处理架构如图 6 所示,工业设备上的生产数据同时

流向两部分,一是 Spark-Streaming 的实时流式数据分析,二是存储在 ElasticSearch 数据库中等待 Spark-MLlib 模型训练的调用,最后的数据处理结果都可以存储在 Redis 缓存中。

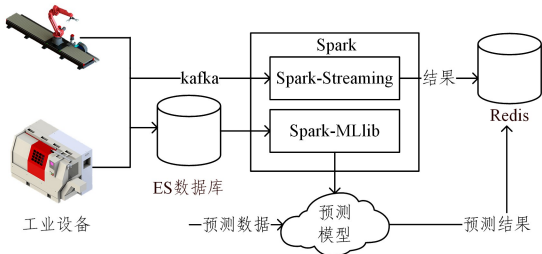


图6 大数据处理架构

Fig. 6 Big data processing architecture

### 5.1 流式数据处理分析

工业机器人、数控机床等工业设备上的底层通信代码 C# 会通过消息中间件 Kafka 进行相关主题的数据消息发布,Spark-Streaming 从 Kafka 订阅相应的数据进行实时计算处理。采用滑动窗口法实现数据的收取,在该窗口时间内的连续数据会被缓存成 RDD 弹性数据集,然后经过统一的 Spark 算子处理,来决定是否要发出警报。当这一段时间的数据处理完成后,数据全部清除,重新开始缓存新的时间窗口数据。如果出现警报,警报数值的相关信息则会存入 Redis,并且通过建立的 WebSocket 前后台连接直接向可视化界面发送报警故障信息。流式数据阈值筛选处理流程如图 7 所示。

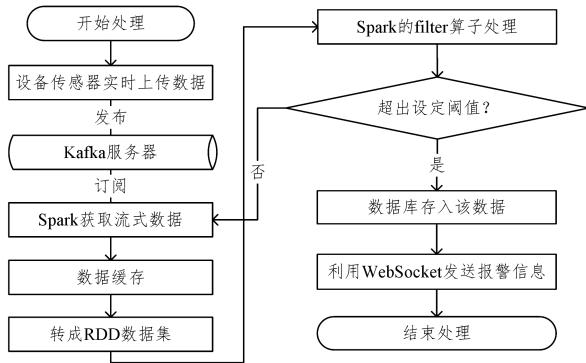


图7 流式数据阈值筛选流程

Fig. 7 Streaming data threshold filtering process

### 5.2 预测模型训练分析

机器学习模型预测是该架构的核心,是工业大数据技术的主要体现。为了区别于普通数据并提升数据获取速度,该模块采用单独的分布式文档数据库 ElasticSearch,它可以在极短的时间内存储、搜索和分析大量的数据。机器学习预测模型的训练流程如图 8 所示,当数据库 ElasticSearch 中的数据达到设定阈值数量时,将该数据集进行数据清洗、补全等预处理操作之后,提取数据特征值,选择合适的机器学习回归算法,建立特征模型,再用处理完后的数据对模型进行训练。为了使模型更加贴合实际,防止产生过拟合现象,将数据集分成 70% 的训练集和 30% 的测试集,通过测试集评估预测模型的准确性。该模型可以对实时数据进行设备状态预测,返回设备零部件剩余使用寿命,从而达到预测性维护的目的。服务器向 Spark-MLlib 模块发送需要预测的数据,通过训练模型

的预测方法即可获得数据预测结果,并存入 Redis 中。

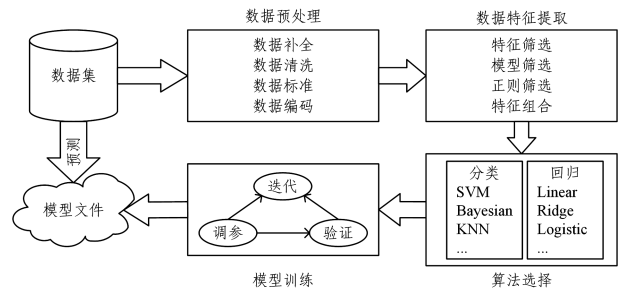


图8 离线模型预测流程

Fig. 8 Offline model prediction process

本模块选取的机器学习算法之一就是线性回归算法,该算法是当前工业界比较常用的机器学习算法。线性回归在本系统中就是通过对历史数据中每条数据进行特征值提取,计算得出每个特征值对应的权重。特征值与权重的乘积再加上偏置值,就是模型所预测的设备剩余使用寿命。该算法公式如下:

$$y = \omega_1 \times x_1 + \omega_2 \times x_2 + \omega_3 \times x_3 + \dots + \omega_n \times x_n + b$$

其中,  $n$  代表特征数目,定义  $\omega_0 = b, x_0 = 1$ , 则公式转换为:

$$y = \omega_0 \times x_0 + \omega_1 \times x_1 + \omega_2 \times x_2 + \dots + \omega_n \times x_n$$

假设有  $m$  个样本,写成矩阵形式为:

$$x = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \dots & x_n^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_1^m & x_2^m & \dots & x_n^m \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}$$

$$\omega = [\omega_0 \quad \omega_1 \quad \omega_2 \quad \dots \quad \omega_n]$$

公式就简写成:

$$y = xw^T$$

损失函数为样本预测值  $\hat{y}_i$  与样本实际值  $y_i$  的均方误差:

$$cost = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

在 Spark-MLlib 中,该线性模型被设计为带有梯度下降算法,使其在学习训练过程中能逐渐减小损失函数  $cost$  的值,使预测值逐渐逼近实际值。

因此在本系统中,当预测模型训练完毕时,用户只需要在设备预测性维护界面输入实际系统的自变量  $X$  值,系统经过后台的预测模型计算,就能返回设备的剩余使用寿命  $Y$ ,并且可以设定使用寿命阈值,在零至该阈值区间内,系统就会产生报警信息。

以训练工业机器人预测模型为例,基于 Spark 架构在 IDEA 编程软件平台上开发的步骤流程图如图 9 所示。1) 在工程项目管理类工具 Maven 中导入 Spark-MLlib 的相关 jar 包;2) 首先在集成开发环境中创建用于选择 Spark 机器学习算法库中算法的 MLSelect 类;3) 前端页面用户以 txt 文本形式上传工业机器人历史数据集和选择相应算法名称,后台接受数据集并解析存放在项目资源对应用户下的文件夹中;

4) 通过 Java 反射机制获取 MLselect 类对象和相应训练方法,对上传的数据集进行训练后得到预测模型,调用 Spark 中的 model.save 或者 model.toPMML 方法,根据不同场景需要,将模型存储为模型文件或者预测模型标记语言(Predictive Model Markup Language, PMML)形式的配置文件。

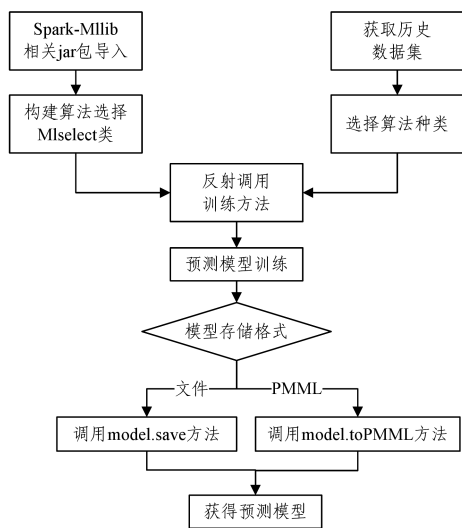


图 9 模型训练开发流程

Fig. 9 Model training development process

存储格式为模型文件的模型只能在 Spark 框架下进行调用,但是预测精度高,优化性能好,最符合训练要求;而将模型存储为标准的 PMML 文件形式,优点在于各种不同的开发语言都可以导入该文件做线上预测,不受框架制约,兼容性好,但该格式会牺牲独有的优化性能<sup>[15]</sup>。本系统设计采用 Scala 语言做模型训练,采用 Java 语言载入模型做预测,可以使用第二种方法,但对于 Spark 框架和 PMML 语言来说,目前还不支持所有的模型都以 PMML 格式进行导出存储,所以本系统的预测模型采用 PMML+模型文件两种格式兼用的存储形式。

模型训练成功之后,就可以进行数据预测处理。前端查询项目资源中模型数量和名称并显示在模型选择的下拉框中供用户选择,用户选择模型并输入所需预测数据,提交到后台。对于模型文件,采用 model.load 方法载入预测模型,然后调用 model.predict 方法并写入预测数据,得到预测结果;而对于 PMML 文件,先将该文件转成二进制流,然后再通过 unmarshal 方法进行格式转化,转化成可以进行预测的非文本格式,并且根据 PMML 文件内部的数据量输入相应数据,得出目标 target 的值,该值即为预测结果。

## 6 实验验证

本文根据所设计的面向预测性维护的工业设备管理系统,搭建设备管理和设备维护实验系统。实验系统包括 1 台工业 UR5 工业机器人设备、1 台连接外网的交换机、1 套底层 PLC 控制系统、1 台具有公网 IP 的云端服务器、1 台用户电脑和 1 个具有云传输功能的工业摄像头。

### 6.1 设备管理实验

首先在云端服务器上搭建该 Web 项目所需要的环境,包括 Java 运行环境、数据库 MySQL、缓存 Redis、分布式数据库

ElasticSearch、FTP 服务器、Web 应用服务器 Tomcat、消息传输 MQTT 协议服务器 mosquito,并且进行简单的网页发布实验,验证云服务器的连接性能是否良好。

将本地搭建的 SpringBoot 架构用项目管理工具 Maven 进行打包操作,根据项目管理配置文件 pom.xml 中的配置参数生成项目的压缩 war 包。将该 war 包拷贝到云端服务器下的 Web 应用服务器 Tomcat 的 webapps 文件夹中,运行 Tomcat 启动命令, Tomcat 将项目 war 包进行解压,完成项目的发布,从而可以通过公网 IP 对该文件夹下的资源文件进行访问,达到外网浏览的目的。

通过输入公网 IP、端口号加项目内文件名访问本监控系统登录首页,输入用户名和密码之后进入系统首页,显示用户设备数量、报警信息和分布情况;设备列表展示设备连接状态、设备名称、操作人员等信息,还可对设备进行添加、删除或修改等操作;数据图表可以对工业机器人的电压、温度等生产数据进行查询,Echarts 图标组件对数据进行折线图展示,并且能通过选择显示对象进行显示图表切换;设备报表可以对过去 24 小时之内的设备运行状态进行报表显示与导出;设备控制是当设备发生运行故障时,工程师可以通过网络进行异地控制,借助高清的工业摄像头的视频云传输,观察设备控制情况,直至报警解除,设备恢复工作状态。各个功能模块显示界面如图 10 所示。

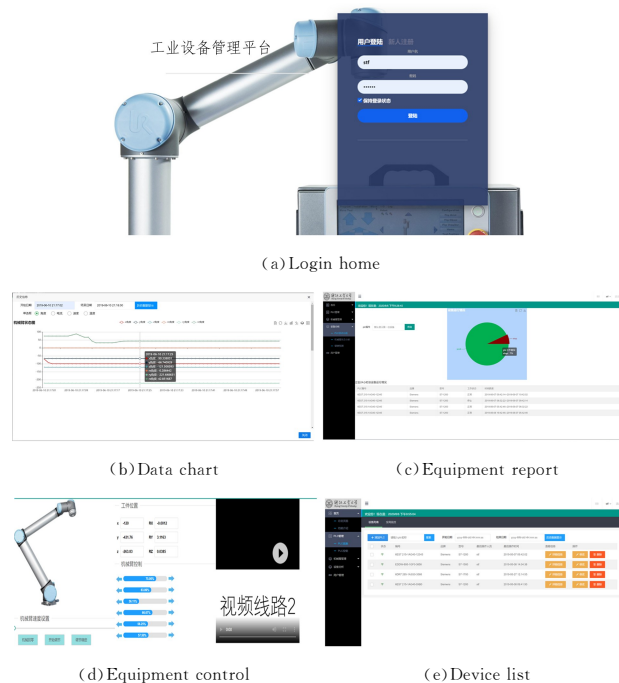


图 10 系统界面

Fig. 10 System interface

### 6.2 数据处理实验

#### 6.2.1 实时数据处理

借助工业网关,本地服务器可以将设备实时的状态数据发送到 Kafka 的 Broker 服务器内的对应主题, Spark-Streaming 订阅相应主题并连续接收主题内的数据进行实时处理。由于 Kafka 是分布式架构,首先需要打开服务器主机上的分布式应用协调服务 Zookeeper,用于注册管理 Kafka 的 Broker 服务器,且同一个 Topic 的消息会根据 Zookeeper 的负载均衡

