

杭州电子科技大学

硕士学位论文

题 目: 光伏电站远程运维系统的
设计与实现

研 究 生 徐志杰

专 业 电子与通信工程

指导教师 陈科明 副教授

企业导师 曾佳

完成日期 2022 年 05 月

杭州电子科技大学硕士学位论文

光伏电站的远程运维系统设计
及其关键技术研究

研究生：徐志杰

指导教师：陈科明 副教授

2022 年 05 月

**Dissertation Submitted to Hangzhou Dianzi University
for the Degree of Master**

**Design and key technology research of remote
operation and maintenance system for
photovoltaic power station**

Candidate: Zhijie Xu

Supervisor: Associate Prof. Keming Chen

May, 2022

杭州电子科技大学 学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：徐志全 日期：2022年5月31日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：徐志全 日期：2022年5月31日

指导教师签名：陈利明 日期：2022年5月31日

摘要

随着光伏能源在我国能源结构占比中的逐年剧增，如何智能高效地运维光伏系统已成为行业的一大热门问题。伴随物联网及机器学习技术的发展，人工维护光伏系统的场景逐步被机器人和算法替代。光伏电站的运维主要体现在智能清洁、功率预测、热斑检测三个方面，工业界产生了许多光伏电站系统的运维方案，但大部分系统存在如下缺点或不足：清洁低效，存在人力清洁；光伏输出功率的预测时间较长，算法较复杂；热斑检测方面则存在较高的漏检率，无法有效地区分热斑和虚警。

本文根据目前行业内光伏系统的需求和不足的现状，设计了一种光伏电站的远程运维系统。系统具备智能清洁、功率预测、热斑检测、设备远程更新等功能。其中硬件部分由以 ARM Cortex-M3 为内核的控制板和通信板组成。控制板实现机器人的清洁控制、数据采集等功能，通信板则主要实现边缘终端和云端的数据通信。系统通过霍尔传感器和控制板实现了单排机器人的智能清洁控制。运维系统拥有以微服务架构为基础的后端功能群，包含了功率预测、热斑检测、远程更新、数据分析等多个功能。

光伏的功率预测常用于提高电网的稳定性，同时能够反映光伏系统的健康状态。本文通过控制板采集环境数据来作为功率预测的数据集。在研究多种光伏功率预测方案后，本文提出了一种基于区间划分的 K 最近邻 (K-Nearest Neighbor, KNN) 短期光伏功率预测算法，通过计算实时数据和历史数据的欧式距离得到预测功率对应的输出区间，并以区间内前 K 个值的加权平均作为预测输出。该方法实现简便，具备快速的接口响应速度，算法的平均绝对百分比误差达到 7.83%，均方根误差达到 0.5053。

功率输出反映了光伏系统的健康状况，而造成光伏电池受损的主要因素则是光伏板的热斑效应。本文设计并实现了一种基于 XGBoost 优化 Flood Fill 的热斑检测算法。边缘终端通过红外成像仪获取红外图像，经过灰度变换后使用 Flood Fill 识别热斑的形态，并通过 XGBoost 算法进行热斑形态的学习。最终测试表明本方法识别热斑的精确率较高，并且能够有效地减少非热斑光源造成的误判，热斑识别的精确率达到 92.31%。

关键词：光伏机器人、KNN、微服务、热斑检测、XGBoost

ABSTRACT

With the sharp increase of photovoltaic energy in China's energy structure year by year, how to intelligently and efficiently operate and maintain photovoltaic system has become an important topic in this filed. With the development of Internet of Things and machine learning technology, the scene of manual maintenance of photovoltaic system is gradually replaced by robots and algorithms. The operation and maintenance of photovoltaic power station is mainly reflected in three aspects: intelligent cleaning, power prediction and hot spot detection. The industry has produced many operation and maintenance schemes of photovoltaic power station system, but most systems have the following shortcomings or deficiencies: low efficiency of cleaning and manual cleaning; The prediction time of photovoltaic output power is long and the algorithm is complex; In terms of hot spot detection, there is a high missed detection rate, which can not effectively distinguish hot spot from false alarm.

According to the current situation of the demand and deficiency of photovoltaic system in the industry, this thesis designs a remote operation and maintenance system of photovoltaic power station. The system has the functions of intelligent cleaning, power prediction, hot spot detection, remote equipment update and so on. The hardware module uses arm Cortex-M3 as MCU of control board and communication board. The control board is mainly used to realize the cleaning control and data acquisition of the robot, and the communication board mainly realizes the control of the edge end and the data communication of the cloud. The intelligent cleaning control of single row robot is realized by Hall sensor and control board. The operation and maintenance system has a back-end function group based on microservice architecture, including power prediction, hot spot detection, remote update, data analysis and other functions.

The prediction of photovoltaic power is helpful to avoid the fluctuation of grid connection and reflect the health state of photovoltaic system. In this thesis, the environmental data collected by the control board is used as the data set of power prediction. After studying various photovoltaic power prediction schemes, this thesis proposes a K-nearest neighbor (KNN) short-term photovoltaic power prediction algorithm based on interval division. The output interval corresponding to the predicted power is obtained by calculating the Euclidean distance between real-time data and historical data, and the weighted average of the first K values in the interval is used as the prediction output. This method is easy to implement and has fast interface response speed. The MAPE of the algorithm

reaches 7.83% and RMSE reaches 0.5053.

The power output reflects the health of the photovoltaic system, and the main factor causing the damage of photovoltaic cells is the hot spot effect of photovoltaic panels. This thesis designs and implements a hot spot detection algorithm based on XGBoost optimized Flood Fill. At the edge, the infrared image is obtained by the infrared imager. After gray transformation, the shape of hot spots is recognized by Flood Fill, and the shape of hot spots is learned by XGBoost algorithm. The final test shows that this method has a high accuracy in identifying hot spots, and can effectively reduce the misjudgment caused by non hot spot light sources. The accuracy rate of hot spot identification is 92.31%, the missed detection rate is less than 8%, and the false detection rate is less than 9%.

Keywords: Photovoltaic robot, KNN, microservice, hot spot detection, XGBoost

目录

摘要.....	I
ABSTRACT.....	II
目录.....	IV
图清单.....	VII
表清单.....	IX
第 1 章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	3
1.2.1 光伏清洁机器人的研究.....	3
1.2.2 光伏板功率预测的研究.....	3
1.2.3 光伏板热斑检测的研究.....	4
1.2.4 光伏电站的后端模式研究.....	4
1.3 光伏电站运维系统的现有不足.....	5
1.4 论文结构安排.....	5
第 2 章 光伏电站的远程运维系统设计.....	7
2.1 系统总体设计.....	7
2.1.1 需求分析.....	7
2.1.2 总体架构.....	8
2.2 边缘终端设计.....	9
2.2.1 整体设计.....	9
2.2.2 控制模块设计.....	10
2.2.3 热斑识别模块设计.....	12
2.2.4 网关模块设计.....	12
2.3 云平台.....	17
2.4 后端架构.....	18
2.4.1 后端架构选型.....	18
2.4.2 后端业务架构.....	19
2.4.3 后端系统架构.....	20
2.5 服务设计.....	22

2.5.1 基础服务设计	22
2.5.2 业务服务设计	27
2.5.3 分析服务设计	32
2.6 应用层设计	32
2.6.1 WEB 前端设计	32
2.6.2 小程序设计	33
2.6.3 系统展示	33
2.7 本章小结	35
第 3 章 基于区间划分的 KNN 光伏功率预测设计	36
3.1 整体选型	36
3.2 KNN 介绍	38
3.3 数据预处理	39
3.3.1 数据获取	39
3.3.2 相关性分析	40
3.3.3 数据归一化处理	42
3.4 KNN 预测光伏功率的实现	43
3.4.1 算法总体实现	44
3.4.2 最优区间划分的实现	45
3.4.3 结果输出实现	47
3.5 模型测试与评估	48
3.5.1 网格搜索法优化 K 值	48
3.5.2 KNN 预测结果与评估	49
3.6 本章小节	51
第 4 章 基于 XGBoost 优化的 Flood Fill 热斑检测设计	52
4.1 光伏组件热斑形成的原理	52
4.1.1 光伏组件的热斑效应	52
4.1.2 热斑形成的原理	52
4.2 基于红外成像的热斑检测方法	53
4.3 Flood Fill 的热斑检测方法实现	55
4.3.1 Flood Fill 算法介绍	55
4.3.2 Flood Fill 解决热斑问题	56
4.3.3 Flood Fill 进行热斑检测的测试	59
4.4 Flood Fill 优化的热斑检测方法实现	60

4.4.1 XGBoost 算法.....	60
4.4.2 数据处理.....	62
4.4.3 XGBoost 的参数选择和构建.....	63
4.5 模型的测试与评估.....	64
4.5.1 模型的评估指标.....	64
4.5.2 模型测试.....	64
4.6 本章小节.....	66
第 5 章 总结与展望.....	67
5.1 总结.....	67
5.2 展望.....	67
参考文献.....	69
附录.....	74

图清单

图 1.1 2012-2020 年我国光伏发电量及增速情况.....	1
图 1.2 2015 与 2020 年新增电力装机结构对比	1
图 1.3 Ecoppia 便携式光伏清洁机器人.....	3
图 2.1 系统总体框架图.....	8
图 2.2 光伏板与清洁机器人.....	9
图 2.3 边缘终端整体示意图.....	10
图 2.4 电机控制示意图.....	10
图 2.5 控制板工作示意图.....	11
图 2.6 网关与数据采集模块组成示意图.....	12
图 2.7 网关软件结构图.....	14
图 2.8 帧类型示意图.....	14
图 2.9 数据上报示意图.....	15
图 2.10 远程更新状态转移图.....	16
图 2.11 远程更新原理.....	17
图 2.12 设备云平台界面.....	18
图 2.13 设备运设备在线展示.....	18
图 2.14 系统设计架构图.....	19
图 2.15 系统业务架构图.....	20
图 2.16 系统架构图.....	21
图 2.17 系统缓存架构图.....	22
图 2.18 故障恢复示意图.....	24
图 2.19 远程更新基本流程图.....	25
图 2.20 远程更新主任务时序图.....	25
图 2.21 并行请求优化时序图.....	26
图 2.22 服务端业务数据获取流程图.....	30
图 2.23 雨水控制示意图.....	30
图 2.24 定时控制流程图.....	31
图 2.25 集中式光伏集群.....	33
图 2.26 光伏运维系统硬件展示图.....	34

图 2.27 WEB 端光伏机器人远程运维总览界面	34
图 2.28 WEB 端远运行日志展示	34
图 2.29 WEB 端电流监控记录	35
图 2.30 小程序端展示图.....	35
图 3.1 光伏板倾斜角及摆放对受光的影响	37
图 3.2 KNN 的距离函数	39
图 3.3 数据集生成流程图.....	40
图 3.4 辐照度、温湿度、风速和光伏输出的关系.....	41
图 3.5 光伏输出和光伏板倾斜角度关系.....	41
图 3.6 关键特征与功率的相关性热力图.....	42
图 3.7 KNN 算法预测输出功率框图	44
图 3.8 KNN 对光伏输出的分类区间	44
图 3.9 KNN 对光伏输出的归类示意图	45
图 3.10 步长划分流程图.....	46
图 3.11 KNN 结果计算流程图	47
图 3.12 K 值的网格搜索统计图	49
图 3.13 不同倾斜角下的 KNN 光伏输出预测对比图	50
图 3.14 在倾斜角为 30° 情况下的 KNN 光伏输出预测对比图.....	51
图 4.1 红外成像显示下光伏板上的热斑.....	52
图 4.2 Bishop 模型模拟光伏电池板反向特性	53
图 4.3 热斑检测获取灰度矩阵过程.....	54
图 4.4 传统识别方法误判光源虚警.....	54
图 4.5 XGBoost 优化的 FloodFill 热斑检测流程图	55
图 4.6 光伏板区域的灰度图矩阵.....	55
图 4.7 Flood Fill 算法的基本流程图	56
图 4.8 灰度矩阵的频率统计和变换率图.....	57
图 4.9 光伏板区域的灰度图矩阵.....	58
图 4.10 Flood Fill 算法填充热斑区域.....	58
图 4.11 经典场景对比.....	59
图 4.12 含虚警场合对比.....	59
图 4.13 特征值获取.....	62
图 4.14 特征值获取流程图.....	63
图 4.15 虚警场合对比.....	65

表清单

表 2.1 当前常用无线通信技术.....	13
表 2.2 不同缓存相关特点.....	23
表 2.3 远程更新核心接口.....	24
表 2.4 光伏管理系统核心模块及其功能.....	27
表 2.5 基础服务核心数据库及其功能表.....	28
表 2.6 分析预测服务核心数据库及其功能表.....	29
表 2.7 业务核心数据库及其功能表.....	29
表 3.1 时间序列算法优缺点对比.....	38
表 3.2 传感器功能选型表.....	40
表 3.3 光伏输出功率 KNN 算法预测输出对比	49
表 3.4 KNN 与 BP 神经网络的 MAPE 和 RMSE 对比	51
表 4.1 Flood Fill 热斑检测测试结果	56
表 4.2 热斑特征.....	62
表 4.3 XGBoost 参数选择.....	63
表 4.4 混淆矩阵格式.....	64
表 4.5 实验结果的混淆矩阵.....	65
表 4.6 热斑检测参数对比.....	65

第1章 绪论

1.1 研究背景和意义

根据 2021 年中国能源报告以及国家统计局给出的能源发展数据，综合图 1.1 可得：2020 年光伏新能源发电达到 2611 亿千瓦时，相对 2019 年比增加了 16%^[1]。

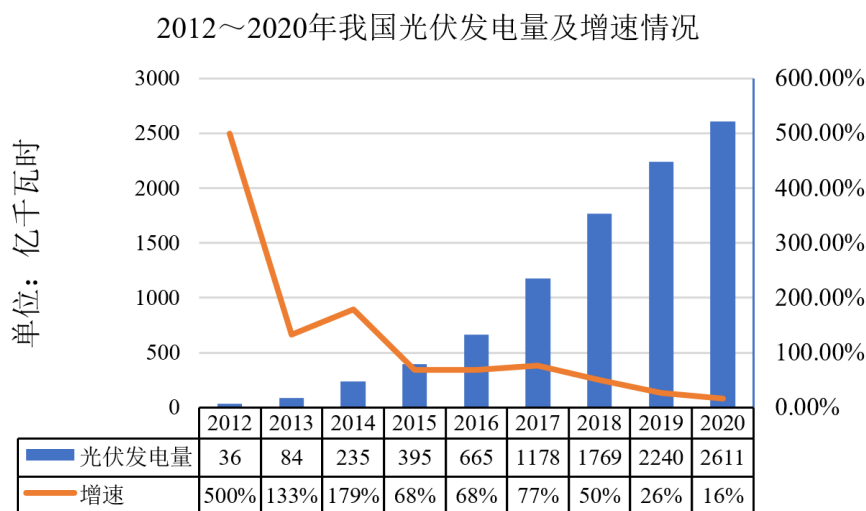
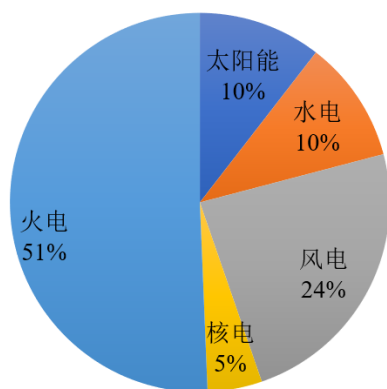


图 1.1 2012-2020 年我国光伏发电量及增速情况

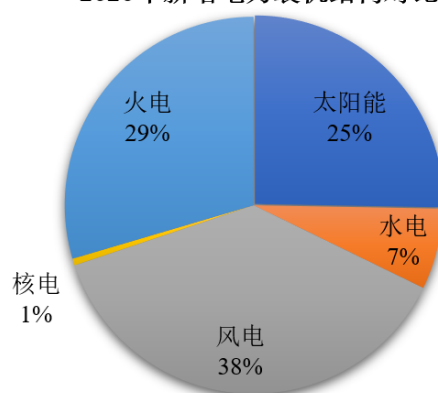
随着社会的进步以及人类用电需求的剧增，无污染的清洁能源的使用开始逐年增长，光伏能源作为一种新型环保能源，代替火电、水电等其他传统发电这一趋势完全符合我国可持续发展战略。光伏发电凭借着它能源的高效性以及排放的无污染性等优点得到了企业和国家的关注^[2]。图 1.2 为 2015 年和 2020 年新增电力装机占比统计图，图中表明光伏发电在所有发电场合中的占比逐渐增大，而基于光伏发电的生态环境和应用场景也在逐步完善。

2015年新增电力装机结构对比



(a) 2015 年

2020年新增电力装机结构对比



(b) 2020 年

图 1.2 2015 与 2020 年新增电力装机结构对比

光伏发电的主要场合可以分为两类，一类是小规模下的分布式发电场景，这种场景主要用于散户，多用于房车、住房等用户端为主的场合，有着低污染、低噪声等优点^[3]；另一类是本文主要研究的场合：集中式光伏电站，这种场合适用于集中式的大批量发电场景。大部分集中式光伏电站应用于戈壁、沙漠、大棚等环境，这给光伏面板的清洁运维造成诸多不利因素。集中式光伏发电多存在于我国的西部地区，因为西部地区人烟稀少，海拔高度高，降水少，日照时间相比东部地区长，这些因素天然的适合集中式光伏发电的使用^[4]。

随着物联网、机器学习、大数据等技术的发展，企业希望对光伏电站进行数字化的管理和操控，以及对光伏电站进行智能化运维。电站智能运维的一个重要方面就是光伏板的清洁，在光伏电站的清洁方面使用机器人代替传统人工清洁不但节省了人力成本，并且在清洁力度上相比更加的到位。在机器人控制方面，物联网技术带来的万物互联可以使得操作人员通过网络远程控制清洁机器人。

地球表面的颗粒碎屑、工厂排放的污染物、自然环境下的沙粒等这些灰尘颗粒覆盖于光伏板表面，都会导致光伏系统发电量下降。采用清洁机器人对集中式光伏电站的组件进行清洁，对比清洁前后的效果可以发现：大气环境良好时，该电站的日发电量可提升近 20%，大气污染严重时，电站日发电量提升近 40%^[5]。由此可见，组件的清洁度对发电量的影响至关重要。

随着光伏集群规模的增大，如何保障集群高可靠地运行已经称为行业内的痛点。影响光伏集群可靠性的因素主要有：热斑效应带来的光伏组件性能的下降以及间歇性功率输出带来的并网问题。热斑会使得光伏系统整体的使用寿命缩短 30%，严重的热斑更会导致周围电子器件的损害继而导致整个系统的功率输出降低。热斑发现不及时会使得热斑产生的高温过长，甚至引发严重的火灾。光伏功率的预测也是保证系统可靠性的重要目的，主要体现在三个方面：（1）在预测方法精确度很高的情况下，预测功率和实际功率的误差可以反映光伏系统的健康程度；（2）保证电网安全稳定运行；（3）调度管理光伏的并网。

针对上述背景，如何有效地对光伏集群中的热斑进行智能化地识别以及光伏功率地准确预测就很具有重要意义。在机器学习快速发展的当下，产业希望通过光伏系统的环境数据来预测整体系统的输出功率，以及通过相关算法有效检测出光伏板的热斑。

一个完整的光伏电站运维系统需要具备智能清洁和维护预测的能力。其中光伏清洁机器人保障光伏板的智能清洁，功率预测和热斑检测实现维护预测能力。为保障系统高效可靠运行还需保证清洁机器人的正常工作、热斑的有效识别、输出功率的有效预测以及高可用的后台系统^[6]。由此可见，一个具备分析功率产出、热斑检测、远程运维的智能化光伏电站运维系统是非常重要的。

1.2 国内外研究现状

1.2.1 光伏清洁机器人的研究

光伏板表面的灰尘积累会显著地影响光伏系统的功率输出，由于光伏集群常常布置于戈壁滩等宽旷的恶劣环境。随着时间的积累，光伏板上必定会积累一定量的灰尘，从而导致光伏系统的输出受到影响而降低。实际上，当灰尘厚度达到 $3\text{mg}/\text{cm}^2$ 时，阳光的透射会衰减到 50%^[7]。故定期清理光伏板就显得格外重要。

光伏板的智能清洁是采用机器人来代替传统的人工清理。机器人从形态和原理上总共分为三类：单排、跨排和便携式清洁，其中单排机器人的技术在国内较为成熟。单排清洁机器人以南京天创为主要代表，南京天创使用基于固定支架的光伏电站清扫机器人已被广泛商用^[8]。

而便携式机器人凭借着其优秀的便携性在国外光伏运维场景中广泛应用，如图 1.3 为以色列公司 Ecoppia 通过横向移动支架和纵向移动机器人达到多方位高效清洁光伏板的效果^[9]。Ecoppia 采用超细纤维毛刷旋转并使用气流吹扫为辅来实现表面灰尘的清理。



图 1.3 Ecoppia 便携式光伏清洁机器人

随着物联网技术的发展，光伏电站的运维也逐渐开始使用物联网技术来代替传统的人力或在线运维。在国内，刘旭提出了基于 LoRa 的低功耗方案来实现了机器人的无线通信，使得工作模式下的系统整体电流降低为 10 毫安^[10]。薛家祥则采用了 Wi-Fi 这种无线通信技术和微逆变器来实现了光伏系统的远程监控，同时提供了 WEB 端和安卓端的应用服务^[11]。另外还有，庄清霖使用 ZigBee 协议实现了分布式光伏电站的监控^[12]。在国外，Jorge Elizalde 等人则使用蓝牙实现了光伏系统的低功耗无线通信^[13]。

1.2.2 光伏板功率预测的研究

功率预测作为整个光伏系统的核心，一直以来都是国内外的热点研究方向。对于光伏系

统的输出预测主要是分为短期输出预测和长期输出预测。本文主要研究光伏的短期输出预测。短期输出预测总体上分为：物理方法和机器学习方法。

物理方法预测光伏输出的原理主要是基于光电转换模型，但由于实际场合光伏所处环境不是理想环境，受到地理位置和气象条件的影响，物理方法的使用场景受限较多^[14]。

目前工业界较为流行使用机器学习来预测光伏系统的输出。常用的算法有 BP 神经网络、支持向量机（Support Vector Machine, SVM）、最小二乘法、混合分类等。杨磊等人通过经典的 BP 神经网络来实现了光伏功率的预测，并通过共轭梯度法等方法来改进了 BP 神经网络的收敛速度^[15]。王倩等人通过粒子群来优化了 RNN 算法，整体能够稳定有效地预测出光伏系统的输出^[16]。Dong, Tian 团队则在对初始训练集上使用 K-Means 算法进行聚类分析，之后在每个类别上使用 LSTM 算法进行了预测。

综上所述，可见国内外在光伏预测领域使用了相当多的方法。但是行业内的预测方法大多在模型训练上存在速度较慢的缺点，并且模型较为复杂，拓展性也不是很好。

1.2.3 光伏板热斑检测的研究

光伏电站的热斑问题本质是光伏板发电的一个可靠性问题，热斑效应是光伏老化的主要原因，热斑效应不仅会损坏光伏板周围的电子元器件^[17]，严重时还会导致太阳能电池发生火灾事故^[18]。如何精准地识别热斑对于光伏系统来说是非常重要的。

光伏热斑地识别一般分为物理法和机器学习法。物理法通过监控相关元器件的电器和电路特性来预测是否有热斑等。机器学习法一般是利用摄像头识别相关图像，并使用边缘检测和图像处理以及其他算来识别图像中的热斑。

在使用传统物理方法检测光伏热斑中，文献^[19]使用了一种基于累计密度函数分析的方法来将热斑分成八个不同类别，使用了功率损耗百分比和退化率两个物理参数进行热斑种类的区分。文献^[20]中则使用了标准热成像图和 Canny 边缘检测算子作为热斑检测的算法，该算法可以有效识别热斑，但对于虚警等干扰光源的识别效果却不太好。文献^[21]中根据热成像图像的纹理和梯度直方图特征进行分类，通过朴素贝叶斯分类器进行热点的相关检测。文献^[22]则是在子图像块温度特征的提取上使用 SVM 来实现热斑检测的，最终该方法的误检率达到了 17.64%，主要为虚警相关的误检。

1.2.4 光伏电站的后端模式研究

由于光伏系统功能庞大复杂，单一服务的部署会导致系统维护困难，可靠性较低。最新的后端技术往往采用微服务架构进行设计，微服务将系统从功能上拆解成多个小型的、分布式的独立服务。微服务模式的亮点在于：（1）根据业务场景编排；（2）智能发布并部署；（3）分散控制；（4）高内聚低耦合。这种风格可以设计出模块化且随着已于维护的系统，各个服务之间耦合性较低从而以极大的敏捷性来设计、开发、测试和发布服务^[23]。微服务中的各个

服务运行在不同的服务器或者进程上，每个服务只负责整个系统的一部分功能。

1.3 光伏电站运维系统的现有不足

根据大量的实际应用和技术调研以及工业界现有现状成果的研究，本文总结出当前光伏电站运维系统的不足有如下几个方面：

(1) 光伏电站在智能清洁机器人方面的功能还不够庞大和健壮，各类系统往往欠缺部分功能。机器人清洁并未结合实际天气信息，导致产生很多无效地清洁，耗费电力和人力。机器人设备程序需要大批进行手动更新，大大提高了运维成本。

(2) 功率预测方面的相关机器学习算法较为复杂，模型训练时间较长，需要通过数据集调节多组参数。由于算法计算耗时过长，无法快速响应后台接口地请求。在此，需要一种轻便、高效、快速并能够保证一定准确性的预测算法。

(3) 光伏热斑检测方面的传统算法会产生误判，主要体现在把光源造成的虚警或其他干扰物体误判成热斑，导致系统的识别率下降。故需要设计一种能有效区分干扰虚警和热斑的检测算法。

(4) 大部分光伏后台系统还停留在传统的单题服务层面，并没有使用高效的微服务架构，也没有将开源的云原生技术应用到后端，无法有效地降低后端的运维和人力成本。

1.4 论文结构安排

本文针对传统光伏电站运维系统的多处不足，使用智能网关、微服务、监控系统、机器学习等技术设计实现了一套功能强大的光伏电站远程运维系统。为光伏电站系统提供了光伏板智能清洁、功率短期预测、热斑检测、智能网关的远程更新等多项功能。系统设计并实现了一种单排机器人的智能清洁和远程运维方案，提出了一种使用功率区间划分的 KNN 光伏输出功率预测方法，并使用基于 XGBoost 优化的 Flood Fill 算法来实现了光伏热斑的智能检测，同时能够有效地区分热斑是否是光源在光伏板上形成的虚警。

文章的内容安排如下：

第 1 章：绪论。本章对整个光伏运维产业的技术背景和国内外最新研究成果进行了介绍，并分析了目前行业上所采取的技术方案，确立了系统的设计方向和任务。

第 2 章：光伏电站的远程运维系统设计。本章首先具体地介绍了整个光伏运维系统的组成和基本架构包括：硬件、嵌入式、网络通信、后台管理、数据存储、后台微服务架构、监控系统、分析预测系统。最后进行实物的展示。

第 3 章：基于区间划分的 KNN 光伏功率预测设计。本章将使用无监督算法快速预测功率输出。系统通过获取环境变量，经过相关性分析后得到预测功率系统的输入量，并设计了一种基于区间划分的算法来将光伏功率的输出映射到各个区间上，最终通过 KNN 算法来预测系统的功率实时输出。

第 4 章：基于 XGBoost 优化的 Flood Fill 热斑检测设计。本章首先介绍了光伏板上热斑形成的原理，然后介绍了 Flood Fill 识别光伏热斑的基本步骤。通过测试说明了虚警造成的干扰，并使用 XGBoost 算法优化 Flood Fill，对热斑形态进行学习，使得系统能够精确识别热斑并排除虚警的干扰。最后和传统的基于 SVM 识别热斑的方法进行了对比。

第 5 章：总结与展望。此部分会总结系统的整体设计和对关键结果进行总结。并对相关的方案进行展望和不足的说明，并进一步的提供研究建议。

第2章 光伏电站的远程运维系统设计

针对目前工业界光伏电站运维操作的繁琐以及智能化分析的需求, 本文设计了一种综合性的光伏电站远程运维系统。系统具备智能清洁光伏板、功率预测、热斑检测、远程运维等功能。系统包括边缘控制端、网络控制端、云平台、微服务后台、监控平台、算法平台以及后台 APP。边缘终端负责对光伏清洁机器人进行驱动以及环境数据的采集。网络控制端负责接受边缘终端的数据, 以及作为网关负责后台和边缘终端的网络通信。微服务后台负责后台业务的实现包括: 数据统计服务、天气预测服务、功率预测服务、热斑检测服务、监控服务等。

2.1 系统总体设计

2.1.1 需求分析

本文设计的光伏电站的远程运维系统包括远程清洁机器人、功率预测、热斑检测等基本功能模块。基本的功能需求包含如下几部分:

(1) 通过控制直流有刷电机来达到控制清洁装置清洁光伏板, 清洁装置可以进行前向和后向的水平运动从而达到清理光伏组件的目的。

(2) 智能网关通过 RS485 通信总线来控制电机模块, 并能够采集环境数据, 智能网关可以将电机相关监控数据以及环境数据(包括: 温湿度、倾斜角、风力、风向、辐照度等)上传至云平台。

(3) 边缘终端使用大疆的热成像无人机获取光伏板的热成像图, 通过内置 APP 将识别图像上传至云平台, 用于热斑的识别。

(4) 后台微服务通过远程过程调用 (Remote Procedure Call, RPC) 进行服务之间的通信, 通过注册中心负责服务的发现, 通过配置中心实现服务器的自动配置, 统一应用程序接口 (Application Programming Interface, API) 网关负责向前端提供超文本传输协议 (Hyper Text Transfer Protocol, HTTP) 接口。主要服务包含: 控制服务、功率预测服务、天气预报服务、热斑检测服务、基础服务等。

(5) 后台控制服务, 通过 HTTP 的方式向智能网关进行命令的下发, 从而达到远程控制光伏机器人的目的。

(6) 功率预测服务可以根据智能网关上报的数据结合历史数据进行当前环境下的光伏系统功率输出预测, 天气预报服务可以结合实时天气以及未来天气进行智能的清洁预测和干预。

(7) 热斑检测服务通过无人机携带的红外摄像头采集红外图像后, 经过相关算法分析得

到单位区域内是否存在热斑，并且能够识别出虚影造成的干扰光斑。

2.1.2 总体架构

整体系统自底向上分别包含：边缘终端、网络层、云平台、后台基础设施、服务层、应用层。系统总体架构图如图 2.1 所示：

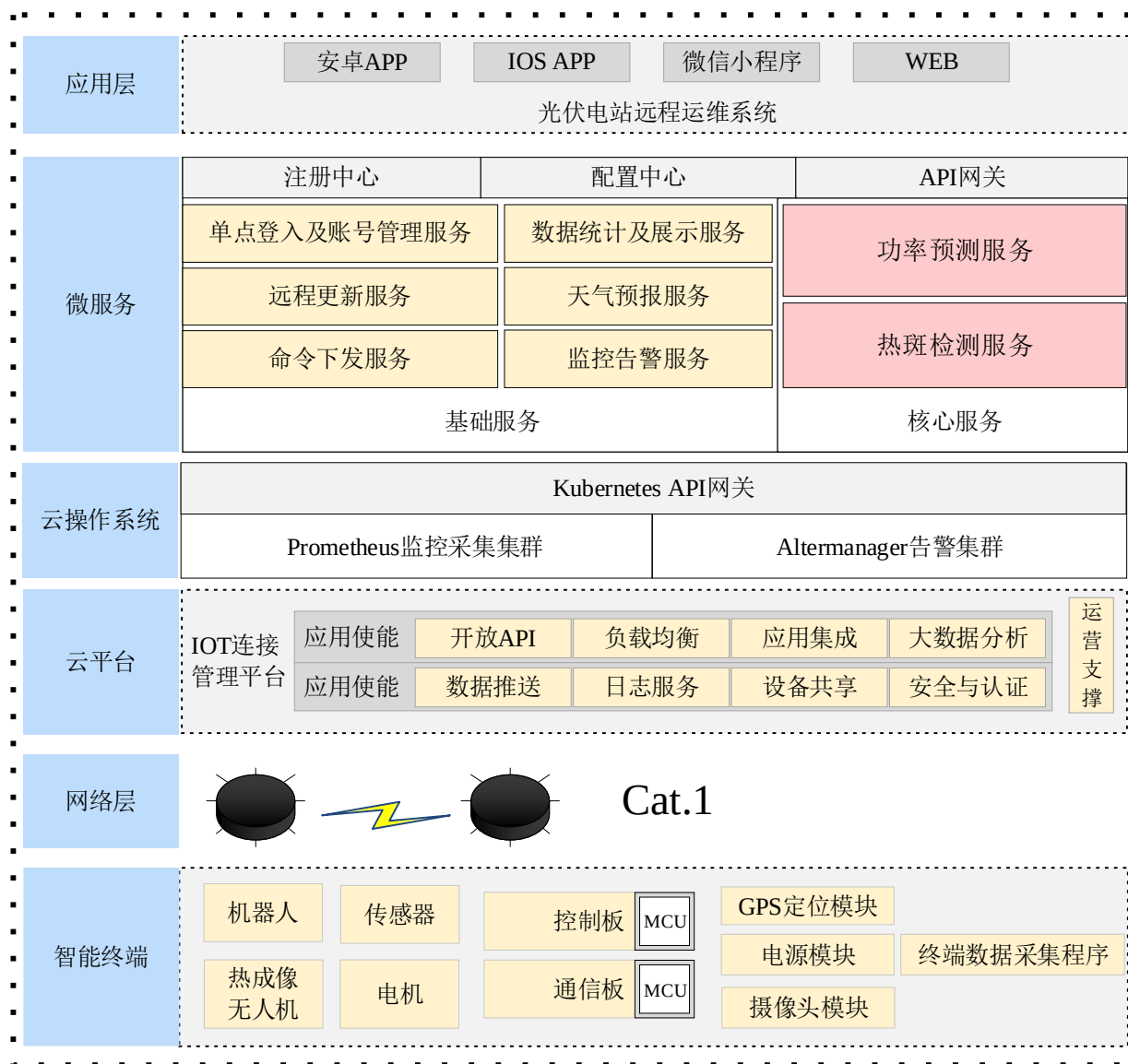


图 2.1 系统总体框架图

系统的硬件部分以边缘终端为主，包含以 ARM Cortex-M3 为内核的控制板和通信板、采集环境数据的传感器组以及携带热成像仪的无人机构成。控制板的 MCU 通过产生 PWM 波来控制光伏清洁机器人的电机，并负责对电机相关数据进行采集。采集的数据通过 RS485 总线传输给边缘终端网关。网关将自身采集到的环境数据和电机相关数据通过 Cat.1 发送给互联网服务提供商（Internet Service Provider, ISP）提供的云平台。携带热成像仪的无人机负责进行红外图像的采集，无人机通过无线方式将图像传送给内置的 APP，APP 通过 HTTP

协议将图像传输给业务服务。

系统的软件部分是一个以微服务、云原生为基础的后台分布式服务。其中的数据统计服务负责将采集到云平台的数据进行展示和持久化。后台命令下发服务完成对清洁机器人的远程控制。功率预测服务和天气预报服务通过机器学习等算法提供了对系统功率输出预测和机器人行为预测的基础服务。热斑检测服务以边缘终端传输的图像数据为基础来识别光伏板上是否存在热斑。微服务系统使用基于云原生的监控系统进行监控。应用层则提供了 APP、微信小程序、WEB 端的一些基础应用。

2.2 边缘终端设计

2.2.1 整体设计

边缘终端需要实现控制、通信、数据及图像采集等功能。在集中式光伏发电场合种，光伏板的一个清洁单位是由若干片小型光伏板组成的最小工作系统，这样一个集合的上下方会固定着清洁机器人行驶的轨道，如图 2.2 所示。光伏清洁机器人的本质就是一个装有清洁装置的长方形物体，该物体通过控制板驱动物体下方的电机来带动上下 4 个轮胎，从而使机器人能在轨道上运动。运动过程中机器人下方的毛刷通过摩擦和旋转来达到清洁光伏板的目的。



图 2.2 光伏板与清洁机器人

本系统中，边缘终端的设计需完成以下两个目标：

- (1) 进行清洁机器人的行走控制。
- (2) 进行网络通信，并将环境数据、图像数据上报云平台。

根据上述两个目标，边缘终端分为控制板、通信板、热斑识别三个模块。控制板通过电机驱动控制光伏导轨上的机器人进行运行。通信板和控制板通过 RS485 总线进行有线。同时，

通信模块使用 Cat.1 和 ISP 的移动基站进行通信，将相关控制数据、环境数据上报给云平台。热斑识别模块负责采集光伏板的红外成像图。边缘终端的整体示意图如图 2.3 所示：

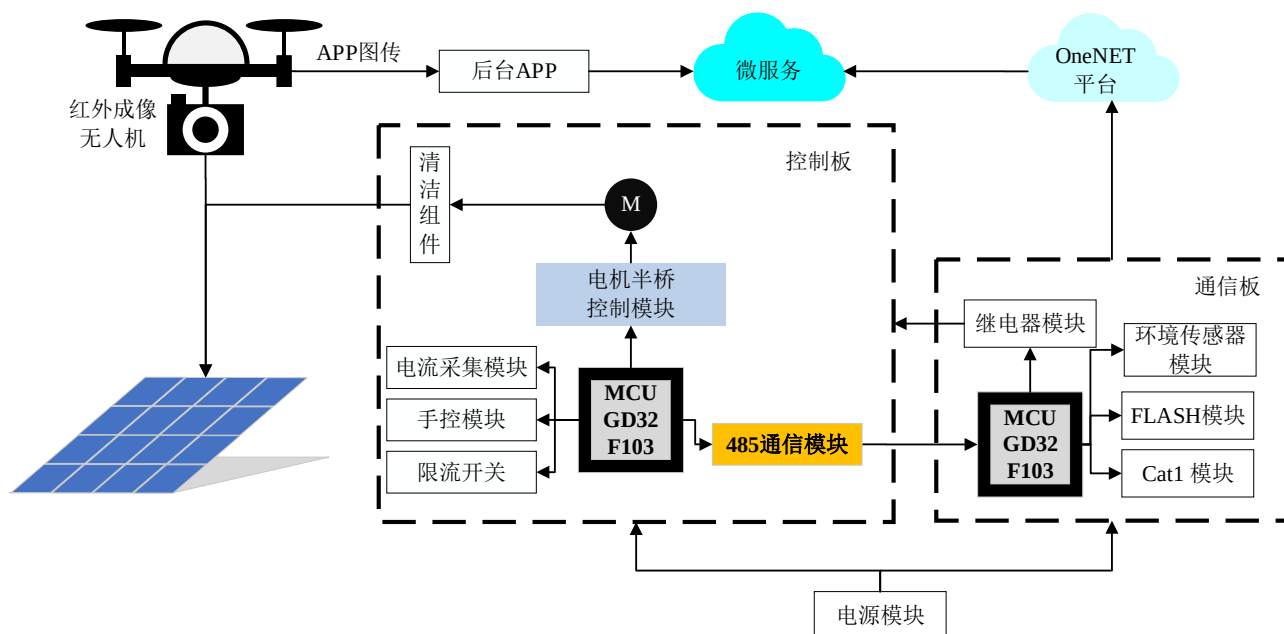


图 2.3 边缘终端整体示意图

2.2.2 控制模块设计

2.2.2.1 电机驱动设计

电机驱动板使用国产芯片 GD32F103 作为主控芯片，该芯片对标的是意法半导体的 STM32F103 系列，但是价格相对来说便宜很多，对于大批量集中式光伏控制系统来说，GD32 在性价比上明显高于 STM32。整个电机驱动系统的设计如图 2.4 所示：

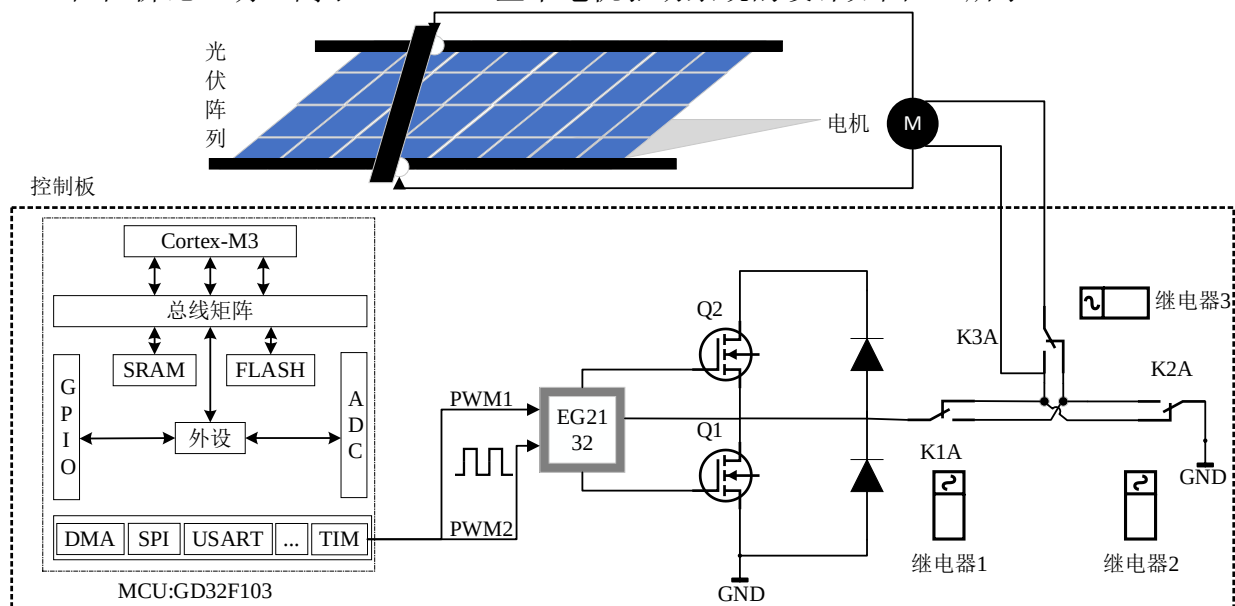
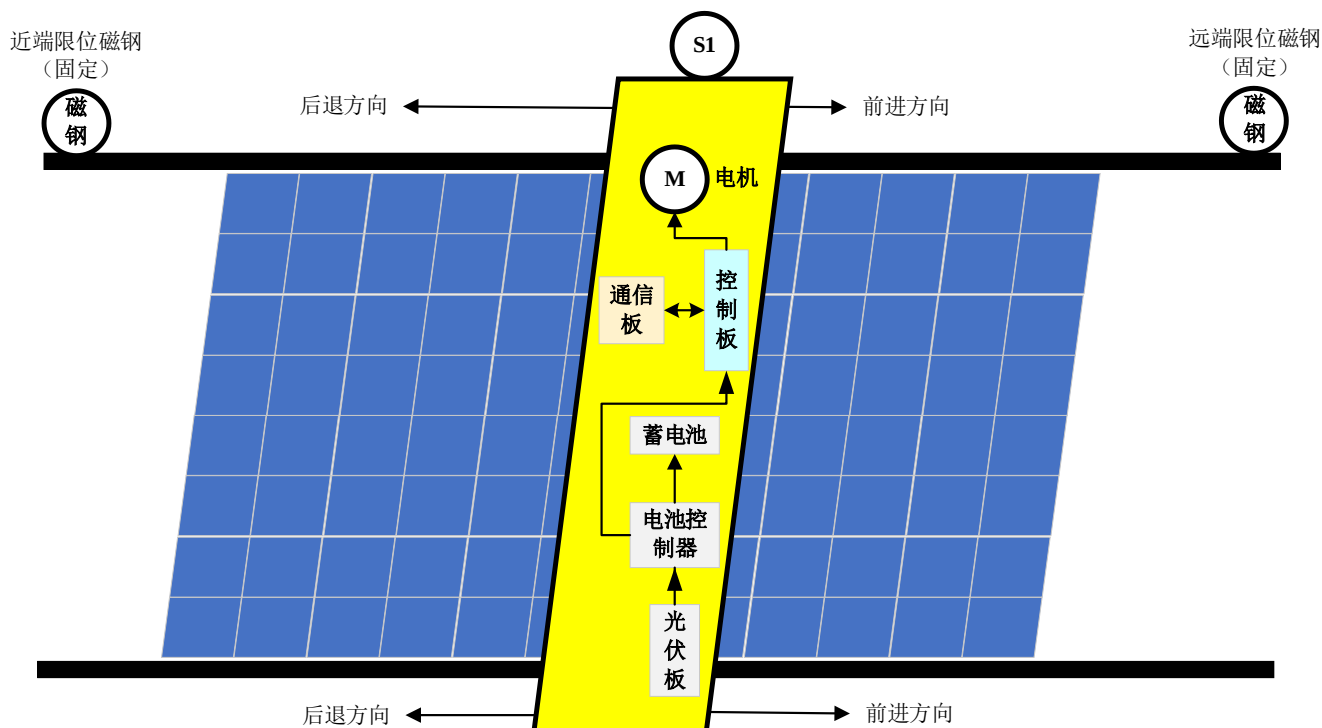


图 2.4 电机控制示意图

带动机器人运行的电机是 12V, 40W 的直流有刷电机, 电机通过 3 个继电器进行控制, 三个继电器分别实现电机的刹车和换相, 继而控制机器人进行刹车、前进和后退。MCU 通过内置的定时器产生了相位相反的两个 PWM 波形, 由于 MOS 管需要 10V 的电压进行驱动, 而 ARM 控制模块则使用 3V 电压进行供电, 此处使用了 EG2132 驱动芯片将 MCU 输出的 PWM 电压信号进行放大, 放大后的电压可以驱动半桥电路。半桥电路有着较高的功率输出, 弥补了驱动芯片功率较低的不足, 同时, 半桥电路可以控制电机的反向和截止。

2.2.2.2 电机控制板工作原理

控制板本质是一个 12V 单电机控制接口电路板, 提供了 6 个接口分别是: 电机接口、电源接口、手动调试接口、触摸屏接口、远近端限位接口、天线接口。集成后的板子仍然具有 2 个 RS485 接口, 1 个可以与调试用的触摸屏进行通讯, 当调试完成即拔掉。另外可预留以备后来第三方软件平台做接口用。另外 1 个可以与太阳能充电控制器进行通讯。具体如图 2.5 所示:



图列:

- | | |
|---|---|
|  磁钢: 用于霍尔传感器的检测 |  S1 限位传感器: 霍尔传感器, 用于检测是否到达边缘 |
|  M 无刷电机: 经过半桥驱动带动整个机器人运动 |  光伏阵列: 清洁机器人清理对象 |
|  导轨: 机器人运行轨道 | |

图 2.5 控制板工作示意图

电机正常运行时，机器人最初处于光伏板的左右两端，左右两端通过磁钢来响应机器人上的限位传感器。机器人检测是否到达光伏左右两端是通过霍尔传感器来实现的。当机器人运行到磁钢旁时，其上的霍尔传感器会检测到磁钢^[24]，同时产生一个低电平使得光耦导通，从而使得另一侧单片机引脚收到检测信号。单片机接受到这个检测信号会进行相关的软刹车、硬刹车等逻辑的处理。

机器人在左右两端开始启动时会延迟若干时间后才会进行霍尔信号的检测，因为此时启动，磁珠就在机器人后方，会对机器人的运动产生干扰。同时当机器人检测到刹车信号时，具体刹车时间和速度的关系可以采用线性刹车或使用归一化等算法得出的函数来进行控制。

控制板通过不同的控制信号来对机器人进行控制，其中有：启动、停止、查询、复位、前进、后退、停止等信号。同时可以通过触摸板和后台来进行电机上下限电流、超时时间、下限停机延时等参数的设置。

2.2.3 热斑识别模块设计

热斑识别模块使用了搭载热成像相机的大疆御 2 系列无人机。通过大疆提供的 OcuSync 2.0 图传方案将实时将热成像无线传输给后台的 APP，APP 软件通过 HTTP 协议将图像实时传输给后台微服务系统中的热斑识别服务进行热斑识别。

2.2.4 网关模块设计

2.2.4.1 网关硬件设计

网关模块的硬件总体设计如图 2.6 所示。网关模块主要包括：主控芯片 GD32F303、外部 Flash、RS485 模块、Cat.1 通信模块、环境数据传感器采集模块。

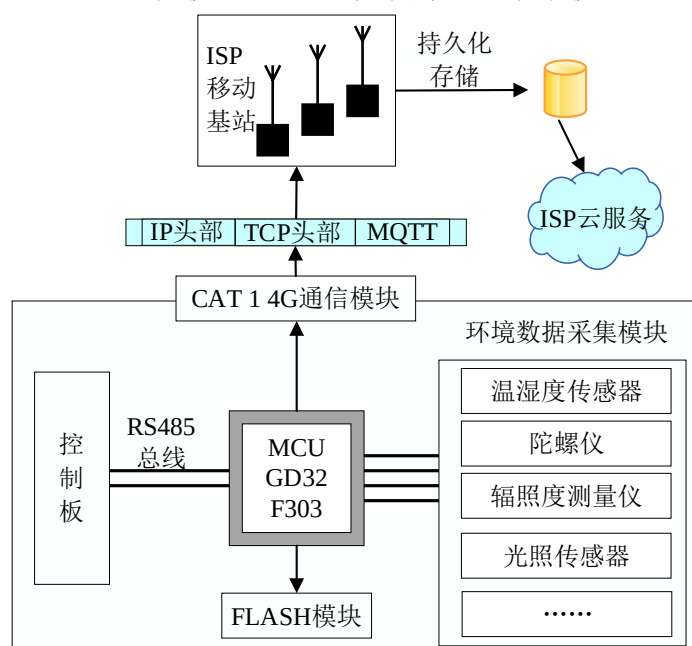


图 2.6 网关与数据采集模块组成示意图

整个通信板部分的供电是由光伏机器人上的电池进行供电的，具体硬件的功能如下：

(1) MCU：和控制板一样，这里采用的是 GD32F303 芯片。具有直接内存访问控制器、TIM、UART、IIC、SPI 等资源。MCU 通过 1 个 UART 和 Cat.1 模块相连，通过 1 个 UART 使用 485 总线和控制板相互通信，通过 SPI 方式和 FLASH 模块通信。

(2) FLASH 模块：由于系统软件程序量较大，功能复杂，代码量早以及超过 MCU 自带的 ROM 容量。所以采用外界 FLAHS 的方式加大程序的存储。

(3) RS485 接口：该接口直接对接工业设备，为了避免共模干扰，常常在对应电路上防止对应的光耦隔离单元。

(4) Cat.1 通信模块：Cat.1 也叫做 eMTC，其中的 Cat 为 Category 的缩写，全称是 LTE UE-Category 1，在工业场景的网络通信设计上，Cat.1 可以无缝接入现有 LTE 网络中，无需针对基站进行软硬件升级，网络覆盖成本低^[25]。选择通信方式时，本系统总结对比了如表 2.1 所示的通信方式：

表 2.1 当前常用无线通信技术

	Zigbee	Cat.1	NB-IoT
组网方式	基于 Zigbee 路由器	蜂窝数据	基于移动网络
部署方式	网关+节点	节点	节点
网络延时	不到 1S	不到 100ms	6S-10S
网络节点容量	200-500 个	10 万个	20 万个
传输距离	<10m	>5Km	>10Km
频段	未认证 2.4G 频段	2.4G 运营商频段	认证频段 运营商频段
传输速度	理论是 100kbps	理论 5.2Mbps-10.3Mbps	理论低于 100kbps，受限于串口速率
适用场景	室内	室外，对速率有一定要求	室外，大规模传感器网络

由于集中式光伏电站多数处于戈壁这种环境，戈壁滩上基于 NB-IoT^[26]的基站的建设没有那么完善，Cat.1 和 Cat.4 都是 LTE 制式的只不过等级不同，而戈壁上 LTE 制式的基站比较完善，并且 Cat.1 的速度远远快于 NB-IoT，所以相比 NB-IoT 来说，Cat.1 更加合适。

(5) 环境传感器：环境传感器的设立是为了采集光伏工作环境下的基本物理量，这些物理量都会直接或间接的影响到光伏系统的功率输出，环境传感器根据传感器与 MCU 通信的不同一般会使用 IIC、SPI、串口等通信方式。

2.2.4.2 网关软件设计

网关软件部分的设计主要涉及到命令下发、数据上传、远程更新这三大模块。网关系统软件的整体结构如图 2.7 所示。

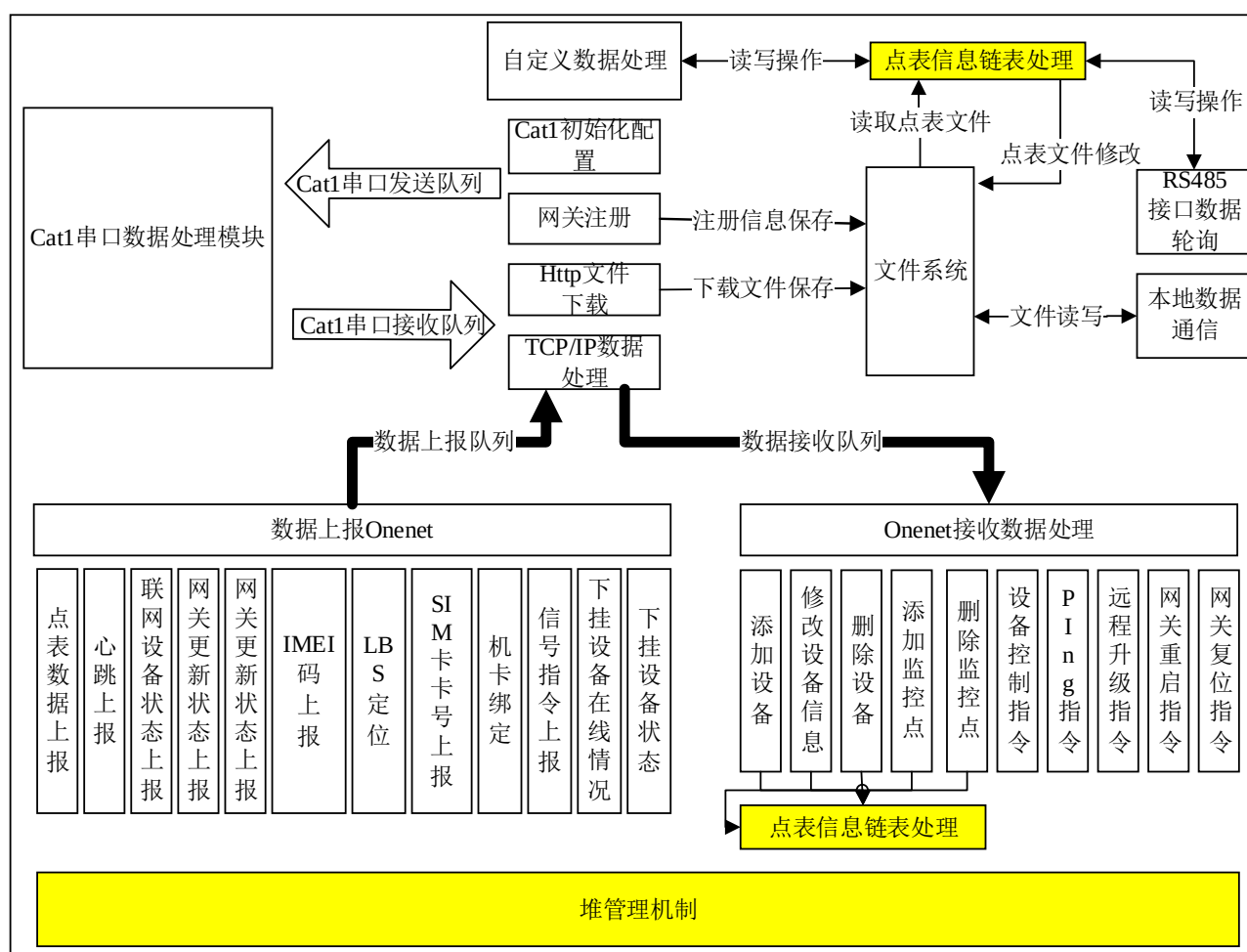


图 2.7 网关软件结构图

帧类型：网关需要对命令、告警、和数据分别组帧进行与后端平台的交互。根据帧类型分别分为：数据帧、命令帧和告警帧，如图 2.8 所示。

数据帧

帧头				数据域	帧尾	
帧起始符	控制码	帧计数器	数据域长度	数据	校验码	结束符

命令帧

帧头				数据域		帧尾	
帧起始符	控制码	帧计数器	数据域长度	命令标识	参数	校验码	结束符

告警帧

帧头				数据域		帧尾	
帧起始符	控制码	帧计数器	数据域长度	告警来源	告警内容	校验码	结束符

图 2.8 帧类型示意图

命令下发：命令下发是后台人员远程命令下发给云平台，云平台通过消息队列遥测传输（Message Queuing Telemetry Transport, MQTT）^[27]协议将相关指令序列化，通过移动蜂窝

数据传送给网关设备，网关设备反序列化出 MQTT 协议中的相关命令，并对控制板进行相关指令操作。云平台将数据帧组以后，通过定义的 JSON 格式数据发送给网关，再由网关进行透传操作，透传完成以后，如果云平台有回复，则将回复的数据内容，使用 16 进制字符串的形式，上报到指定的监控点。

数据上传：控制板通过 RS485 总线将电机相关数据和监控点数据上传给网关设备，网关将相应数据存储在点表信息链表当中，如图 2.9 所示。通信板 MCU 通过全局事件轮询的方式在相应时刻执行链表中不同设备和数据监控点的轮询，将相关数据和异常数据通过 Cat.1 模块上报给云平台。

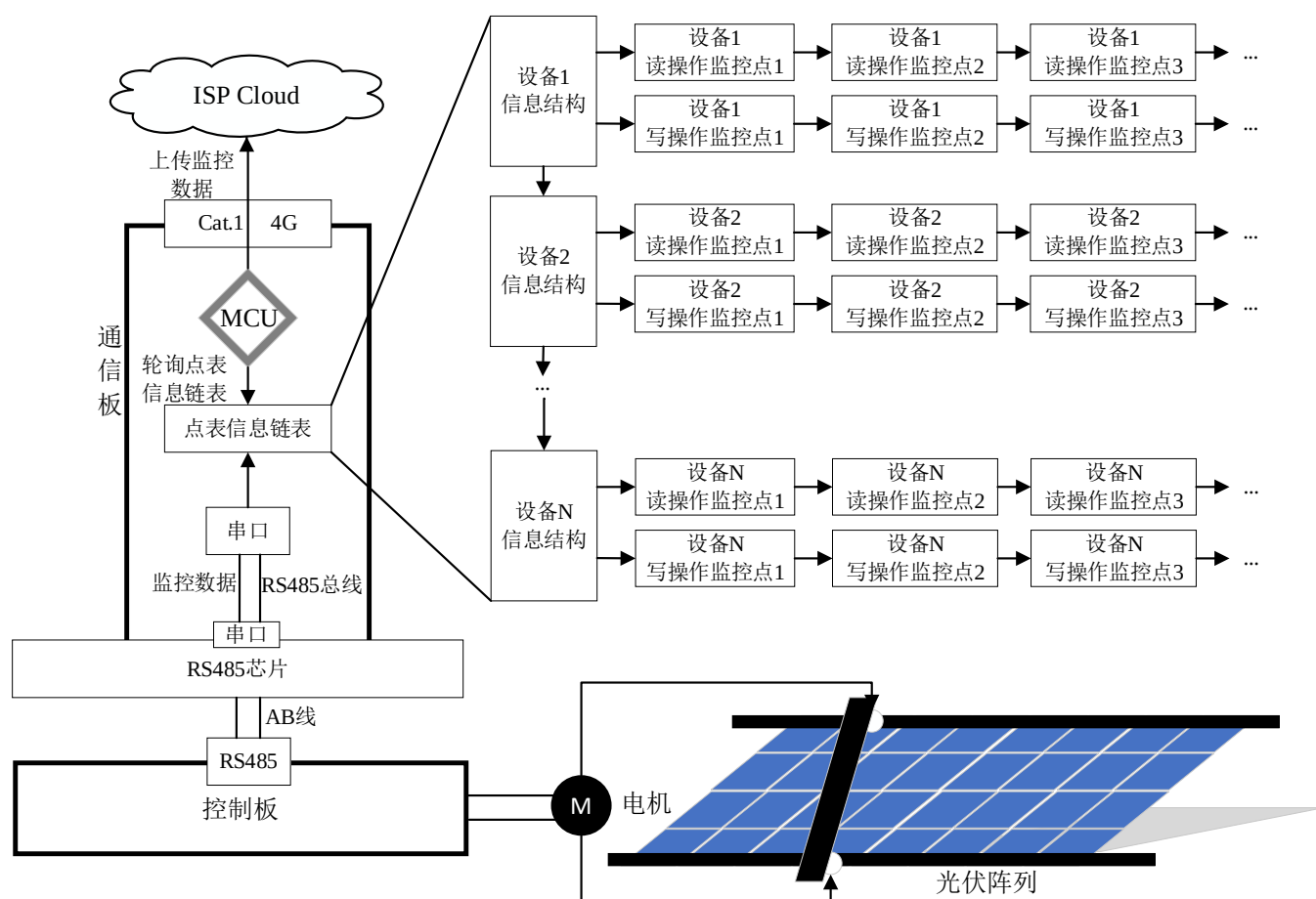


图 2.9 数据上报示意图

远程更新：分为后端部分和嵌入式部分，网关模块通过一个开源的轻量级 TCP/IP 库 LWIP 实现了 HTTP 服务。网关通过后端提供的文件下载接口，通过 HTTP 的方式实现文件的远程下载，将更新的二进制文件存放于 FLASH 中。网关远程更新的状态转移图如图 2.10 所示。

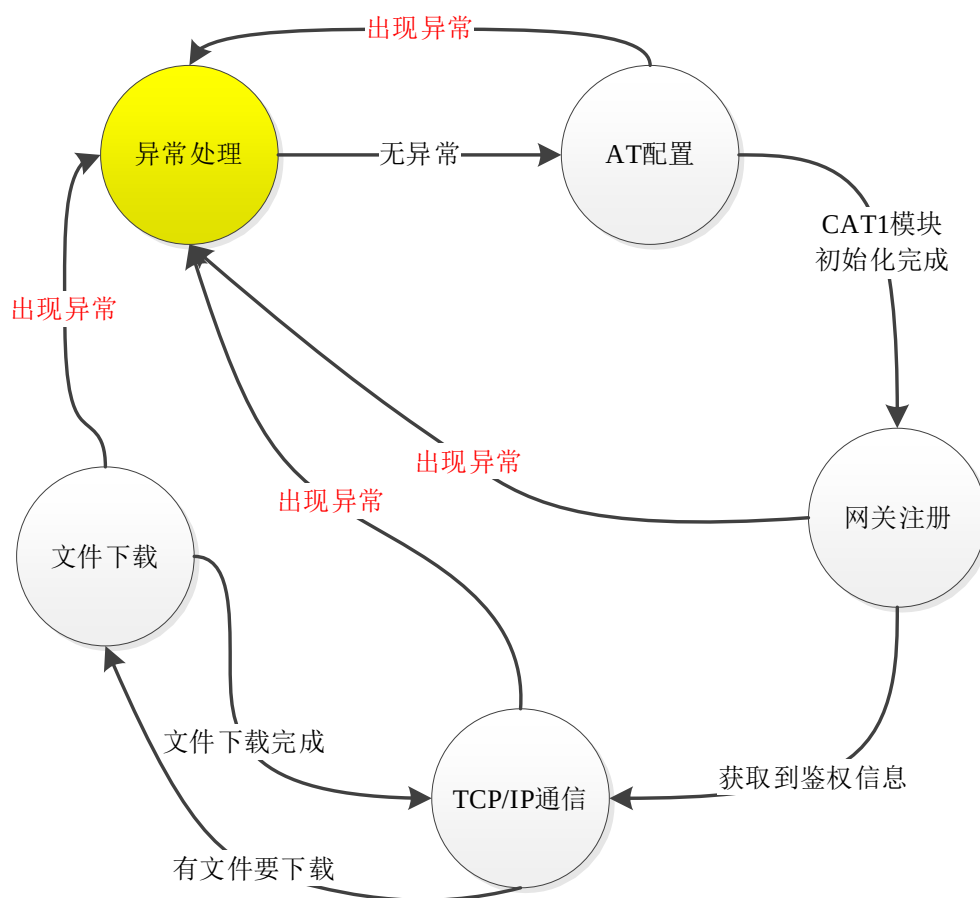


图 2.10 远程更新状态转移图

系统重启时通过判断 FLASH 之中的一个标记位是否改变从而决定是否将 FLASH 中的程序覆盖掉当前 MCU 存放于 ROM 中的老程序。详细的更新操作示意图如图 2.11 所示。用户启动远程更新任务后，后台将相关更新指令进行 JSON 序列化，通过 ISP 的云服务告知通信板，通信板获得信号后会访问后台的文件服务，使用 HTTP 下载文件，并通过 DMA 的方式将二进制文件存放于 FLASH 中。

通信板和网关板相关的更新逻辑是不同的。通信板在进行更新时是通过 BootLoader 将 FLASH 中的新的程序覆盖掉本地 ROM 中的老程序然后进行重启，以实现远程更新。控制板的更新需要借助通信板，通信板通过 RS485 总线将控制板的程序下载到控制板的 ROM 中，此时串口会改变波特。控制板重启后，BootLoader 会检测是否要进行新版本的更新，如果检测到需要更新，则会将待更新程序覆盖掉老程序，并进行新程序的启动。

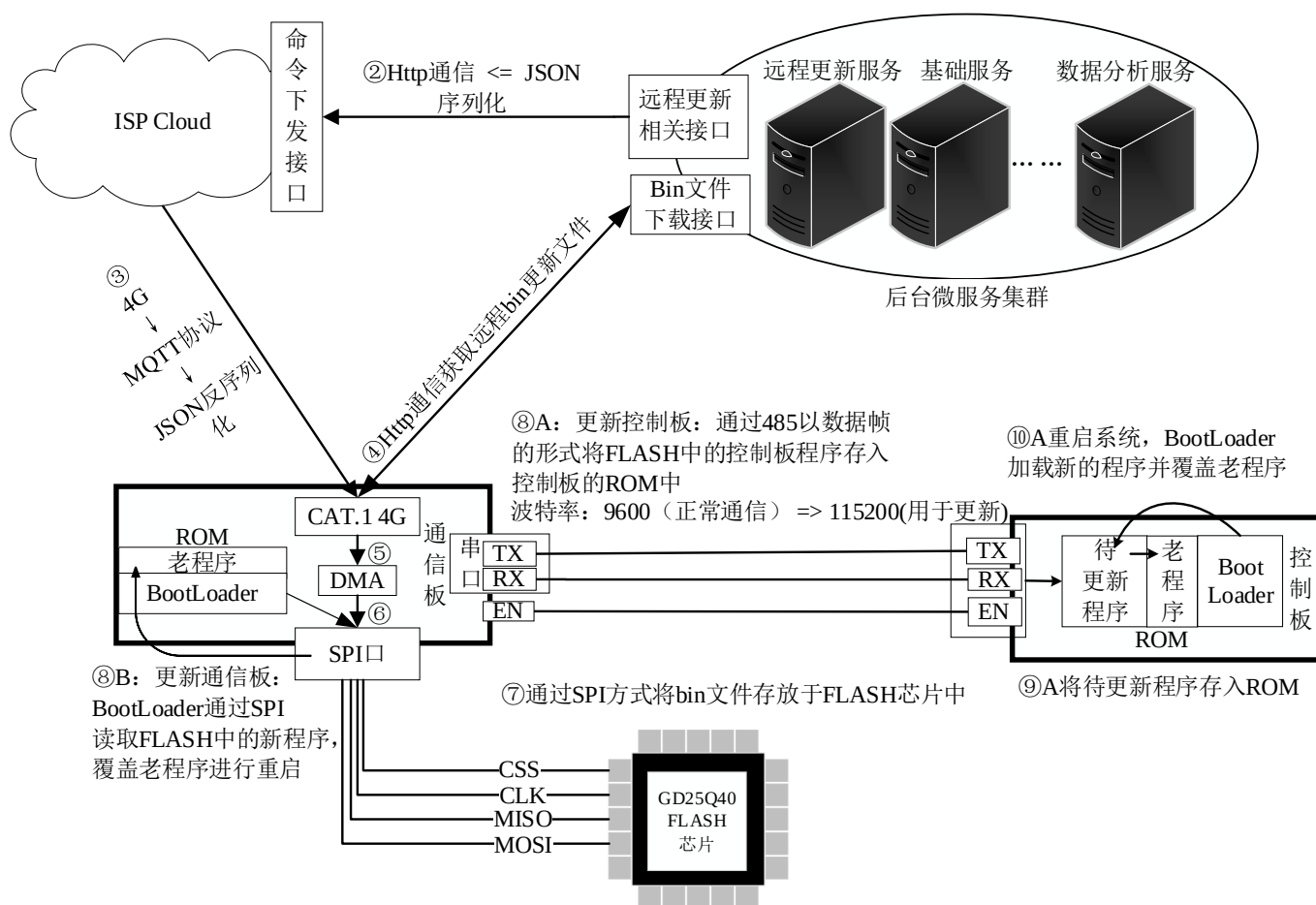


图 2.11 远程更新原理

2.3 云平台

云平台在系统中起到沟通边缘终端并要负责分析和展示边缘终端的数据，也要对边缘终端进行相应的远程控制。云平台总体分为：设备云、业务云。其中业务云主要是一整套基于云原生和微服务的分布式系统，而设备云主要是对接边缘终端的，是 ISP 提供的数据库操作云平台。

本系统由于在网络层采用的是 Cat.1 通信，使用的载体为移动的物联网卡，所以相关的数据也经过基站持久化到了移动运营商的数据库中，采用的设备云是中国移动开发的物联网开放平台：OneNET 平台。

OneNET 平台通过 MQTT 应用层协议，将边缘终端 JSON 序列化后的数据上报给云平台，并提供了丰富的 API 接口，比如：设备监控点批量查询，设备命令下发等。平台的操作页面如图 2.12、2.13 所示。开发者可以根据业务的需要进行产品管理、运维监控、设备管理等操作。

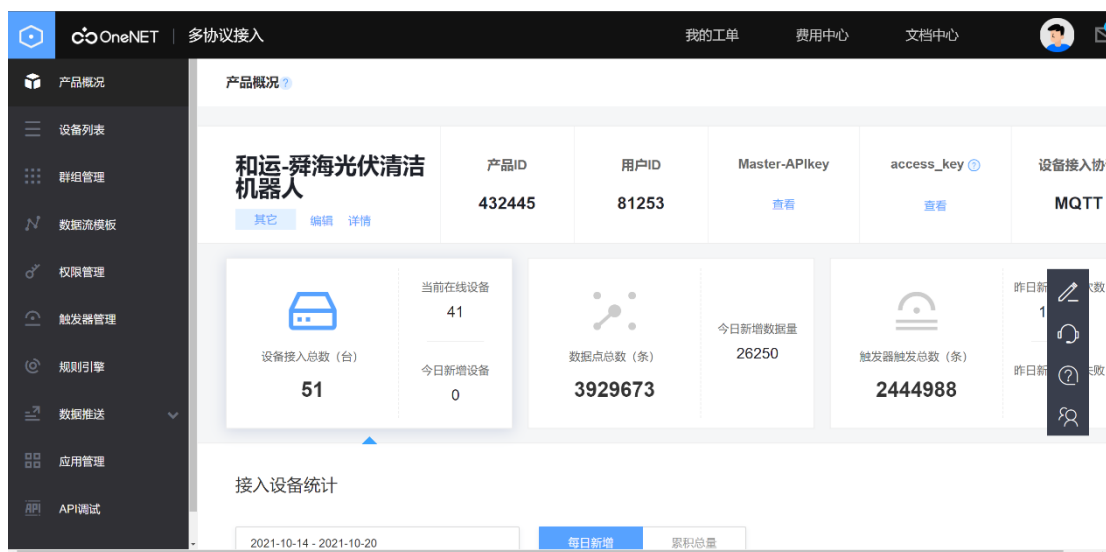


图 2.12 设备云平台界面

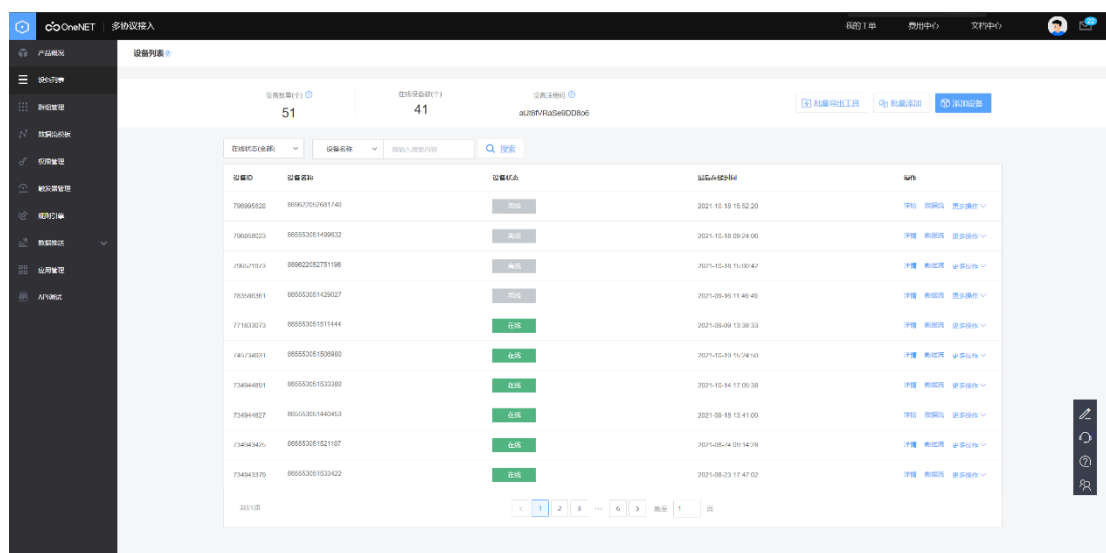


图 2.13 设备运设备在线展示

2.4 后端架构

2.4.1 后端架构选型

后端是整个系统的核心，它负责对接设备云，并且提供数据的持久化操作、数据的统计分析、系统监报告警、以及对具体业务提供基础设施的支持。

整个系统通过 SpringBoot 和 SpringCloud 框架来实现微服务^[28]。Spring 是为了简化 java 企业级项目设计而研发的一个框架。随着微服务以及云原生技术的发展，Spring 系列推出了 SpringBoot 和 SpringCloud 来支持了微服务架构下的项目开发。SpringBoot 在 Spring 的基础上省去了大量的配置代码，通过服务提供接口（Server Provider Interface，SPI）技术实现了基于自动配置类的自动装技术在技术，大大的降低了开发的复杂度，使得用户可以专注于具体的业

务实现。而 SpringCloud 是专门用于微服务而设计的，它提供了微服务中最为核心的技术如：服务发现、服务注册、熔断限流、负载均衡、动态配置、弹性伸缩等。

在持久化层，系统选择了业内最为流行的 MySQL 作为数据库。同时软件层面的对象关系映射（Object Relational Mapping，ORM）层使用了 MyBatis 来进行 MySQL 的操作。SpringBoot 通过整合 MyBatis 的 Starters 来实现自动配置，同时通过 XML 文件进行 SQL 的操作映射。

整个系统在业务层还需要使用分布式缓存来进行一些其他基础服务的支撑如短链接服务、缓存服务等。为此系统使用了 Redis 作为分布式缓存。Redis 使用内存作为数据的存储媒介，内存的速度是磁盘速度的 10 倍，所以 Redis 的速度比传统的基于硬盘的数据库会快很多。后端采用 Redis 缓存作为接口的多级缓冲层，降低了大批量请求下数据库的压力，同时提高了接口的访问速度。

系统采用 Nginx 作为后端的网关，负责将原始的 HTTP 请求负载均衡到不同的微服务 API 网关上，达到请求分流的作用。

最终整个系统的设计架构图如图 2.14 所示。

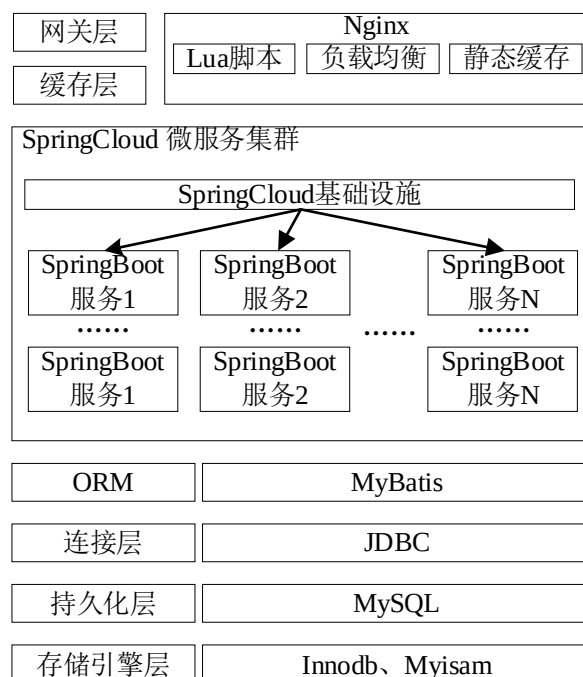


图 2.14 系统设计架构图

具体的微服务设计会在后端系统架构和业务架构设计中展开讨论。

2.4.2 后端业务架构

后端业务系统经过功能化拆分后一共拆分出来了三类服务：业务服务、分析预测服务、基础设施服务。业务服务对用户提供光伏电站业务相关的服务。分析预测服务包括短期功率预测模块和热斑识别模块。基础设施服务则是为上述两个服务提供基础功能的，一般是各个

业务的一些通用的服务。下面对三类服务展开介绍。

业务服务：该服务是核心业务的实现，主要为光伏管理者提供数据监控、显示、管理等功能。包含系统概况、电站管理、设备管理、综合管理等基础模块，可以实时展现集中式光伏集群的运行情况和相关信息。

分析预测服务：功率预测部分会根据边缘终端上传的环境数据作为数据输入，预测光伏电站的实时输出预测功率。同时该服务会根据天气实时信息和天气预报进行清洁机器人相关行为的干预。热斑检测部分通过边缘终端的摄像头识别的图像作为输入，通过机器学习等算法对热斑形态进行学习和识别。

基础设施服务：为核心业务提供基本接口，如文件上传下载、远程更新、设备云的沟通、单点登录等。

整个后端从系统业务功能的角度来看的逻辑架构图如图 2.15 所示。

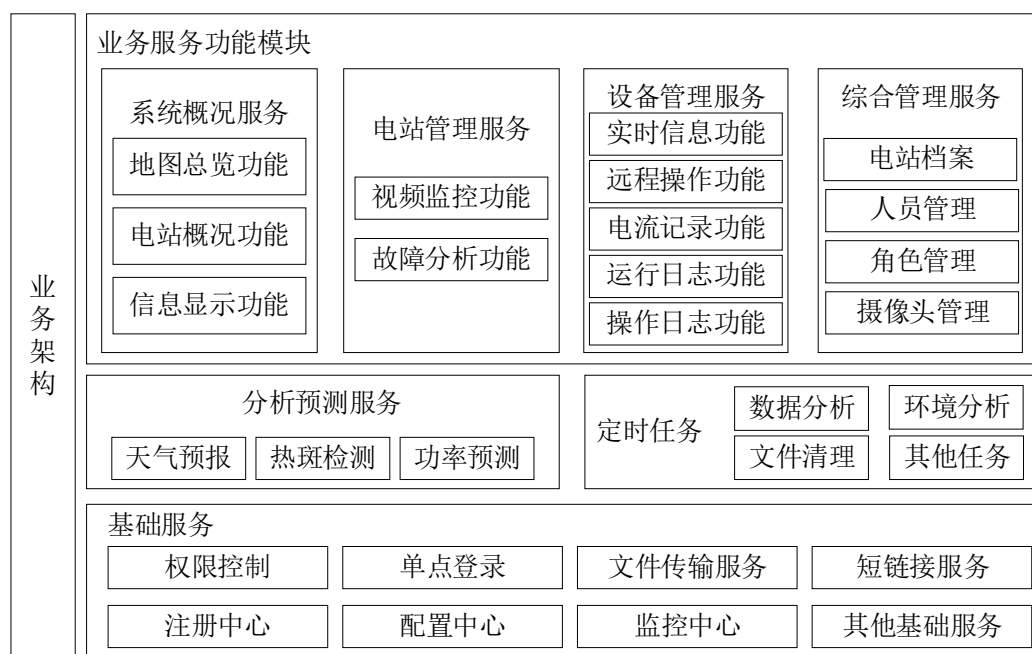


图 2.15 系统业务架构图

2.4.3 后端系统架构

后端技术组件完成选型后，现对整体业务的实现进行具体的实现。整个系统以基础微服务实现的业务架构和以 Redis 为核心的分布式多级缓存为中间处理层。API 网关和 Nginx 为访问层，MySQL 集群为持久化层，以 Kubernetes 为云操作系统的监控层。具体的设计图如图 2.16 所示。

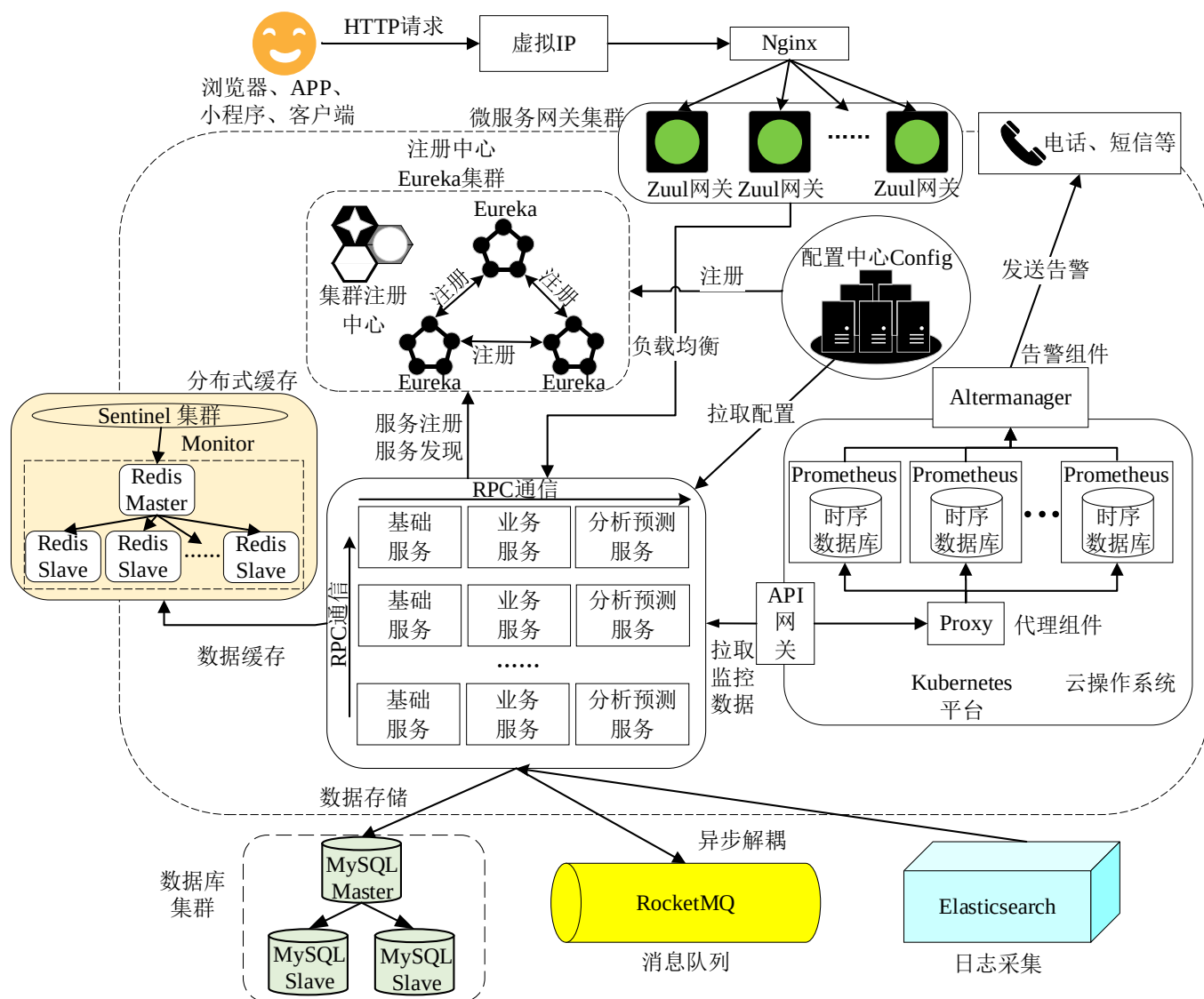


图 2.16 系统架构图

用户通过 WEB 端或者客户端发起相关 HTTP 请求，首先会通过系统绑定的虚拟 IP 映射到 Nginx 网关，Nginx 网关首先判断请求的响应是否可以通过静态缓存或者 Lua 脚本进行前置处理，如果可以那么请求将在 Nginx 端解决，这样做得目的是加快请求响应，降低服务端压力。当 Nginx 端无法解决时，它会将请求通过轮循的方式负载均衡到 SpringCloud 层的 Zuul 网关，Zuul 网关^[29]根据请求类型路由到正确的业务服务进行处理。业务服务首先查询分布式缓存中是否有缓存数据如果有则直接返回，如果没有则进行数据库查询，将所得数据先缓存于 Redis 中然后在返回。系统中的各个服务通过注册中心 Eureka 进行服务的注册和其他服务的发现。各个服务通过 HTTP2.0 协议进行通信。配置中心负责存储系统的配置，服务可以拉取配置中心的配置达到动态配置的效果。基于 Kubernetes 的容器集群中运行着多个 Prometheus 实例，Prometheus 实例每隔一定时间会将各个组件的监控数据拉取到本地时序数据库（time series database, TSDB），并分析是否需要告警，如果需要则调用 Altermanager 这

个告警组件发送告警。

2.5 服务设计

本小节将针对系统中各类服务进行具体设计的说明。

2.5.1 基础服务设计

2.5.1.1 缓存设计

缓存服务作为请求获取数据的第一道关卡，起到加快系统访问速度、防止批量请求下对后台数据库造成压力的作用。所以缓存服务在设计的时候必须要考虑到几个点：高可用、高效能、低耦合。

本系统通过使用多级缓存来降低了系统的耦合度，通过使用内存作为缓存达到了高性能的目的，对 Redis 缓存采用哨兵模式，实现了系统的高可用^[30]。

系统的缓存设计如图 2.17 所示。

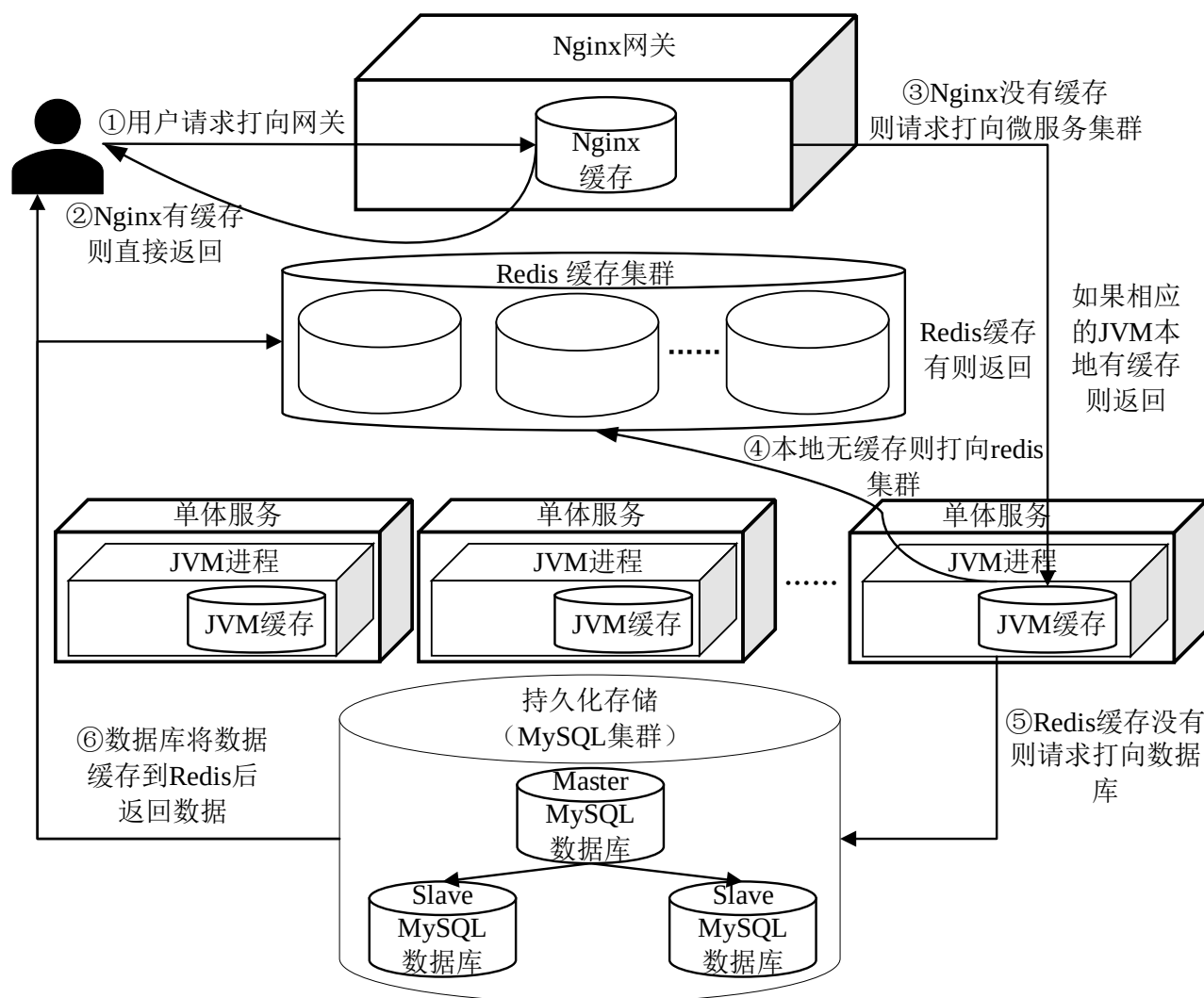


图 2.17 系统缓存架构图

缓存服务的一级缓存位于 Nginx，本质是 Linux 系统的一块内存区域，当请求打向系统是，Nginx 会根据配置的映射规则判断是否使用 Nginx 缓存如果匹配成功，则直接返回请求数据。未匹配成功则请求会放行到本地二级缓存，本地二级缓存本质是虚拟机（Java Virtual Machine, JVM）本地堆缓存，它主要缓存一些各个进程中改变频率不大但经常被读取的数据或常量。如果相关业务接口判定可以使用本地缓存，则请求直接返回，否则 JVM 进程会向远端分布式缓存获取相应缓存。

Redis 分布式缓存容量设置未 16GB，存储了大量的数据缓存，是整个缓存服务的核心，如果 Redis 中没有缓存数据，意味着系统发生了缓存击穿，请求会打向数据库，数据库进行 IO 操作后将数据返回给 JVM 进程，业务逻辑会将数据缓存一份至 Redis 中然后再将数据返回。表 2.2 总结了不同层级的缓存的不同特点。

表 2.2 不同缓存相关特点

	Nginx 缓存	JVM 进程缓存	Redis 分布式缓存	MySQL
缓存层级	第一层	第二层	第三层	最后一层
缓存容量	1GB	单进程 500MB	16GB	100GB
请求返回速度	非常快	非常快	快	慢
数据变更频率	不变	几乎不变	变更不频繁	存储所有数据
数据类型	系统相关不变的数 据或文件	进程相关不变的 数据	经常查询的数据	存储所有数据

Redis 集群作为缓存的最后一道门槛，在设计的时候必须要保障它的高可用。Redis 支持分片集群和哨兵集群两种高可用方案。本系统采用了哨兵模式，如图 2.18 所示。哨兵集群有多个哨兵节点，每个哨兵节点都会定时的向其他哨兵以及 Redis 节点发送心跳包和命令。当哨兵发现主机节点长时间未进行心跳回复时，这个哨兵会主观的认为该 Master 宕机了，此时哨兵系统会发起投票，如果哨兵集群中有一半认为目标机器宕机了，哨兵集群就会从 Slave 中挑选一个与当前主机的 Offset 最接近的节点作为主机，并告知其他节点新的 Master 节点信息。同时哨兵集群会监控宕机的老 Master 节点，当宕机节点恢复后，哨兵集群会让它成为新的 Slave 并连接新的 Master 节点。哨兵集群间则通过 Raft 一致性算法保证节点之间的通信正常和一致性判决。上述过程完成了缓存服务的故障切换，从而达到高可用的目的^[31]。

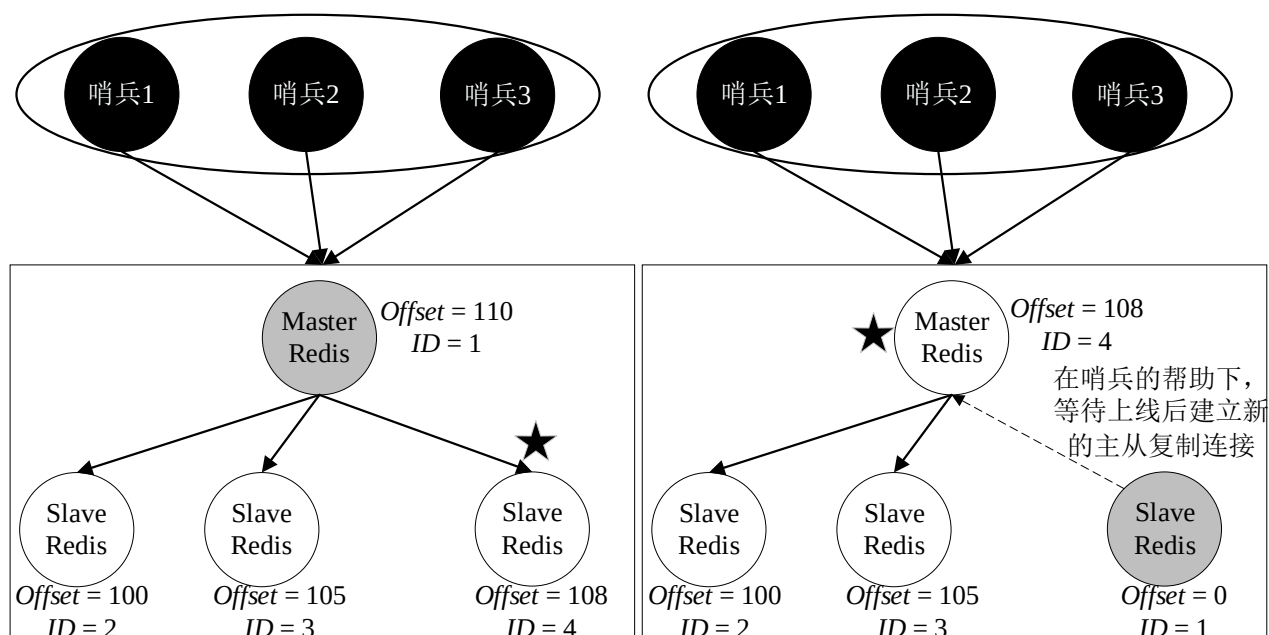


图 2.18 故障恢复示意图

2.5.1.2 远程更新设计

远程更新服务主要服务于边缘终端网关设备和控制机器人的控制板的程序更新。该服务的实现以操作设备云为基础。

在数据库层面，远程更新核心的表结构设计有 6 张表分别为：用户权限表、本地产品表、主任务表、更新设备表、软件更像包表、用户表。核心接口 API 如表 2.3 所示。

表 2.3 远程更新核心接口

接口名称	请求 URL	请求方式
创建远程更新主任务	/api/onenet/new	POST
开始主任务	/api/onenet/update/{id}	GET
获取在线绑定设备信息	/api/onenet/device/bind	GET
绑定主任务更新的设备	/api/onenet/device/bind	PUT
查看升级详情	/api/onenet/device/{taskid}	GET
查询任务详情基本信息	/api/onenet/device/base	GET
创建远程更新产品	/api/local/product	POST
添加升级包	/api/package/create	POST
查看某产品下的升级包	/api/package/product/info	GET

远程更新的基本操作流程如图 2.19 所示，流程图的每一个步骤都由远程更新服务的相关接口进行支持。远程更新下，不同产品有不同产品的更新包，系统通过定时任务，定时的拉取 OneNET 上的产品信息到数据库中。用户通过上传最新版本的二进制文件，来实现不同包

的存储。发起更新任务时，通过选取已知产品下的其中一个升级包来确定升级任务的信息。通过查询设备接口，可以将任务下关联的设备展现出来，可以选取要更新的设备来进行批量更新。一旦任务确定，用户调用开始接口，远程更新任务将在后台开始进行。

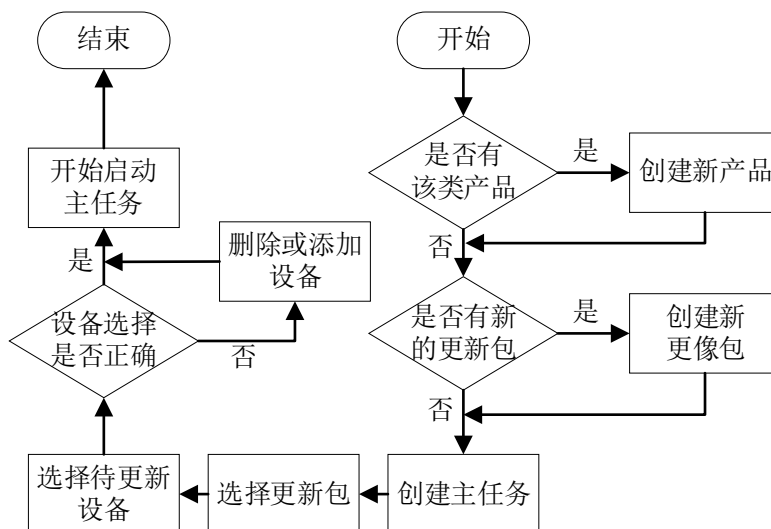


图 2.19 远程更新基本流程图

用户的正常操作流程会在启动主任务之后结束，然而主任务是一个非常复杂的后台接口。主任务的执行过程如图 2.20。

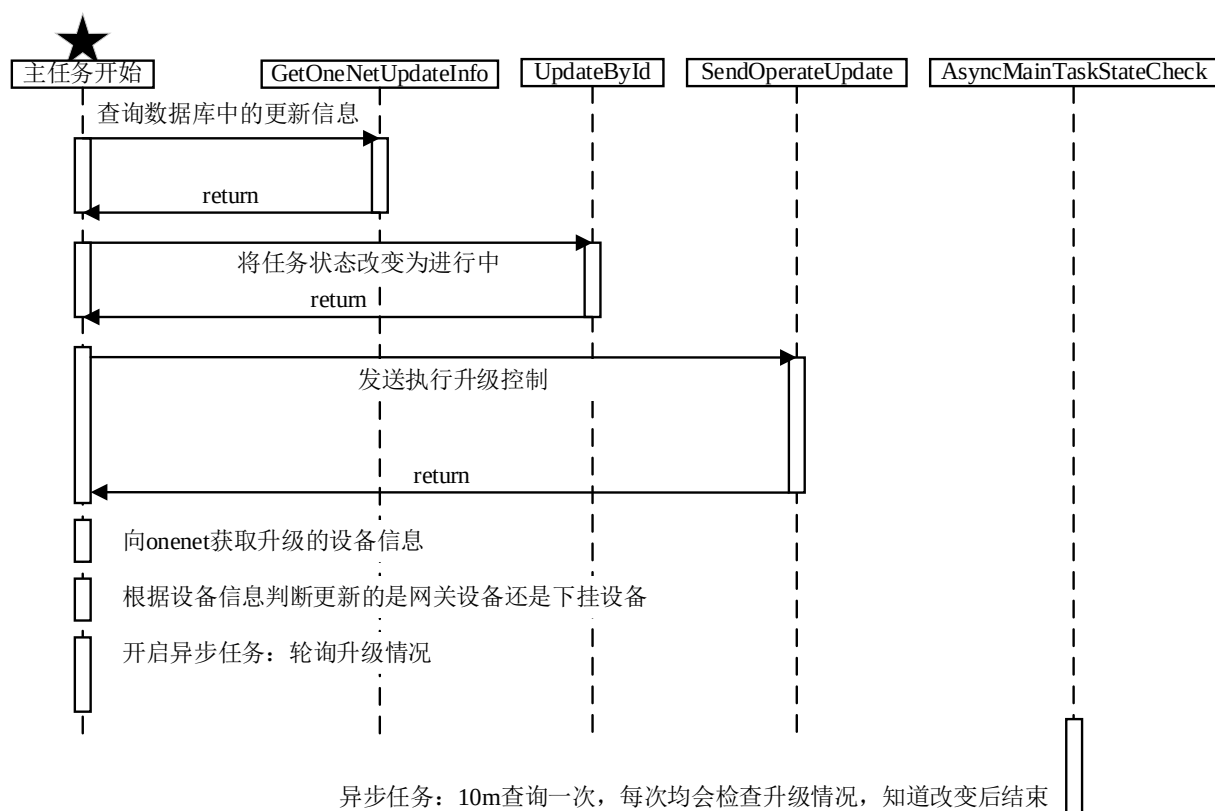


图 2.20 远程更新主任务时序图

主任务一开始会从数据库中获取相关的更新信息，并将数据库中的任务状态设置为进行中，设置启动时间、任务 id 等相关字段。然后主任务开始升级控制，升级控制主要保证主任务所关联的设备开始执行远程更新。主任务会判断是进行网关更新还是控制板更新，并启动后台线程进行更新任务的轮循，直到后台线程发现任务状态改变后，它会将新的状态写入数据库，并结束自己的生命周期。

2.5.1.3 远程更新多线程优化

上文讲述了远程更新如何进行主任务的启动和主任务的基本流程，然而在批量更新远程设备时，由于移动 OneNET 平台提高的命令下发接口最多只能允许对下挂的 100 台设备进行更新，当机器人设备数量达到千级别的时候，对千级别的设备进行更新，单个串行处理接口需要调用几十次的 HTTP 请求，所以为了提高效率系统需要对程序进行并发编程^[32]。

如图 2.21 (a) 所示，传统编程在处理大批量设备时，会通过单一线程发起 HTTP 请求，请求后的数据存储在服务器的容器中，然后再继续发起请求，这样高性能服务器上的其他 CPU 就得不到充分的利用，并且在数据量大时，单个线程处理大批量请求会造成接口处理效率较低，影响用户体验。如图 2.21 (b) 所示的方法充分利用了 CPU 资源，但是注意并行编程的一大特性是需要满足无状态性，由于远程更新需求在批量更新时，无论是先更新拿几台设备，更新的结果都是一致的，所以说系统满足无状态性，并行请求结束后，只需合并响应结果即可。

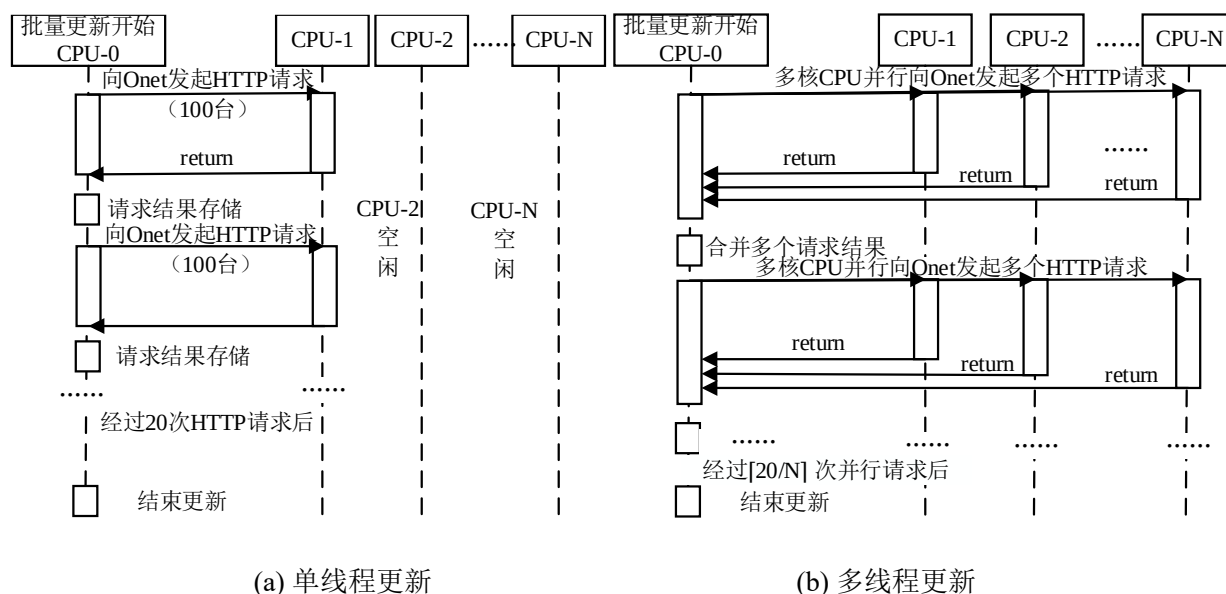


图 2.21 并行请求优化时序图

对于 K 台远程运维微服务器，每台服务器的核心为 C 的系统来说，在一次主任务为 N 台更新设备的情况下，假设 OneNET 的一次更新请求的响应时间为 T ，则使用串行编程的更新响应时间为： $NT/(100K)$ 。而使用并行编程的更新响应为： $NT/(100KC)$ 。然而多线程下的线

程之间的切换是会造成性能的影响，也就是说使用多线程不一定比使用单线程的效果更好，特别是在设备数量 N 比较小的情况下，单线程的效率甚至远远高于并行请求的效果。

2.5.2 业务服务设计

本小节，将介绍光伏管理系统的后端业务设计，将从接口需求设计、数据库设计、逻辑设计几个角度来进行说明。

2.5.2.1 接口需求设计

业务端逻辑需求进行电机、用户、电站设备等相关的管理和控制。按照模块划分的功能如表 2.4 所示。

表 2.4 光伏管理系统核心模块及其功能

模块名称	模块名称	模块 URL
人员管理模块	操作人员的权限管理	/manage
电站模块	电站相关信息和操作	/station
设备模块	网关设备和控制设备相关操作	/device
数据模块	电机、光伏相关数据展示	/data
控制模块	对光伏机器人进行远程控制	/control
数据统计模块	对相关数据进行数据统计和分析	/statis
摄像机模块	提供直播、实时视频监控、红外成像功能	/camera
告警模块	提供紧急告警功能	/warn

整个业务系统分为四大管理模块，分别是概况、电站管理、设备管理、综合管理。

(1) 概况：概况业务根据本地数据库中的电站地理位置信息表和电站数据表以及电站注册信息表等信息，每个电站对应一个集中式光伏发电区域，概况模块统一的展示了每个电站区域的基本信息统计和故障分析情况。概况模块总体分为：地图总览、电站概况和所有电站这三大模块。

(2) 电站管理：电站管理通过分页查询可以显示不同区域下的电站情况，包括电站的地址码、电机数量、离线数量、在线数量、故障数量。电站管理支持在线的视屏监控，可以通过摄像头远程直播电站的实时情况。

(3) 设备管理：设备管理模块可以进行实时设备的信息查询。这个模块是真正意义上的实现了远程运维的功能，对接设备云对下挂设备进行命令发送正是这个模块的核心功能之一。它还具备了实时的电机电流记录，以及通过监控平台、边缘终端实时示数获取的电机运行日志。远程控制电机可以进行电机参数相关的设置，同时可以设置定时的运行任务。对于电机的自动化运转，模块提供了多种模式如：普通模式、暴雪模式等。

(4) 综合管理：该模块主要是对远程运维人员、电站档案等其他辅助事务进行管理。包

含了角色的权限管理和人员档案管理，也可以管理上述模块的摄像头。电站档案的相关管理在新增电站时需要建立。

远程运维后台不仅提供了对接前端操作页面的大量接口，同时后端存在着大量的定时任务、逻辑功能任务、告警监控、命令控制等相关的事件。

2.5.2.2 数据库设计

数据库整体上采用微服务的设计方案，多个服务公用一个数据库集群，不同数据库之间的耦合性较低，防止底层查询效率过低。数据库是否公用主要看不同服务之间的耦合性，系统将不具备依赖或耦合的服务分为一组，每组内的多个服务具备依赖关系，而不同组之间的任意服务都不具备依赖关系。

光伏运维系统共有多个数据库集群分别为：基础数据库集群、业务数据库集群、数据分析数据库集群。其中基础数据库集群负责基础服务的数据存储。业务数据库是整个系统的数据核心，用于存储关键业务数据。数据分析数据库用于存储分析预测服务相关的数据，这些数据大部分是环境数据、天气数据以及图像二进制流数据。下面具体讲解各个模块数据库的设计。

基础服务数据库主要是存储了一些微服务相关的设置配置或者账号管理、单点登录^[33]相关的数据。具体如表 2.5 所示。

表 2.5 基础服务核心数据库及其功能表

表名称	表功能
api_info	微服务 API 网关路由表
base_log	API 调用历史日志
base_oss	单点登录相关
base_menu	用于展示 WEB 界面菜单
base_auth	权限表
base_user	用户管理
base_region	服务器信息表
base_datasource	数据库动态配置表
base_role	系统角色表

基础服务还存储了关于远程更新相关的表以及文件传输服务相关的表。这些基础服务可以使用在任意业务场景下，和具体的业务耦合性不高。

算法分析服务的数据库主要用于存储环境数据和天气信息以及图像的相关的数据，用于功率预测和热斑检测服务，与基础服务以及业务无关，单独拆分成独立服务。其中图像数据使用二进制作为存储。核心表及其功能见表 2.6 所示。

表 2.6 分析预测服务核心数据库及其功能表

表名称	表功能
algo_base	分析预测服务基本信息
algo_current_wea	存储最新天气信息
algo_future_wea	存储未来天气信息
algo_stream	存储光伏环境数据
algo_pic	存储光伏图像数据
algo_result	存储算法分析的结果

业务服务的存储承担着系统的核心功能。业务服务拥有大量的表，表在设计上使用了 Innodb^[34]存储引擎和 Mysiam 存储引擎。对于不变的业务数据，或者不涉及变更的表，系统采用了 Mysiam 作为存储引擎。业务核心表及其功能如表 2.7 所示。

表 2.7 业务核心数据库及其功能表

表名称	表功能
tb_command	分析预测服务基本信息
tb_elec	电量相关信息
tb_station	存储电站相关信息
tb_operate_analysis	指令下发操作历史记录
tb_monitor_config	监控点配置表
tb_gateway	存储网关相关信息

2.5.2.3 逻辑设计

远程运维业务逻辑部分的实现是根据制定的前端接口、定时任务、基本事件来实现的。核心的逻辑如：数据处理、控制下发、电站管理等。

根据系统模块功能的不同，边缘终端上传的数据分为：电机数据、电池数据、环境数据、图像数据、监控数据、天气接口得到的天气数据，这些作为数据的输入，经过系统业务的处理后，进行相关的数据输出。监控数据向监控服务输出、电池数据和电机数据会持久化到数据库并进行前端展示输出。环境数据将输出到分析预测服务平台，进行相关算法的分析。具体的流程图如图 2.22 所示。数据的获取主要是通过定时任务定时的向数据提供方进行查询，并持久化到本地数据库。边缘终端的数据作为监控点向网关传输后进行 JSON 序列化后上报给设备云。监控点上报数据是没有频率的，而是采用差异化上报，一旦监控数据大于设定的变化范围，就会进行数据的上报。

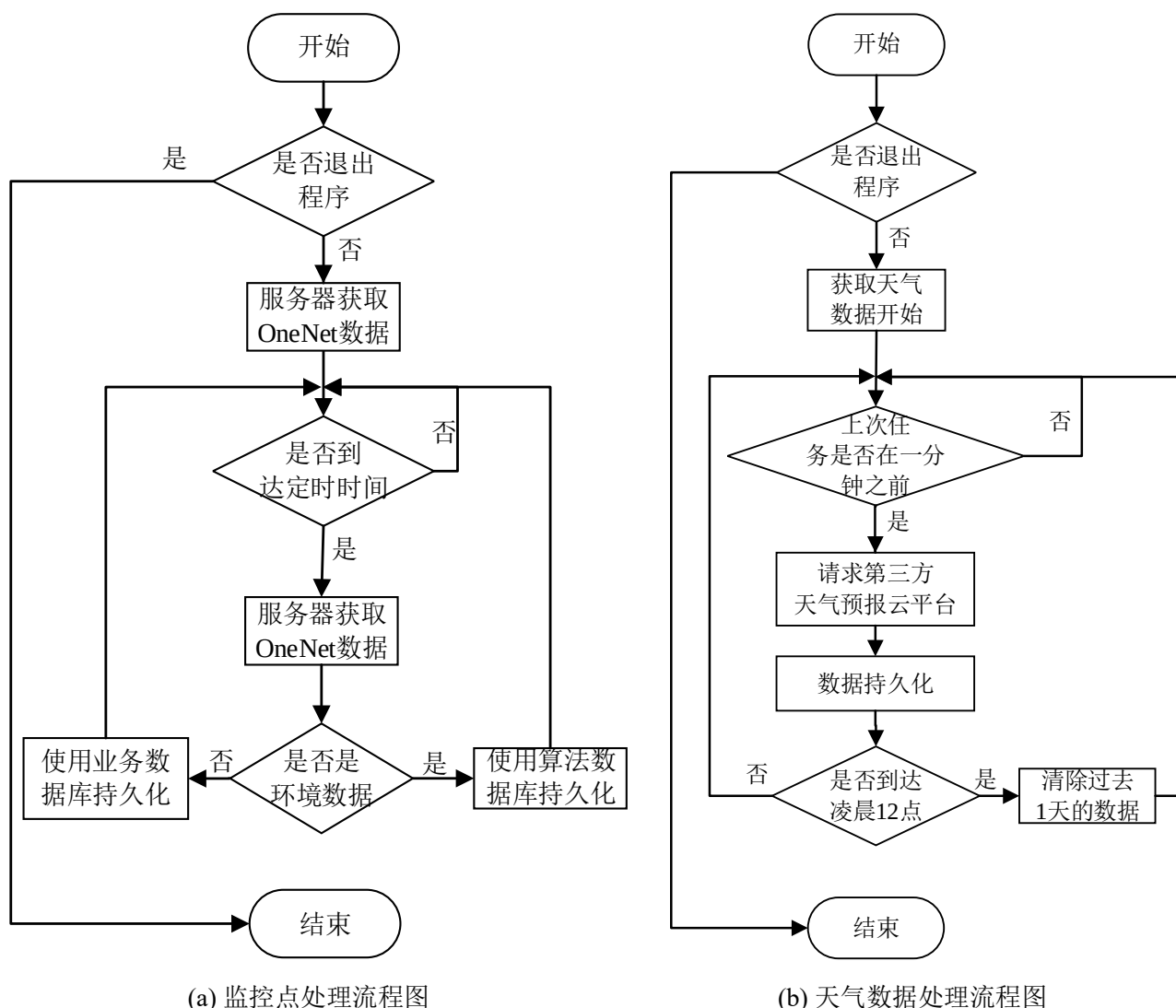


图 2.22 服务端业务数据获取流程图

控制下发功能主要是控制边缘终端控制板的机器人。控制下发的实现又分为：天气预测控制、定时任务控制、直接控制。

天气预测控制模块，会根据天气服务来综合地进行机器人的清洁时间的安排，具体流程如图 2.23 所示：

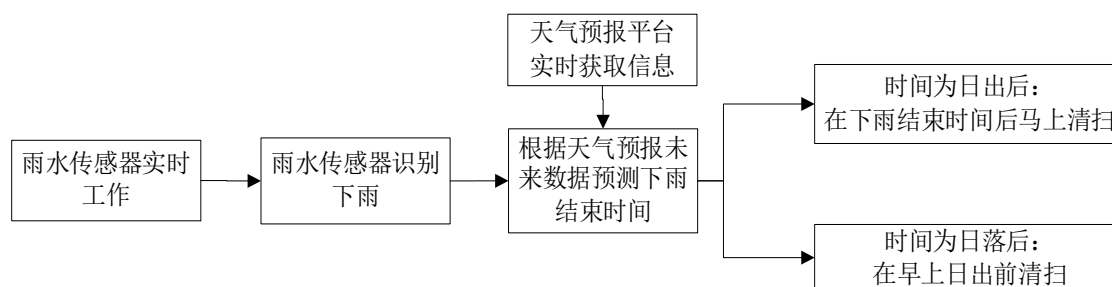


图 2.23 雨水控制示意图

分析预测平台根据雨水传感器识别的下雨指令进行下雨时长判断，由于清洁后短时间的下雨会导致无效清洁，并且下雨会有利于机器人清洁一些污渍，所以最佳的清洁时间安排在雨后。根据天气服务判断下雨时长，这个时长会时刻调整，系统判定下雨时间的持续时长，得出清扫时间位于日落后还是日落前，如果是日落后则在早上日出前进行清扫，如果是日出后，则在下雨结束后马上清扫。

清洁时间的判定也是雨水控制模块的重点，具体功能是根据光照传感器或者辐射传感器获取太阳光的强度，在强度高时不进行清洁，晚间结合天气系统来判断是否进行清洁，具体的实现细节有如下几点：（1）光照或者辐射传感器实时获取光照强度；（2）光照强度在工作范围内则不进行工作；（3）光照强度不足：进行天气判断；（4）如果后面下雨则不进行清理等待雨后进行清理；（5）如果后面不下雨在凌晨太阳升起之前的最低光照下进行清理。

直接控制主要是后端服务通过设备云发送相关指令，网关接受指令后通知控制板进行电机的相应控制。直接控制分为多种模式，其中暴雪模式是大雪挤压光伏板时使用，此时机器人将在光伏板上来回多次运动（为了避免机器人运行到导轨另一端时，清理一端又被积雪堆积，所以需要多次来回运动），直到系统供电不足时，或者通过下发停用暴雪模式才能使得机器人停止运动。普通模式，则是通过设置加刮次数，来进行机器人的运动控制，机器人运动达到了系统设置的加刮次数后，机器人停止运动。

定时任务控制，则是通过前端将相关的任务信息存储到数据库，然后后端服务通过Linux的Cron定时任务调度器来进行任务的定时执行，一旦定时时间到，任务指令将最终被控制板接受并执行。具体流程如图2.24所示。系统可以同时设置多个定时任务。

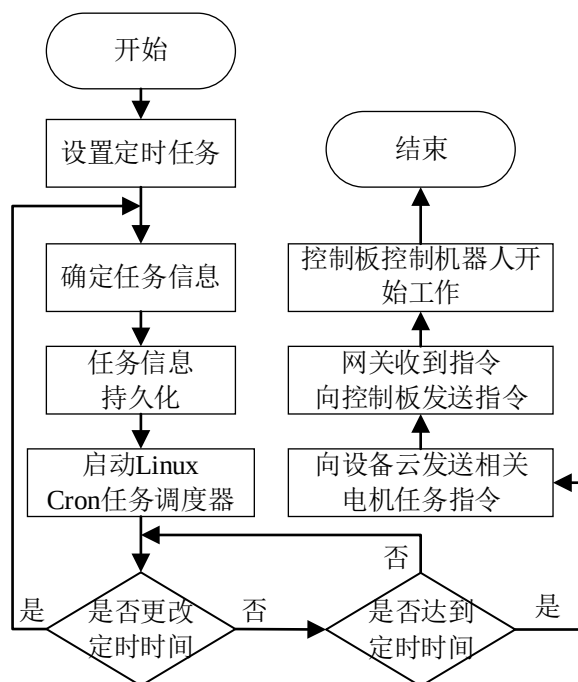


图 2.24 定时控制流程图

控制模块，还可以进行电池参数的相关设置，以及超时工作时间等的设置。这些设置间接来约束控制电机，比如电池相关参数如果不在预期设置范围内，则后台服务会向机器人发送停止电机运动的相关指令，以防止机器人的电机或者电池以及光伏板造成损伤。

电站管理功能则表现为：系统可以同时支持多个区域的管理，每个区域可以拥有多个电站，电站的概念主要是方便多地域光伏集群的管理。通过唯一标识码表示每个电站的信息。同时提供了 GPS 等地理信息。新增一片新地域的集中式光伏系统，则需要人工的将相应的电站信息添加到远程运维系统。后台系统会以电站为维度统计分析不同地域光伏集群的故障率、容错率、效率等重要信息。

2.5.3 分析服务设计

分析预测服务主要由天气预报、功率预测、热斑检测三个服务组成。天气预报服务对接了高德云平台，通过定时任务拉取最新的数据到本地数据库，同时向 WEB 端提供了天气访问接口。通过实时的天气监控，预测是否会对光伏清洁组件操作损坏，比如在恶劣天气：台风、沙城暴、暴雪等情况下，天气服务会通知边缘终端启动防护模式，如暴雪天气下启动暴雪模式，来进行积雪的清理。

雨天刚结束的时候，由于积水的残留有助于光伏清洁机器人更好的清理，同时长时间的连续下雨，光伏机器人无须在中途下雨过程中进行清理。天气服务结合环境传感器的信息和未来天气信息，预测雨停的时机，告知光伏机器人在这个时间上进行清理，防止没必要的清理带来的能源损耗。

功率预测服务，则是结合环境传感器采集到的相关数据，使用基于区间划分的 KNN 功率预测算法，来分析预测系统的功率输出。同时根据预测结果和实时结果的对比可以反向推导光伏板的健康状态，检测出是否有局部受损或者长事件局部的灰尘积淀。此部分将在第三章进行详细的介绍。

热斑检测服务通过红外摄像头采集到指定大小和像素下的图像，通过 Flood Fill 算法识别并标注热斑的区域，通过 XGBoost 算法优化识别的准确率。经过优化的算法在存在干扰光源的情况下，依旧可以有着较高的识别准确率。此部分将在第四章进行详细介绍。

2.6 应用层设计

2.6.1 WEB 前端设计

光伏远程运维系统的操作端主要以 WEB 前端为主。前端在系统中起着承上启下的作用，方便用户和后台进行沟通。前端整体包含了静态界面和动态界面的实现。其中静态界面通过 Nginx 静态映射到后台服务器的相关 Html 文件、图片文件等。而动态服务则是通过调用后端接口进行数据和图像的动态显示。前端的功能界面使用了 Vue 框架进行了实现。

前端大屏界面通过 Echarts、Emap 实现了具备设备状态显示、报警类型统计、数据分布

统计的可视化界面。通过 Html5 和监控摄像头实现了在线直播的功能。

2.6.2 小程序设计

为了方便光伏运维人员操作和监控系统，系统设计了运维小程序，大大地加强了系统地可操作性和便携性。微信小程序通过调用后端接口实现了和前端功能一样的控制展示界面。

2.6.3 系统展示

光伏电站运维系统的整体设计如上文所示，系统的各个部分的实物图和 WEB、小程序功能界面如图 2.25 至图 2.30 所示。



图 2.25 集中式光伏集群



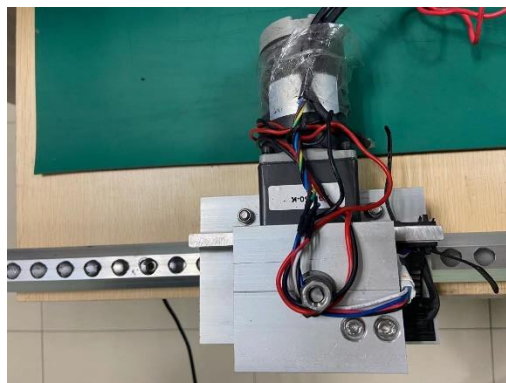
(a) 控制板



(b) 通信板



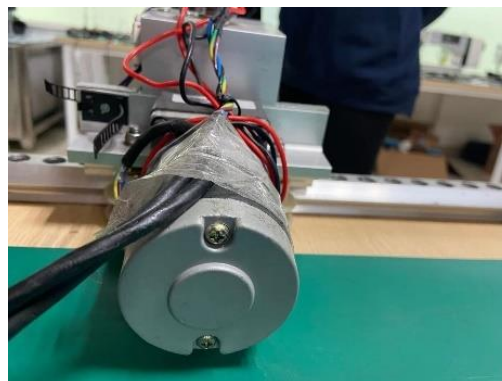
(c) 电源控制器



(d) 电机俯视图



(e) 电机左视图



(f) 电机右视图

图 2.26 光伏运维系统硬件展示图

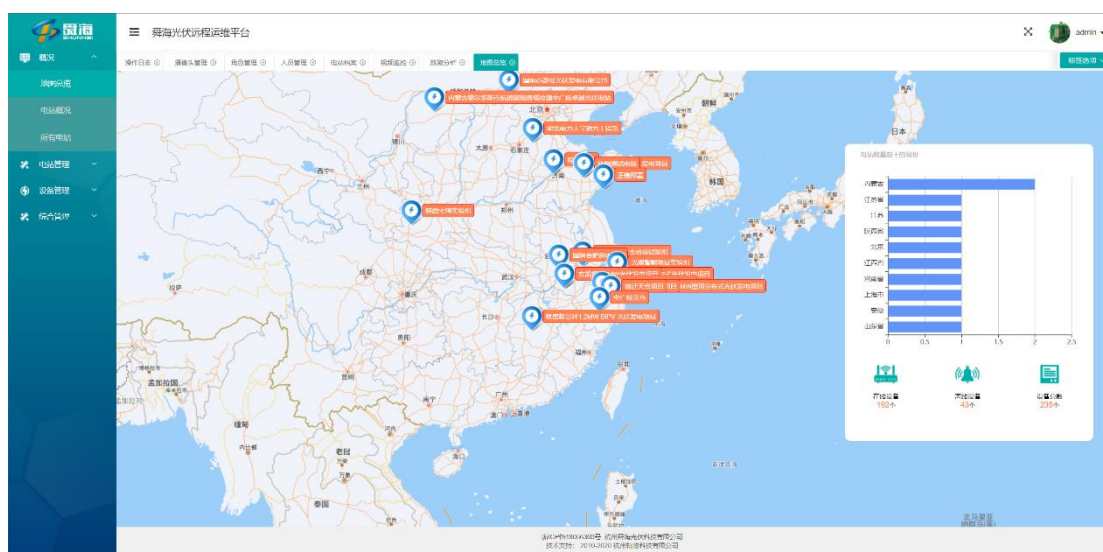


图 2.27 WEB 端光伏机器人远程运维总览界面

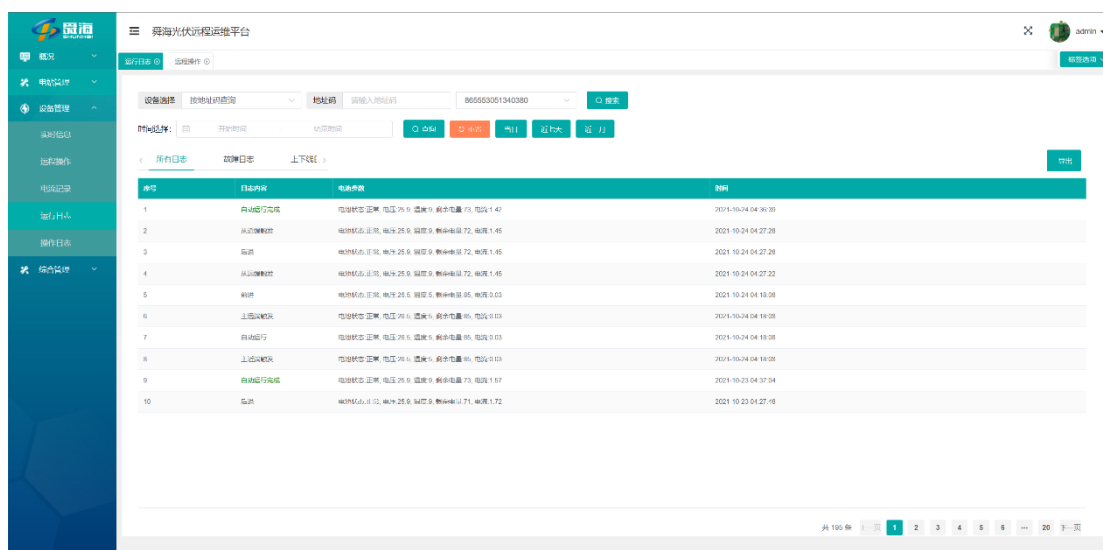


图 2.28 WEB 端远运行日志展示



图 2.29 WEB 端电流监控记录



(a) 小程序登录界面

(b) 小程序故障监控展示界面

(c) 小程序远程控制

图 2.30 小程序端展示图

2.7 本章小结

本章首先对整个系统设计进行了介绍，然后按照边缘终端、后端、应用端的顺序分别介绍了各个端的设计。对边缘终端的硬件和光伏机器人的控制原理进行了介绍。对后端如何与边缘终端进行通信进行了说明，详细介绍了后端架构设计和各个服务的设计。最后对应用层做了简单介绍，并进行了实物展示。

第3章 基于区间划分的 KNN 光伏功率预测设计

作为一种间歇式能源，光伏电站的输出会受自然环境的影响，导致输出不稳定，这会影响到电站的并网。准确地预测光伏的功率输出对电网的运维具有现实意义。同时光伏功率的短期预测值和实际功率值的对比可以诊断出光伏系统的健康程度。光伏系统的输出会受到很多因素的影响。

光伏系统的输出主要取决于它吸收的辐射强度，同时也受天气因素的影响，如环境温度和环境湿度等。不同天气下，光辐射强度也有所不同，这是一个无法决定的重要因素，还有一些可以改善的因素如：光伏板的倾斜角度、光伏板的朝向、摆放间距等。光伏发电系统的能量生产是非线性的，因为它受到天气条件随机性的影响。目前，业界还没有一套统一的公式或者方法来处理非线性数据的预测。在数字化技术下，业界常使用机器学习技术来预测解决集中式光伏系统的功率预测。

随着物联网技术的发展和精密传感器元件的产生，使用传感器监督光伏板环境，从而获取光伏板相关工作参数的场景越来越多了。物联网技术可以实时获取相应参数，并将数据传送至云端，云端可以对数据进行更多的分析和处理及显示，从而更好的改善整个光伏系统的运维和工作环境。

经典的光伏功率预测技术包括机器学习法和电路分析法。现有的电路分析法主要是根据电路模型观测，结合气象信息数据与相关的公式联立求得输出功率，电路分析法存在的问题是有很大的不确定因素，需要在物理条件稳定下才能计算出合适的输出功率。光伏板的老化、损坏、灰尘均可能导致电路分析法失效。

机器学习预测光伏功率已成为目前的主流方法，主要有：基于神经网络、灰色模型、混合模型的预测算法。这些算法存在一个通用的缺点就是拓展性、兼容性差、样本需求很大，并且训练时间会很长，模型复杂。这些问题可能导致系统的结构非常复杂，系统的拓展性很差。

本节介绍使用 KNN 预测光伏系统短期功率输出的设计，通过确定预测精度和 K 值并使用滑动窗口的方式对功率区间进行划分，并评估输入在最优区间划分上的每个区间的欧几里得距离，通过选取最小距离来确定最终输出区间，并使用平均数作为预测结果的输出。模型相对简单，复杂度较低，算法执行较快，并且预测性能相对传统的 BP 神经网络也有所提升。

3.1 整体选型

影响光伏发电的因素由很多，纵观多种因素，其中影响最大的莫过于辐射强度和温度。研究表明，辐照强度对电流的影响可以模型化正比例函数，电压主要收到温度影响，接近于

反比例函数^[35]。对光伏输出的预测分析来说，传统分析法往往是根数温度和辐射强度作为二维变量，通过机器学习算法来进行功率的判断，或者使用物理方法来进行输出预测。在数学法预测输出中，使用线性化功率模型是较为常见的一种方法。

传统分析方式主要着重于考虑温度以及辐射对光伏系统输出的影响，但实际上光伏所处的环境是非常复杂的，空气中的灰尘量、湿度等因素都会影响系统接受光照辐射的效果。并且光伏的倾斜角度会影响光伏系统接受光照的效果。如图 3.1 所示，对于处于不同地理位置的光伏板，为了更好的接受光能，倾斜角的大小以及摆放朝向也是不同的。对于密集的集中式光伏发电集群，前后两个光伏板在不同光照角度上可能会由于距离和倾斜角等因素造成后面的光伏板上产生阴影区域导致整体输出降低^[36]。

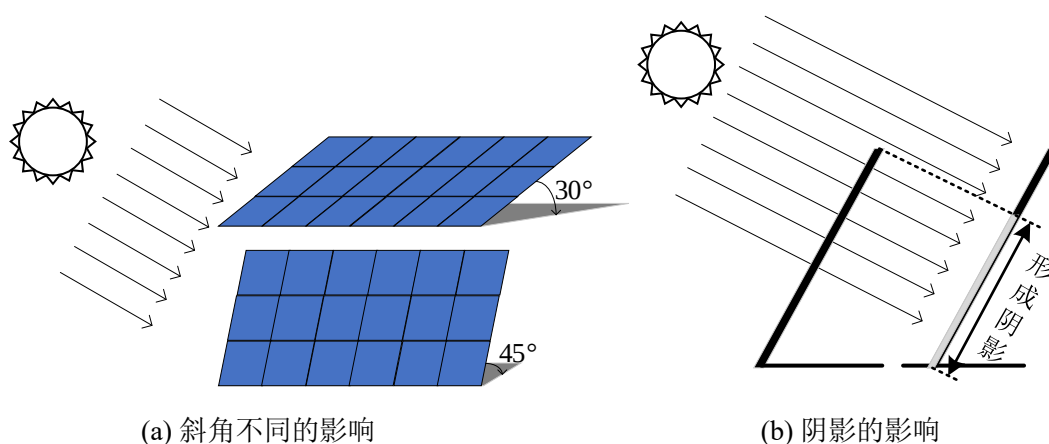


图 3.1 光伏板倾斜角及摆放对受光的影响

上述描述可知，不仅仅是光辐射和温度会影响系统的输出，光伏系统的输出会受多种因素共同作用，本节将使用 KNN 算法来预测在多种因素作用下的集中式光伏系统的输出，并和使用 BP 神经网络来进行输出预测的方法进行对比^[37]。

由于使用物理模型预测光伏输出的方法的鲁棒性和拓展较差，并且实现较为复杂，所以本文采用基于机器学习的统计类方法来预测光伏功率输出。光伏输出预测的经典机器学习方法主要有：最小二乘法 and BP 神经网络等方法。由于大部分光伏系统的输入是时间序列，所以本系统在算法选型上调研了机器学习算法中与时间序列相关的算法，并总结了相关优缺点如表 3.1 所示。

光伏电站运维系统产生的数据是基于时间的监控点数据，数据量大。系统输出受光辐射等环境因素所影响，并且在大部分情况下输出有上下界，处于一个区间范围。光伏运维系统需要实时输出预测功率，对算法运算速度有较高的要求，基于上述特点，本章提出了一种 KNN 预测光伏输出的方案。

表 3.1 时间序列算法优缺点对比

算法名称	优点	缺点
KNN	参数少，训练快速，对突变值不敏感，预测结果不错	内存占用高，不能拟合序列趋势性
BP 神经网络	学习速率固定，非线性映射能力强，有着柔性的网络结构。	算法收敛速度慢，结构选择不一，应用实例与网络规模的矛盾 ^[38]
LSTM	对序列建模有一定优势，实现简单	不利于并行化，效果较差 ^[39]
RNN	对具有序列特征的数据非常有效	只能记忆最近的序列，记忆性差 ^[40]
ARMA	在一个统一框架下适用于所有的时间序列	模型形成需要基于知识的经验判定 ^[41]
SVM	可用于线性/非线性分类。泛化错误低	内存占用高，运行时间长，需要预处理数据，对参数调节敏感 ^[42]

3.2 KNN 介绍

KNN 是一种非参的，惰性的算法^[43]。作为机器学习的经典方法，它凭借着思路直观、实现简单、准确度高等特点广泛运用于工业界的各种属性预测和分类场景之中。作为一种分类算法，KNN 的数据训练过程很快，可以在规定时间给出算法结果，符合后台服务调用的时间延迟。KNN 在大数据量样本下的分类结果相当可观，使用 KNN 算法在对以时间序列为基础的数据预测下有着不错的精确度和较高的可靠性，而光伏发电系统整体产生的数据样本是时间序列样本，这正好给 KNN 提供了更好的用武之地。

KNN 和其他算法相比，其用于分析应用场景的关键点是 K 值的选取。KNN 的原理是：在已知的一个数据集合中，要对数据集合进行分类，数据集合经过算法计算出与之前数据集合的多组相似程度，按照相似度递减的顺序选择前 K 个样本，并通过距离计算公式选择距离最近的结果作为分类结果。

KNN 算法中和 K 值相关联的另一个技术关键点就是距离公式的选取，距离公式有多种经典算法如欧几里得距离、曼哈顿距离、切比雪夫距离、余弦相似度等^[44]。如图 3.2 所示，在二维平面和三维空间中表现的欧几里得距离和曼哈顿距离。在求特征之间的距离之前，还需对不同量纲的特征先进行归一化处理。

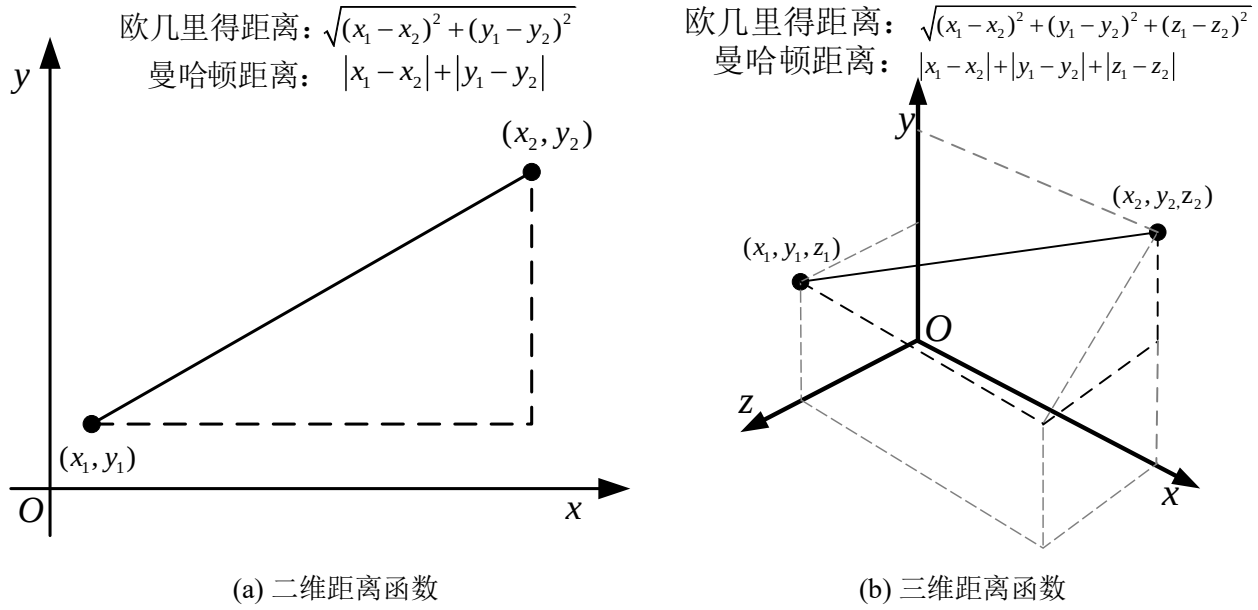


图 3.2 KNN 的距离函数

同理可得, N 维空间的两个向量 a , b 的欧几里得距离 d_{1n} 和曼哈顿距离 d_{2n} 如式 (3.1)、(3.2) 所示:

$$d_{1n} = \left(\sum_{k=1}^n |a_k - b_k|^2 \right)^{\frac{1}{2}} \quad (3.1)$$

$$d_{2n} = \sum_{k=1}^n |a_k - b_k| \quad (3.2)$$

3.3 数据预处理

3.3.1 数据获取

本小节将讲解数据的获取和处理过程。详细流程如图 3.3 所示, 通过边缘终端的嵌入式网关将传感器获得的数据通过 Cat.1 上传至设备云。传感器采集的数据会和云端预报数据以及最近数据进行基本的范围比较来达到异常数据的清洗。后台微服务集群会调用本地天气接口和使用插值算法对数据进行修复。

数据过滤后系统将数据存储在于数据分析服务的 Redis 缓存中。于是, 从光伏板所处环境的传感器所获取的数据通过采集、加工、处理、分析, 得到了用于数据分析的原始数据集。

原始数据集将被用于数据的学习和预测。系统通过建立预测模型在后续工作状态下, 再根据实时环境判断当前环境下的光伏板的输出功率区间。数据获取的基本流程图如图 3.3 所示。

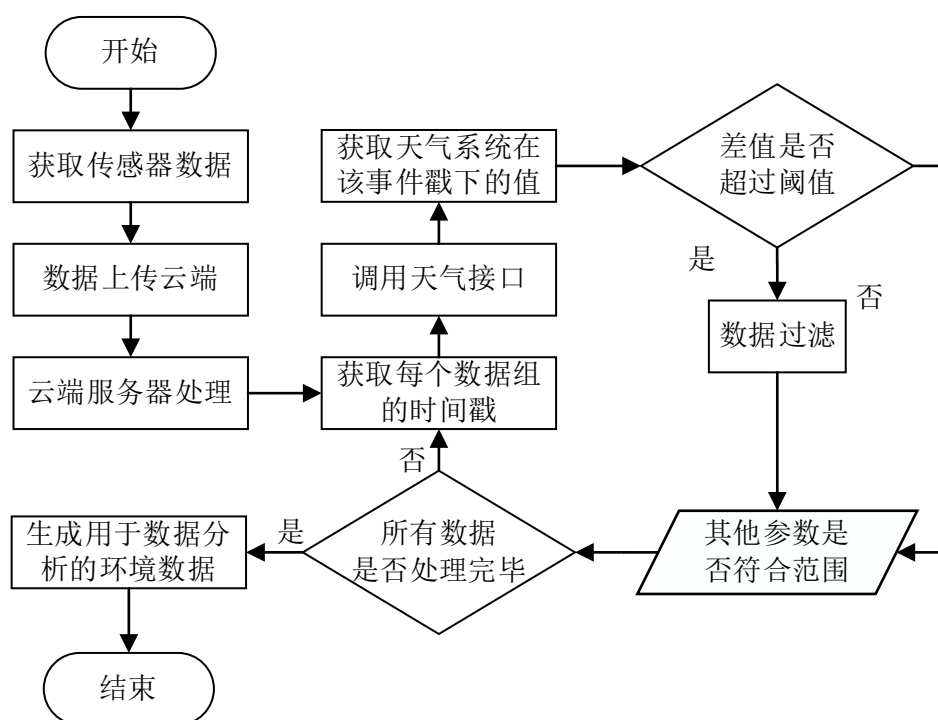


图 3.3 数据集生成流程图

原始数据集主要包括了各种类型的环境数据。环境数据是指边缘终端从各种类型的传感器中收集到的关于光伏集群所处环境的一些数据信息如：气温、相对湿度、光伏组件编号、光伏组件相应编号的数据、位置、辐照度等。具体环境信息对应采集的传感器类型如表 3.2 所示。

表 3.2 传感器功能选型表

传感器	传感器型号	传感器功能
温湿度传感器	DHT1	采集温度湿度数据
光辐射传感器	光电式辐射传感器	采集光照辐射数据
陀螺仪	MPU9250	获取光伏板倾斜角

3.3.2 相关性分析

对采集到的数据进行预测之前，需要对输入特征进行相关性分析^[45]。为了研究环境特征对光伏输出功率的影响，本文在位于 3 月份的杭州地区采集了大量的环境数据和对应光伏的输出功率数据。本文把特征重点放在地球表面的辐照度、温度、相对湿度、倾斜角和风速这几个特征上，通过传感器及小型天气服务接口获取到了温度、湿度、倾斜角、辐照度、风速这五个关键特征以及对应的功率输出。经过测试取样绘制了如图 3.4、3.5 的特征与功率的关系图。

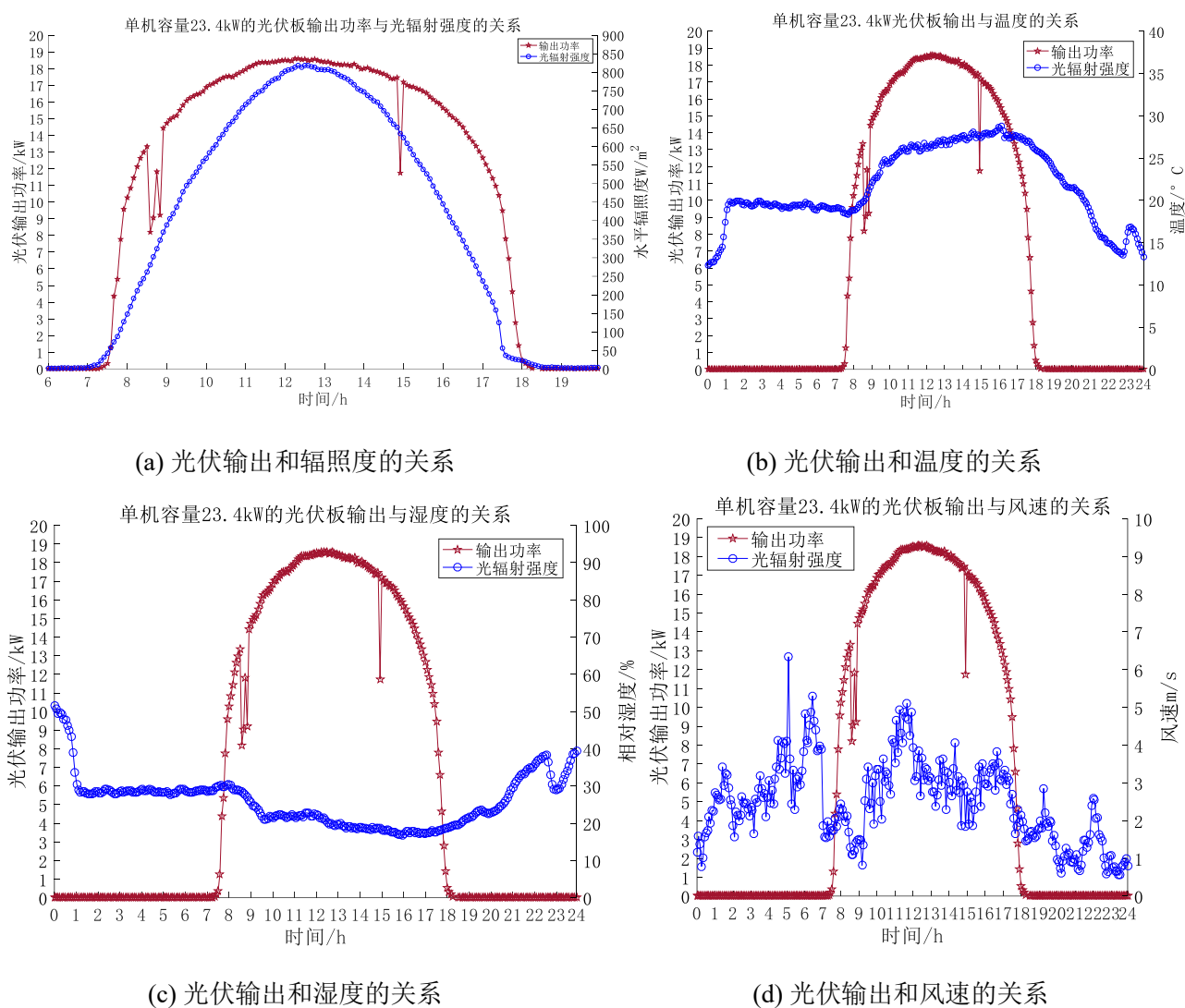


图 3.4 辐照度、温湿度、风速和光伏输出的关系

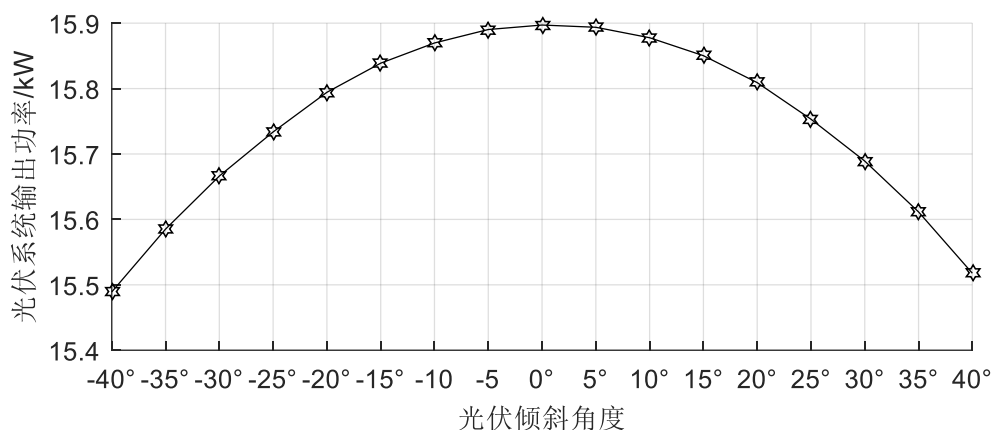


图 3.5 光伏输出和光伏板倾斜角度关系

根据上述几幅图的信息可得到初步的结论：光伏输出功率明显受到辐照度、温度、湿度、

倾斜角这几个特征的影响。其中光伏输出功率和辐照度、温度呈现正相关，而和湿度呈现负相关。功率和倾斜角的关系，主要由倾斜角的绝对值决定，倾斜越多相应的输出功率就会越低。但光伏输出功率受风速的影响较为复杂多变。

为了进一步计算相关特征与功率和特征之间的相关性，本文使用如式（3.3）的方法计算各个特征和输出功率以及特征之间的相关性系数。

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3.3)$$

根据计算结果绘制出了如图 3.6 所示的热力图。

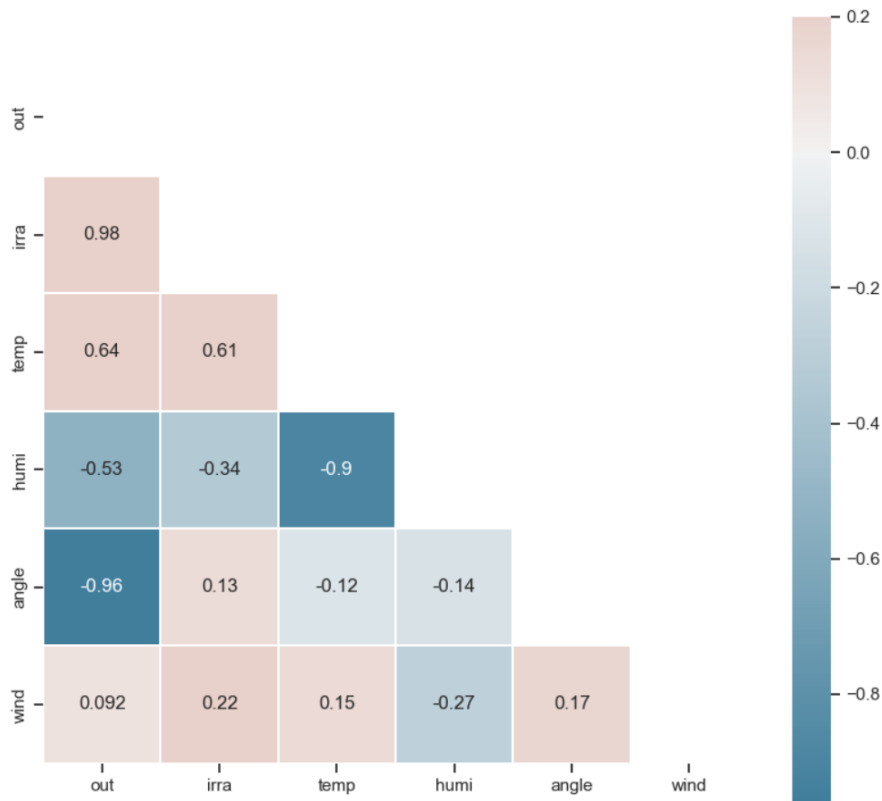


图 3.6 关键特征与功率的相关性热力图

选取相关性的阈值为 0.3，则可以得到关键环境特征为：辐照度、温度、湿度、倾斜角。考虑到温度和湿度的相关性的绝对值达到 0.9，为了防止环境参数的耦合，故去除湿度这一变量。综上所述，系统的输入集特征采用辐照度、温度、倾斜角这三个特征。

3.3.3 数据归一化处理

在使用 KNN 算法进行数据分析之前，首先要确定距离函数，由于各个特征值的量纲不同，这可能导致求出的距离函数的值会偏向某一个特征。如某一天内的温度差只可能会小 10℃，而该天的辐照度差值可能有几百，这就导致求出来的 KNN 距离有较大概率会以辐照

度这个特征为准。为此必须要对数据进行归一化处理。

常见的归一化处理有两种方式^[46]：

(1) 均值方差归一化

该方法能够通过如式（3.4）将原始数据集转化成符合标准正态分布的数据集，使得数据集的方差为 1、平均值为 0。其中 μ 为所有样本数据的均值， σ 为所有样本数据的标准差。

$$x^* = \frac{x - \mu}{\sigma} \quad (3.4)$$

(2) 最值归一化

最值归一化把所有数据映射到区间 [0,1] 之间，如式（3.5）所示，其中 $x_{\min}$ 、 $x_{\max}$ 分别表示特征向量 x 中的最小值和最大值。

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.5)$$

对于上述两种归一化方法，均值方差归一化方法适用于数据本身就符合正态分布的情况，而最值归一化适用于数据比较集中的情况。本文的数据特征正好符合最值归一化的场景，样本中数据的最值不确定的情况下可能会造成归一化结果也不准确，而对于本文的场景来说，温度、辐照度、倾斜角均由明显的上下界，故本文采用了最值归一化的方法对数据进行处理。

3.4 KNN 预测光伏功率的实现

在确定了算法的输入特征并对原始数据集预处理后，得到了用于算法分析的数据集。本节将介绍通过使用 KNN 算法来预测系统短期功率输出的方法。通过环境数据的输入，KNN 将计算出最符合历史结果的功率并输出^[41]。

本文通过光伏系统近一个月的数据获取，将数据集分为训练集和验证集。验证集用于验证算法的准确性和计算各项评估指标。在确定训练集后，系统将对训练集进行训练以评估最佳的 K 值。

在进行算法分析之前需要确定系统预测的精度。由于光伏功率的输出是一个浮点数，本文通过定义精度参数 M 来确定输出功率的精度。 M 参数代表输出功率预测到小数点后的 M 位。本文实验中取 M 为 2，表示功率精度预测在小数点后 2 位。故对于系统的输入功率，数据将会保留到小数点后两位。

使用 KNN 预测功率的总体流程分为以下几步：

- (1) 根据输出功率划分最优功率区间；
- (2) 对训练集进行距离计算；
- (3) 确定预测的功率输出区间；
- (4) 计算并输出最终预测功率。

如图 3.7 所示，为 KNN 实现光伏预测的基本处理框图。

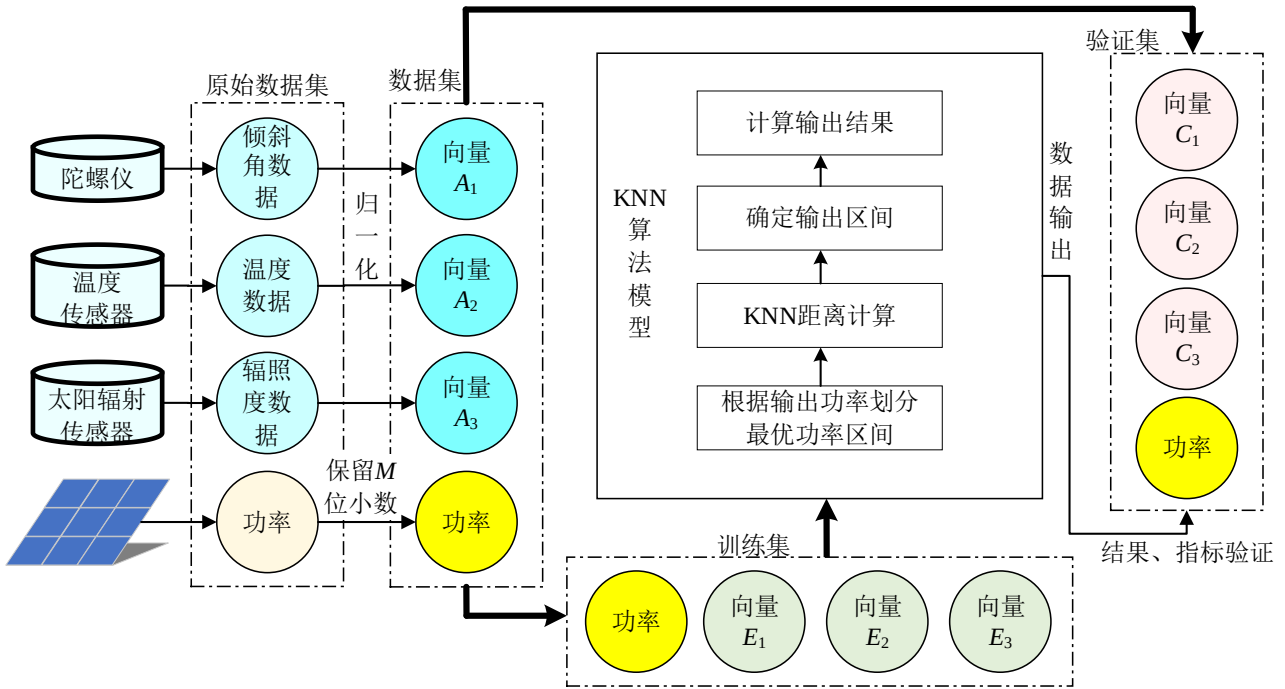


图 3.7 KNN 算法预测输出功率框图

3.4.1 算法总体实现

设边缘终端数据经过归一化等处理后得到了由四个 n 维向量构成的基础训练集 $\{E_1, E_2, E_3, P_1\}$ ，其中向量 $E_i (i=1,2,3)$ 表示三个特征向量，向量 P_1 表示经过精度处理后的实际输出功率。下面将讲述算法的总体实现。

KNN 将光伏输出划分为若干个区间，区间的范围越小 KNN 预测光伏输出的精度将会越高。这主要取决于数据量、精度、以及 K 的大小。如图 3.8 所示，是 KNN 对功率的划分示意图。KNN 将光伏系统的输出划分为若干个以固定值为步长的输出区间。每个区间即为 KNN 预测的一个类别，随着数据量越多，类别也将越多，精度就会越大。并且光伏系统的输出有上下界，故数据量增多只会使得每个类别的数量越多，而不会产生大量的之前未知的数据集。

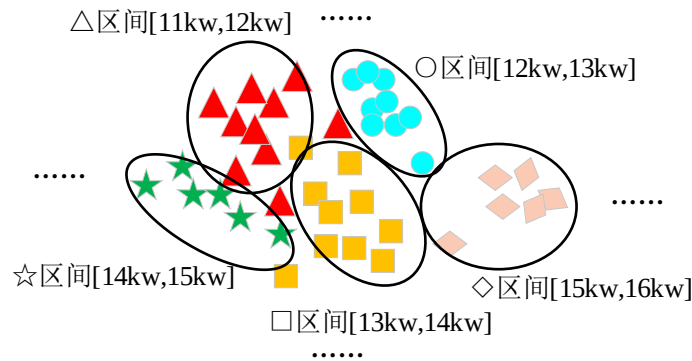


图 3.8 KNN 对光伏输出的分类区间

在已知的分类下，预测输入结果系统的聚类步骤如图 3.9 所示。系统采用欧式距离计算目标输入集合距离最近的 K 个点，如 K 为 4 时，如图 3.9 中计算出 4 个距离最近的点，其中菱形有 3 个，那么 KNN 算法认为目标点集的归属类别就是菱形，KNN 按照归属类型输出归属区间上的功率平均值，即为预测的功率输出。

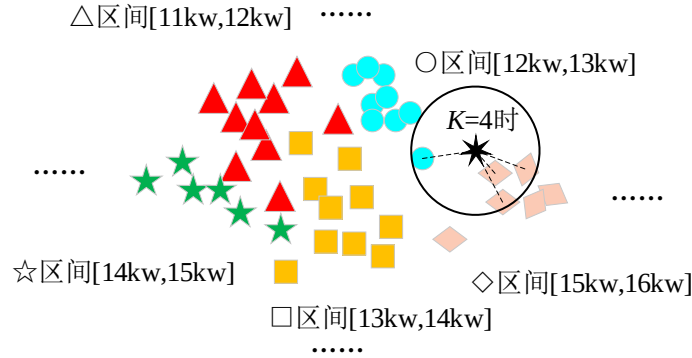


图 3.9 KNN 对光伏输出的归类示意图

3.4.2 最优区间划分的实现

KNN 预测光伏输出的一个核心难点就是功率区间的划分，区间的划分是为了让当前训练集的输出功率能尽量均匀的划分到每一个不同的输出区间，同时要尽量满足每个区间的命中个数大于等于 K 值。定义步长 $step = b - a$ ，表示 $[a, b] (b \geq a)$ 的步长，本文对区间的划分是使用固定距离从 P_1 的最小值 $\min(P_1)$ 开始，以固定间隔为区间步长来划分区间的。对向量 P_1 使用固定步长 $STEP$ 经过划后得到区间集合 S ，其中第 i 个区间 S_i 满足如式 (3.6)，其中 $p_{\min}$ 和 $p_{\max}$ 分别是向量 P_1 中的最大、最小值， $0 \leq i < \text{len}(S)$ ($\text{len}(S)$ 表示 S 的长度)。

$$S_i = \begin{cases} [p_{\min} + i \times STEP, p_{\min} + (i+1)(STEP)], & p_{\max} \geq p_{\min} + (i+1)(STEP) \\ [p_{\min} + i \times STEP, p_{\max}], & p_{\max} < p_{\min} + (i+1)(STEP) \end{cases} \quad (3.6)$$

以上将区间的划分问题就转换成了如何确定步长的问题。对于 P_1 来说，可以有多种步长划分方案，本文通过一个步长集合来记录了所有可能的有效步长，并根据相应划分结果选择最优的步长进行划分。选择最优区间划分一共包含如下四个步骤：

- (1) 计算起始步长、终止步长、步长偏移量；
- (2) 确定步长列表；
- (3) 根据步长列表确定相应的区间划分方案；
- (4) 评估并获取最优区间。

对应的步长划分流程图如图 3.10 所示：

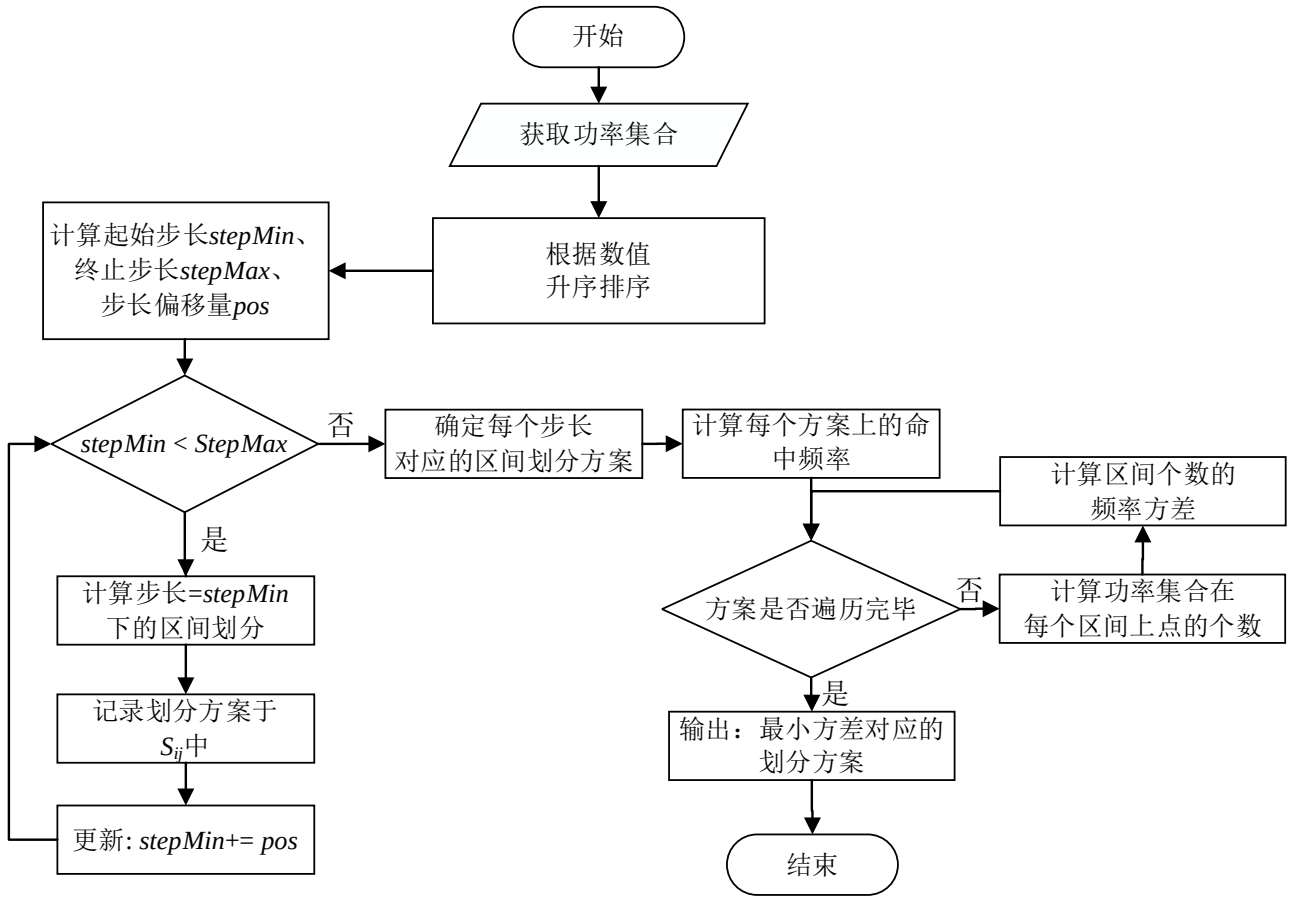


图 3.10 步长划分流程图

(1) 计算起始步长、终止步长、步长偏移量。

对训练集 $\{E_1, E_2, E_3, P_1\}$ 中的 P_1 进行排序得到升序向量 P_{sort} ，步长的选取应当使得最终划分后的区间尽可能平均的容纳下 P_1 中的元素，并且由于最终 KNN 要根据前 K 个值计算距离，区间上命中的 P_1 中的元素个数应当大于等于 K 。根据上述性质定义步长选择区间为 $[stepMin, stepMax](stepMax \geq stepMin)$ ，其中 $stepMin$ 表示起始步长， $stepMax$ 表示终止步长，则取 $stepMax = p_{max} - p_{min}$ 。最小步长要满足划分后的所有区间中，最小的区间所命中的 P_1 中的元素个数要大于等于 K 。 P_{sort} 中相邻 K 个元素的差记为 $dif_i = P_{sort}(i + K - 1) - P_{sort}(i)$ 。使用滑动窗口从 p_{min} 遍历到 p_{max} ，维持窗口长度为 K ，即最少窗口中有 K 个元素，在遍历过程中记录 dif_i 的最大值即为 $stepMin$ 。在 $stepMin$ 的划分下，最小的区间也必定最少包含 P_1 中的 K 个点。

对于步长的偏移量，记为 pos ，为了遍历中不遗漏任何一种情况，取 pos 为系统的精度，根据精度参数 M 可得： $pos = 10^{-m}$ ，对于保留两位小数的精度下， $pos = 0.01$ 。

(2) 确定步长列表。

根据(1)中的 $stepMin$ 、 $stepMax$ 、 pos 可得到最终需要遍历的步长列表如式 (3.7) 所示：

$$sl = \{stepMin, stepMin + pos, stepMin + 2 \times pos, \dots, stepMax\}. \quad (3.7)$$

(3) 根据步长列表确定相应的区间划分方案。

根据公式 (3.5) 的步长向量，遍历所有的向量选择可能，结合公式 (3.4) 分别计算出相应的步长对应的区间划分方案，得到如式 (3.8) 所示：

$$S_{ij} = \begin{cases} [p_{\min} + i \times sl_j, p_{\min} + (i+1)sl_j], & p_{\max} \geq p_{\min} + (i+1)sl_j \\ [p_{\min} + i \times sl_j, p_{\max}], & p_{\max} < p_{\min} + (i+1)sl_j \end{cases} \quad (3.8)$$

式中 S_{ij} 表示在以 sl_j 为步长下第 i 个区间的范围， $0 \leq i < \text{len}(S)$ ， $0 \leq j < \text{len}(sl)$ 。（ $\text{len}(S)$ 表示 S 的长度， $\text{len}(sl)$ 表示 sl 的长度）。

(4) 评估并获取最优区间。

评估最合适的输出区间，需要进行输出区间命中统计，在步长集合 sl 下对于每一个步长 sl_j 对应的区间划分为 S_{ij} ，统计在此步长下的每个区间中 P_1 中元素的命中个数，具体统计方法如下：

定义频率矩阵 $freq$ ， $freq_{ij}$ 表示 P_1 中的元素在步长等于 sl_j 的情况下，在区间 S_{ij} 出现的次数。在此使用标准差来计算数据的偏离程度，对每个向量 $freq_i = \{f_1, f_2, f_3, \dots, f_i\}$ 通过如式 (3.9) 计算出每个步长下的标准差，

$$\sigma = \sqrt{\frac{\sum_{v=1}^n (x_v - \bar{x})^2}{n}}, \quad \text{其中 } \bar{x} = \frac{\sum_{t=1}^n x_t}{n} \quad (3.9)$$

最终选取标准差最小的步长作为最终的区间划分标准。得到以 $STEP$ 为划分步长的输出区间集合 S_{out} 。

3.4.3 结果输出实现

下面进行最终的输出预测。对于整个结果处理的流程如图 3.11 所示。

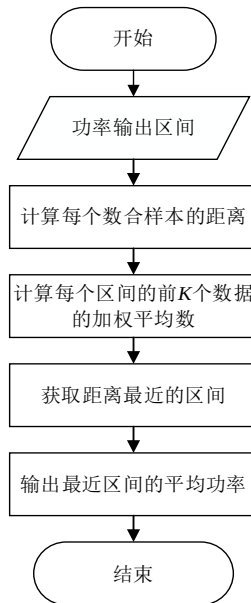


图 3.11 KNN 结果计算流程图

最终的结果处理主要包括如下两个步骤：

- (1) 输入数据的距离计算算法;
- (2) 输出数据的平均归一化算法。

设当前环境下的要判断的输入为 $\{c_1, c_2, c_3\}$ ，假设 P_1 对应的输入为矩阵为 $\{A_1, A_2, A_3\}$ ，分别由 3 个长度为 n 的向量构成。表示辐照度、温度、倾斜角经过归一化等步骤后得到的向量。

步骤(1)中，在 S_{out} 的区间划分下形成 T 个区间如式 (3.10) 所示：

$$S_{out} = \{[t_0, t_1], [t_1, t_2], \dots, [t_{T-1}, t_T]\} \quad (3.10)$$

采用欧几里得距离依次在每个区间上计算该区间上所有特征向量到当前向量的距离。得到距离向量 D 如式 (3.11) 所示：

$$D = \left\{ \sqrt{\sum_{i=1}^3 (A_{i0} - c_i)^2}, \sqrt{\sum_{i=1}^3 (A_{i1} - c_i)^2}, \dots, \sqrt{\sum_{i=1}^3 (A_{im} - c_i)^2} \right\}. \quad (3.11)$$

在对应的 T 个区间分别选取每个区间上的 K 个距离最小的点，选取上述距离值最小的 K 个点，对于区间 $[t_i, t_{i+1}]$ ， $0 \leq i \leq T$ 上的 N_i 个点，按照上述算法求得的距离，选取最近的 K 个点： $N_{i1}, N_{i2}, \dots, N_{ik}$ ，最终输入数据到该区间的平均距离使用的是算数平均数作为输出：平均距离 $\overline{dis}$ 如式 (3.12) 所示：

$$\overline{dis} = \frac{\sum_{v=1}^K N_{iv}}{K}. \quad (3.12)$$

对所有 T 个区间的距离进行比较，找到平均距离最小的一个区间，该区间即为步骤(1)的输出区间。

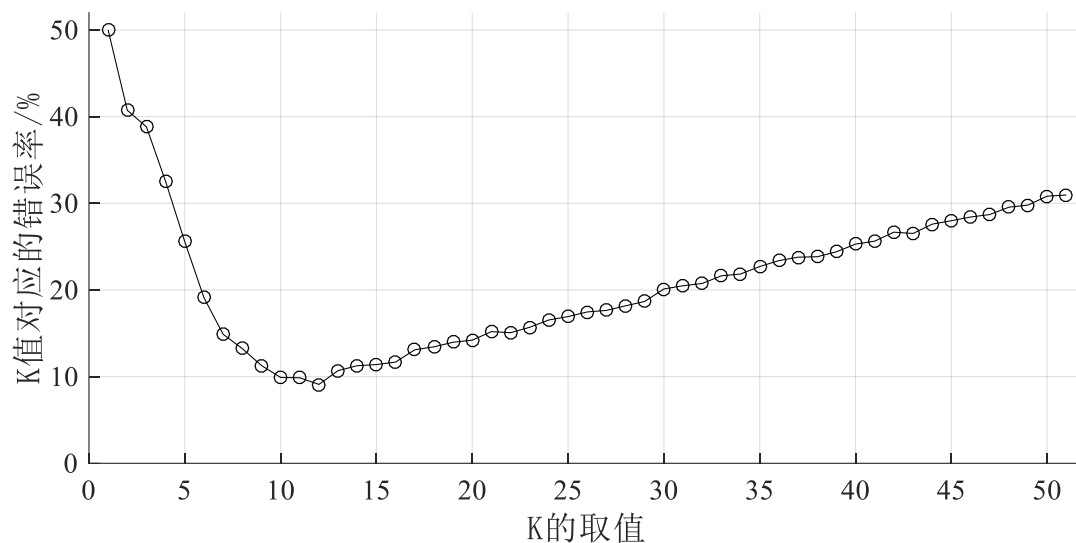
步骤(2)采用的输出是取当前区间的前 K 个距离最小的点的对应输出功率进行算数平均数的运算得到的。

3.5 模型测试与评估

本节将展现 KNN 光伏预测模型下的实验结果与评估，包括功率预测结果与指标评估。同时使用网格搜索法来优化 K 值^[47]。

3.5.1 网格搜索法优化 K 值

实验数据选择了杭州某地区 2021 年 10 月中旬至 2021 年 11 月的光伏数据作为测试。为了测试数据的准确性和优化 K 值，本文使用网格搜索法对 K 值进行网格搜索，并根据不同 K 值选择的误差率得到了如图 3.12 所示的 K 值选取曲线。

图 3.12 K 值的网格搜索统计图

从上图可以看出，当 K 值选取为 11 左右的时候，系统预测的光伏输出错误率达到最低值。故本系统下的 KNN 算法的 K 按照最低误差 K 为 11 进行选取。

3.5.2 KNN 预测结果与评估

系统采集了光伏板在白天 7 点到夜间 7 点的数据，以每秒一个监控点的数据样本分别采集了对应的输入特征值和实际输出功率，并进行为期一个月的数据收集作为本节的训练集。表 3.3 展示了以某天的几个样本数据作为测试集下，使用 KNN 预测后的预测输出、真实输出和对应的误差率的对比：

表 3.3 光伏输出功率 KNN 算法预测输出对比

样本号	实际输出	预测输出	误差率
1	0.00	0.00	0.00%
2	3.10	5.20	67.77%
3	5.10	5.61	10.00%
4	5.60	5.81	3.75%
5	6.23	6.25	0.37%
6	6.90	7.23	4.78%
7	7.12	6.69	6.03%
8	8.91	10.14	13.80%
9	8.99	9.53	6.00%
10	9.11	8.32	8.67%
11	9.34	10.13	8.45%

续表 3.3

样本号	实际输出	预测输出	误差率
12	9.91	9.41	5.04%
13	10.12	10.00	1.18%
14	10.89	10.89	0.00%
15	11.13	10.89	2.15%
16	12.21	13.91	13.90%
17	12.99	13.81	6.31%
18	13.69	12.89	5.84%
19	14.91	15.10	1.27%
20	15.34	15.67	2.15%

为了直观地反映倾斜角对输出预测地影响，如图 3.13 绘制了倾斜角为 0° 、 20° 、 40° 下 KNN 预测光伏输出地对比曲线。从图中可以看出结果数据随着倾斜角地增大而减小符合预期。

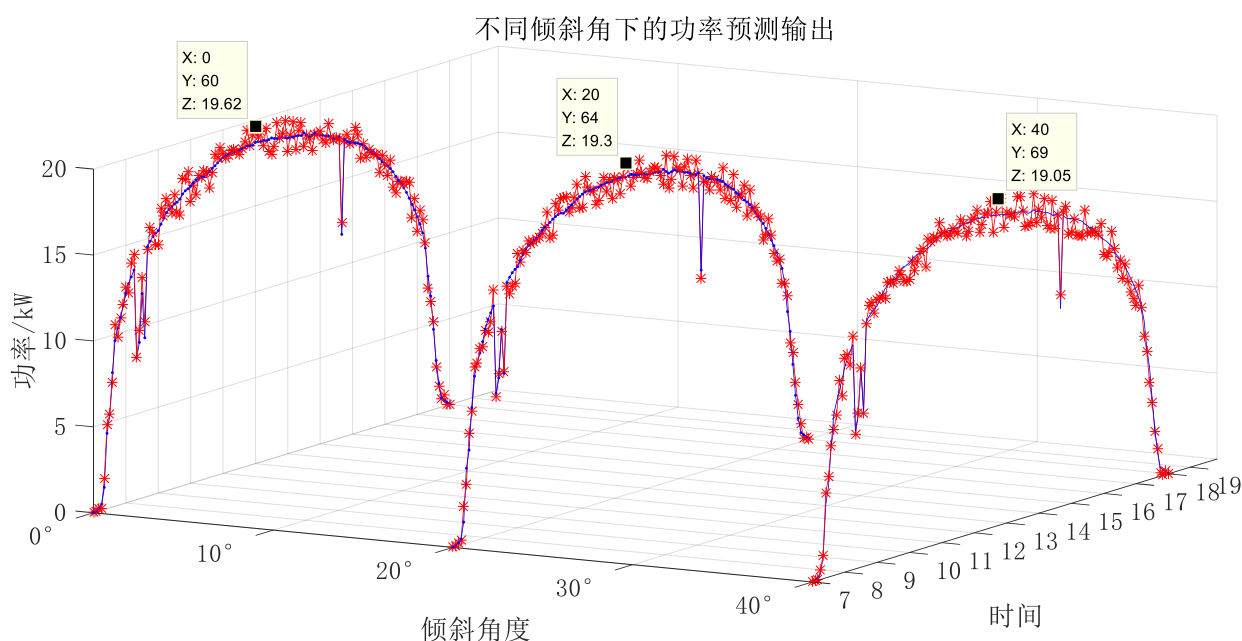


图 3.13 不同倾斜角下的 KNN 光伏输出预测对比图

本系统在使用 KNN 算法进行输出预测的同时，并行地使用了文献[37]的传统 BP 神经网络算法为实验对照组，为了更好地对比下图描绘了两种算法在使用某一天的数据为测试集下的功率输出对比曲线如图 3.14 所示。

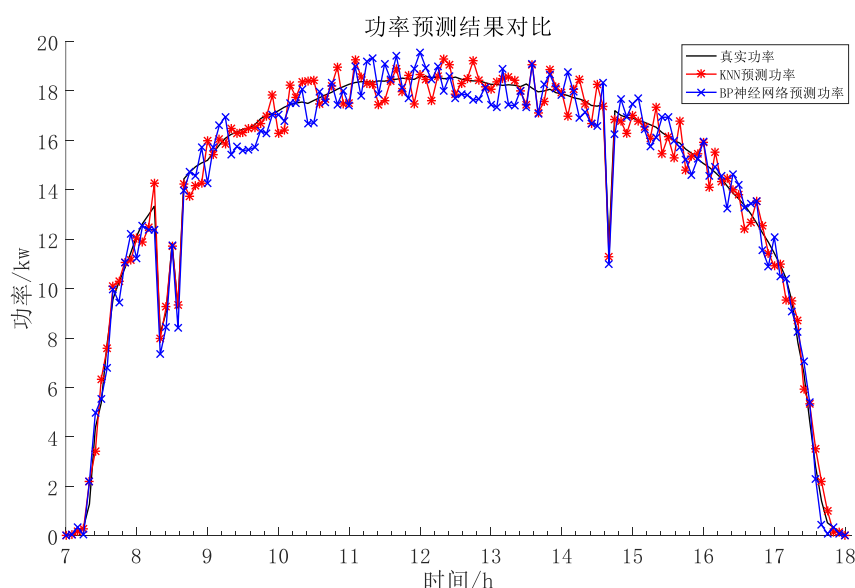


图 3.14 在倾斜角为 30° 情况下的 KNN 光伏输出预测对比图

在上述数据的基础下，使用了经典的平均绝对百分比误差（Mean Absolute Percentage Error, MAPE）和均方根误差（Root Mean Square Error, RMSE）为标准衡量算法来判断预测精度^[48]，并得到了表 3.4。MAPE 和 RMSE 的计算公式如式（3.13）、（3.14）所示。其中， y_i 为实际值， y_i 为算法输出值， n 为样本的个数。

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - y_i}{y_i} \right| \quad (3.13)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_i)^2} \quad (3.14)$$

表 3.4 KNN 与 BP 神经网络的 MAPE 和 RMSE 对比

测试模型	MAPE	RMSE
KNN 模型	7.83%	0.5053
BP 神经网络模型	9.81%	0.6915

通过上表可以看出基于 KNN 的光伏预测效果比基于 BP 神经网络的预测效果稍好，并且 KNN 算法具有很强的输入；适应能力，拓展相比 BP 神经网络更好。

3.6 本章小节

本章先介绍了系统的整体选型以及 KNN 算法的实现原理，并对常用的时序序列的算法进行了对比。同时明确了光伏系统对输出有影响的输入环境变量，确定了基本的影响关系。文中详细地介绍了基于 KNN 实现光伏输出预测的算法流程，主要介绍了基于贪心和滑动窗口确定最优区间划分的过程。通过网格搜索优化了 K 值，最后通过误差计算公式得到算法模型的 MAPE 为 7.83%，RMSE 为 0.5053，并给出了不同倾斜角下模型的输出预测对比图。

第4章 基于 XGBoost 优化的 Flood Fill 热斑检测设计

热斑检测问题一直是困扰光伏企业的一个难题。通过机器学习技术检测热斑能够大大降低企业对光伏板运维和检测的成本，提高识别效率和准确度。针对行业内现有检测方法的缺点和局限性，本章设计了一种基于 Flood Fill 结合 XGBoost 的热斑检测方法。

4.1 光伏组件热斑形成的原理

4.1.1 光伏组件的热斑效应

光伏板的热斑效应指的是，光伏组件在一定条件下由于物体遮挡或者内部电子元器件损坏形成的局部发热区域^[49, 50]，红外成像会形成一个高亮度的红色热斑，如图 4.1 所示。

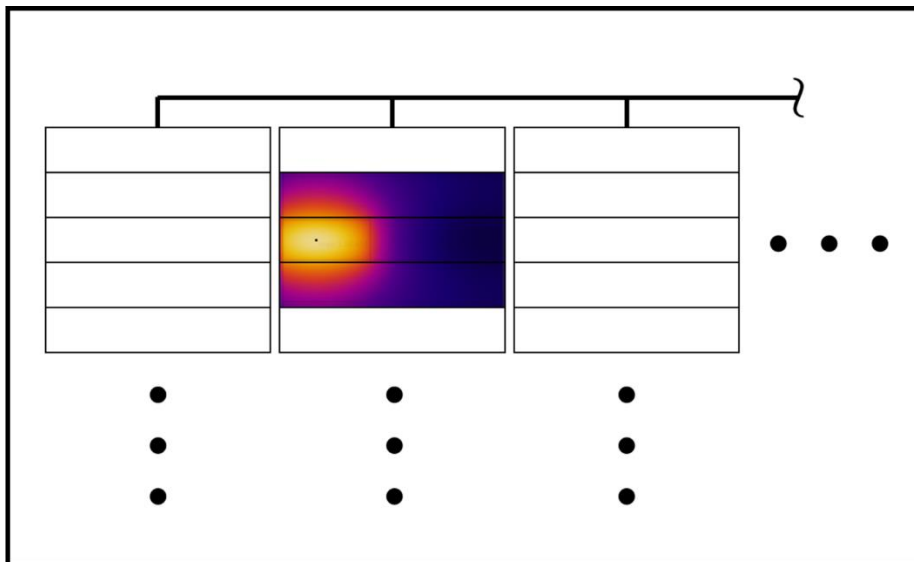


图 4.1 红外成像显示下光伏板上的热斑

热斑效应是光伏业界内非常热门的话题，热斑会造成光伏系统的输出功率减少，严重的甚至会导致光伏板烧坏和发生火灾。热斑的形成分为两类：第一类是由于光伏器件自身造成的损坏，这类热斑需要及时定位并进行修复；另一类则是由于局部被物体遮蔽或物体倒影遮蔽造成的^[51]。

4.1.2 热斑形成的原理

光伏组件以硅电池为主，通过半导体的光电效应进行发电。光伏组件的热斑效应主要是由于辐射强度的不匹配造成的。光伏电池板在长时间的遮蔽下会导致电池退化，进而影响光伏系统的性能。

工业界中广泛使用 Bishop 模型对光伏板进行数学建模^[52, 53]。如图 4.2 所示，Bishop 模型建立的是基于二极管反向特性的光伏模型等效电路，对应的电路公式如式（4.1）所示：

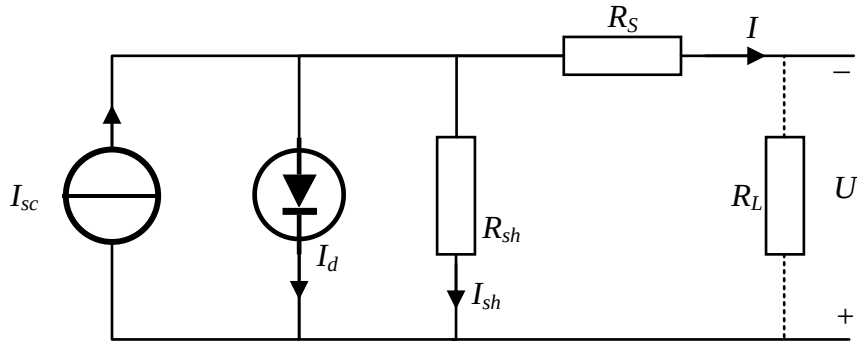


图 4.2 Bishop 模型模拟光伏电池板反向特性

$$I = I_{sc} - I_d \left\{ \exp \left[\frac{q}{AkT} (U + IR_s) \right] - 1 \right\} - \frac{U + IR_s}{R_{sh}} \quad (4.1)$$

式中： I 是光伏板电池输出电流； U 是光伏板电池输出电压； R_s 是串联电阻； R_{sh} 为并联电阻； I_d 为反向饱和电流； I_{sc} 为光生电流； q 为电子电量； T 为电池温度； k 为波尔兹曼常数； A 为拟合常数。

由于光伏组件的等效模型中存在二极管，并且二极管工作在反向特性的状态，如果二极管发生雪崩击穿，则会造成反向电流剧增，由 Bishop 模型知输出电流 I 会剧增并且会反向，这样整个光伏电池板会从功率输出方变为反向的功率消耗方，光伏电池板做负功，区域开始形成局部热点，这就是热斑效应。

由此可知热斑形成的一个可能原因会是局部器件的损坏，这种其情况下系统需要及时发现并进行维修替换。

实际上，光伏热斑的形成并不是都在器件产生故障的时候才会发生，对公式 (4.1) 进行分析可知，当产生反向电流的时候，还有一种情况就是光生电流 I_{sc} 不够大。在 Bishop 模型中^[53]，由式 (4.2) 可知当光照强度不足时 I_{sc} 会变小，则可能会导致电池板发热。

$$I_{sc} = \left[I_{sc_STC} + (C_{Ti} (T_c - T_{STC})) \right] \frac{G_i}{G_{STC}} \quad (4.2)$$

其中 G_i 是入射辐照度， C_{Ti} 是热流系数， T_c 是电池温度， I_{sc_STC} 是短路电流， T_{STC} 是电池温度， G_{STC} 是电池在标准试验条件下的入射辐照度。

入射辐照度变小的可能场景为光伏板被物体遮挡或者光伏板上有污渍（如鸟屎等）。

4.2 基于红外成像的热斑检测方法

本节介绍基于红外成像进行热斑检测的总体流程。红外热像仪能够通过向物体辐射红外线的方式获取物体的温度分布情况图像，并且温度越高部位红外成像就会越亮。根据热斑形成的特性可知，热斑部分的红外成像将会远远亮于正常部位的红外成像。

如图 4.3 为热斑检测系统获取成像图的处理流程。红外热像仪将获取的红外图像，通过灰度变换得到了一张灰度图。灰度图上的每一个像素点都对应着 0 到 255 之间的一个数值，

且灰度值越大，亮度越亮。在 640*480 分辨率的红外成像图像，经过灰度变换后以及滤波后可以得到一个长度为 640 宽度为 480 的灰度矩阵。为了后续算法处理的数据统一，图像获取的距离和像素均保持一致。

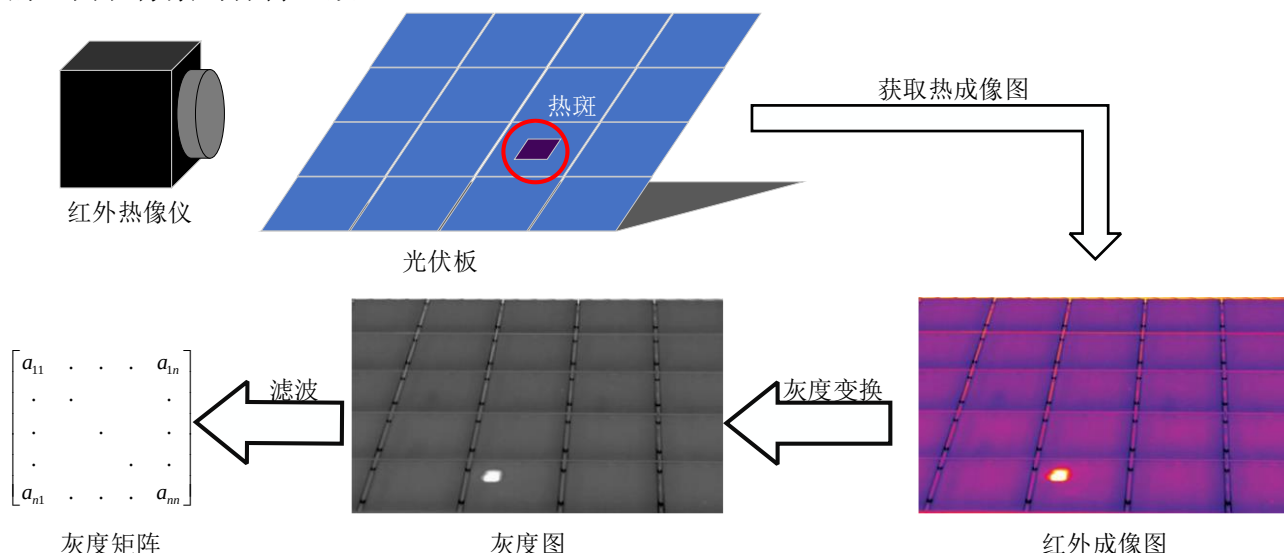


图 4.3 热斑检测获取灰度矩阵过程

传统的光伏热斑检测系统中大多使用边缘检测中的技术来对图像热斑进行识别，但传统方法无法精确地排除光源造成的虚警对热斑检测的干扰，如图 4.4 所示，当太阳光等光源照射在光伏板上，在经过红外成像下也会形成一个热点光斑，此热斑一般称为“虚警”。此时传统的边缘检测方法会将虚警误判为热斑。文献[22]在红外成像的基础上，通过图像变换并使用了 SVM 提取了子图像块的数据特征，利用图像分块技术划分出热斑区域，但也无法避免虚警带来的干扰。

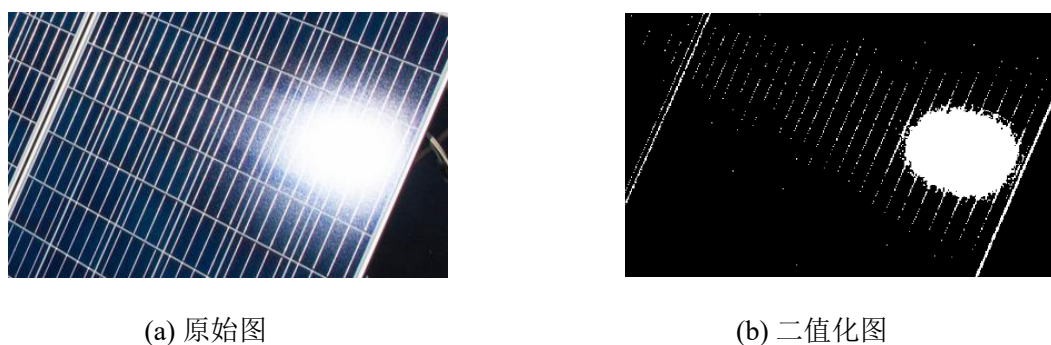


图 4.4 传统识别方法误判光源虚警

本文提出了一种基于 XGBoost 优化的 Flood Fill 热斑检测方法，通过对热斑形态进行学习，防止系统对光源虚警这类非热斑进行误判。具体流程如图 4.5 所示。

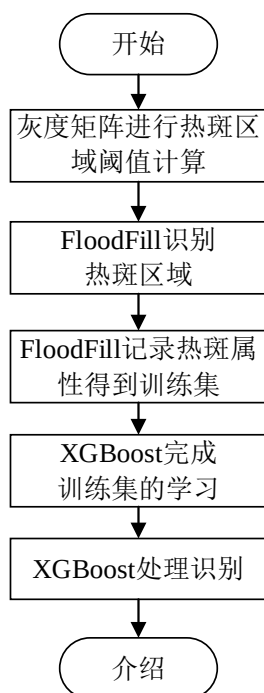


图 4.5 XGBoost 优化的 Flood Fill 热斑检测流程图

4.3 Flood Fill 的热斑检测方法实现

4.3.1 Flood Fill 算法介绍

Flood Fill 算法也被称为种子填充算法，使用场景为：联通多维数组中与给定目标区域相连接的区域并且具备属性匹配的特点，属于一种特殊的遍历算法。该算法的核心思路类似洪水从某一范围蔓延到所有能达到的范围，故以此名相称^[54]。

Flood Fill 算法一般会涉及到三个属性：起始节点、目标颜色、替换颜色。其中颜色通常以具体数值表示。算法通过搜索所有的节点，来寻找矩阵中通过目标颜色和起始节点相连接的所有节点，并将其更改为替换颜色。

在向周围遍历的过程中会根据联通性的不同分为两种不同形式：八联通和四联通。如图 4.6 所示。四联通的相邻节点规定为上下左右，八联通的相邻节点增了斜方向的四个方向。

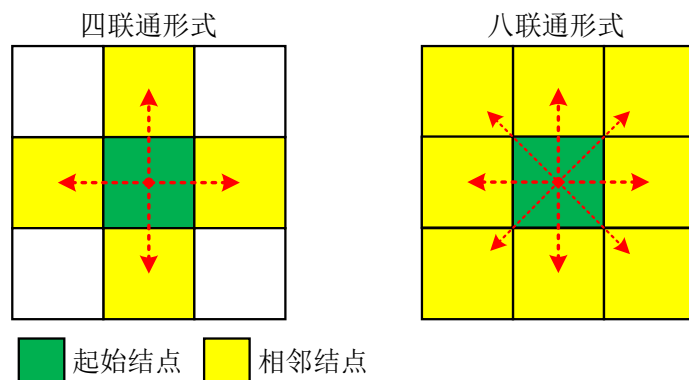


图 4.6 光伏板区域的灰度图矩阵

算法的实现一般有三种：基于递归的深度优先遍历、基于队列的广度优先遍历、基于栈的深度优先遍历。表 4.1 对比了三种实现的优缺点：

表 4.1 Flood Fill 热斑检测测试结果

实现方式	优点	缺点
基于队列的广度优先遍历	运行速度比递归方式快	空间占用相对较大
基于栈的深度优先遍历	不会发生递归栈溢出的问题	实现比递归方式复杂
基于递归的深度优先遍历	实现方便、简单	递归有额外的时间开销

4.3.2 Flood Fill 解决热斑问题

由于热斑区域的灰度值与正常光伏板颜色区域的灰度值不同，从而形成不同的区域分割，灰度值越大表示该点越亮，说明是热斑的可能性就越大。使用 Flood Fill 解决热斑问题的基本流程图如图 4.7 所示。

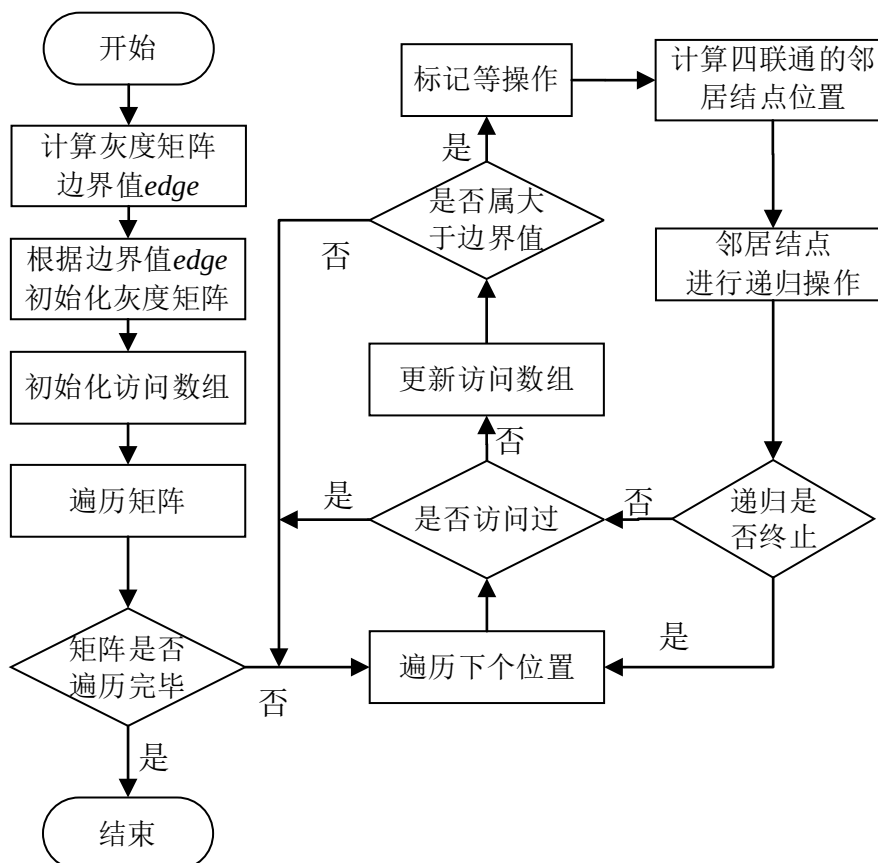


图 4.7 Flood Fill 算法的基本流程图

光伏板的红外图像结果经过灰度处理后得了一个灰度矩阵，为了方便，下面将使用灰度图中某一块含热斑区域的灰度子矩阵进行热斑检测，该灰度矩阵记为 A_{RC} ，其中 R 为灰度矩阵的行数， C 为灰度矩阵的列数。

由于热斑区域的像素点在列方向上可能达到几千个像素点或者更大，使用递归进行搜索可能会导致栈发生溢出，故系统采用基于栈数据结构实现递归方式的 Flood Fill 算法。

(1) 确定边界值

Flood Fill 算法需要确定边界区域的值，设热斑区域和正常区域的边界值 $edge$ 表示非热斑区域的最大值。该结果接近正常光伏板区域经过灰度变换后的值。实际数值大概在 95 左右。而光伏热斑的数值则大于 95，且越接近热斑中心，热斑的数值越大。本文通过统计灰度矩阵中灰度值的频率，通过相邻两点的变换率来确定 $edge$ 。假设灰度值按照灰度大小统计频率后得到长度为 255 的频率向量 F ，定义灰度值 i 在此灰度矩阵的频率变换率为 $freqRate_i$ ，则有定义公式如式 (4.3) 所示：

$$freqRate_i = \frac{F(i) - F(i-1)}{F(i)}. \quad (4.3)$$

经过实验测试得到如图 4.8 所示的结果。可知下图的边界值为 95。

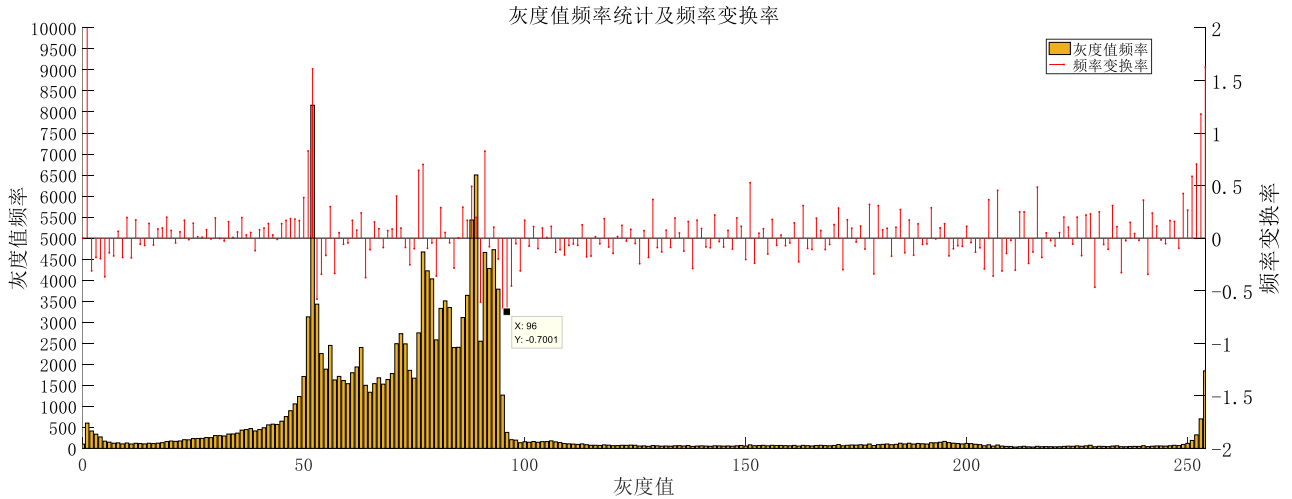


图 4.8 灰度矩阵的频率统计和变换率图

(2) 初始化矩阵

初始化过程主要将灰度矩阵中小于边界值 $edge$ 的数值初始化为边界值。并初始化一个行列等同于灰度矩阵的访问矩阵 V_{RC} ，设置 V_{RC} 的每个位置的值均为 0，表示没有遍历访问过，如式 (4.4)、(4.5) 所示：

$$A(i, j) = \begin{cases} edge & A(i, j) < edge \\ A(i, j) & A(i, j) \geq edge \end{cases} \quad \begin{matrix} 0 \leq i < R \\ 0 \leq j < C \end{matrix} \quad (4.4)$$

$$V(i, j) = 0 \quad \begin{matrix} 0 \leq i < R \\ 0 \leq j < C \end{matrix} \quad (4.5)$$

如图 4.9 所示，(a) 为某图片经过处理得到的原始灰度矩阵，(b) 为某图片经过初始化后的二维矩阵，其中红色区域即为实际的热斑区域。

95	95	95	95	95	95	95	95	92	95
95	95	95	95	95	95	95	95	95	95
93	95	97	100	103	102	99	95	95	95
95	97	102	110	111	108	99	95	95	95
95	97	101	107	109	104	98	95	95	95
95	98	99	100	102	103	97	95	92	95
95	95	95	95	95	95	95	95	95	95
95	94	95	95	95	95	95	95	95	95
95	95	95	95	94	93	93	93	95	95
95	95	95	95	95	95	95	95	95	95

(a) 灰度矩阵

(b) 处理后的矩阵

图 4.9 光伏板区域的灰度图矩阵

光斑检测过程中光斑区域的灰度值必定大于边界值，小于边界的数值通过初始化为边界值可以减少不必要的计算。

(3) 遍历矩阵

根据 Flood Fill 算法的步骤，从行到列遍历整个灰度矩阵，遍历到位置 (i, j) ，首先标记访问矩阵在 (i, j) 下的值为 1，表示灰度矩阵在此位置到达过，防止重复染色和重复遍历。如果灰度矩阵在 (i, j) 的值为边界值则继续遍历下一个元素。如果灰度矩阵在 (i, j) 的值大于灰度值，则开始进行染色操作。

(4) 确定热斑区域

如图 4.10 所示，热斑区域的确定就是 Flood Fill 的染色过程。

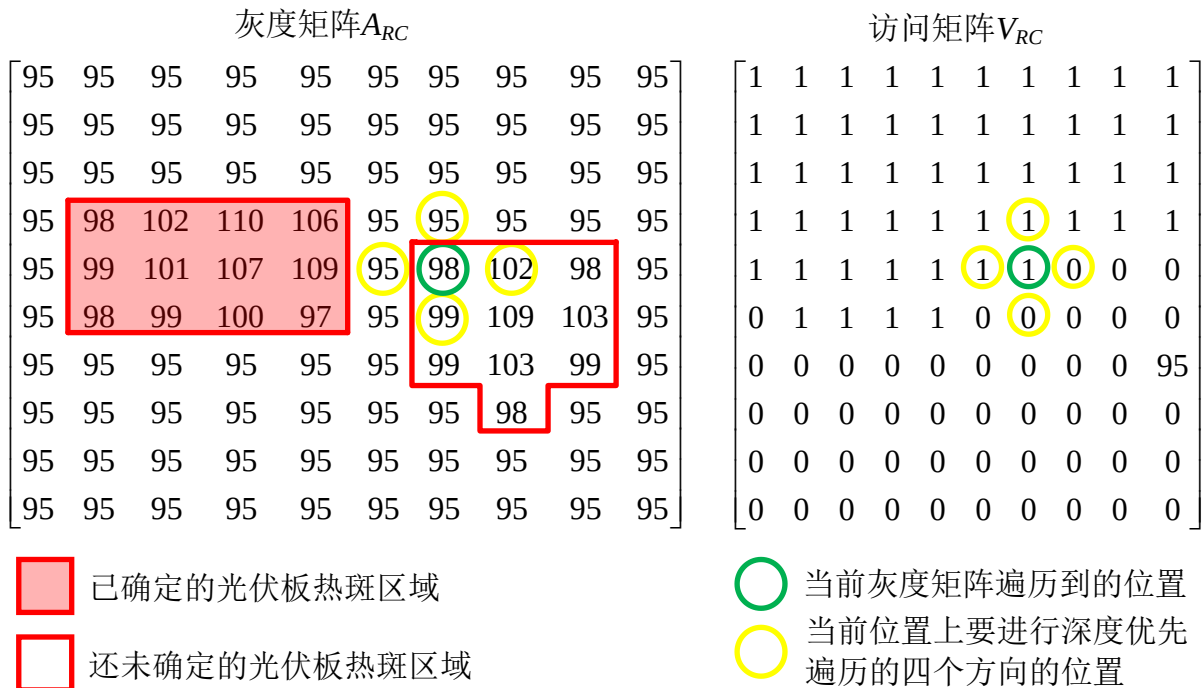


图 4.10 Flood Fill 算法填充热斑区域

使用深度优先搜索标记光斑区域或者使用广度优先搜索来标记光斑区域。以深度优先搜索为例，首先标记访问矩阵 V_{RC} 的 (i,j) 位置的元素为 1 表示访问过，再以当前点 (i,j) 为中心，分别向 (i,j) 的上下左右四个方向进行递归搜索，执行相同的操作。直到热斑区域的所有点均被遍历完毕，结束热斑区域的搜索，设置热斑个数为 1，此时访问数组在热斑区域的值均为 1，热斑区域的数值将不会被重复遍历。

热斑区域确定完成之后，继续执行步骤(2)的矩阵遍历。依次执行上述操作直到灰度矩阵的遍历结束，这样就记录了灰度矩阵中的热斑个数。

上述就是使用 Flood Fill 来进行热斑检测的过程，很明显此方法在判断热斑问题上不仅可以判断一片区域的光伏板上是否存在热斑，而且还可以统计该区域上热斑的个数。

4.3.3 Flood Fill 进行热斑检测的测试

本小节要对 Flood Fill 进行热斑检测的算法进行定性的测试。系统在不同环境下的光伏板下进行了热斑检测，并对比了文献[22]中基于 SVM 进行热斑识别的模型。如图 4.11、4.12 为两种算法在基于无干扰的经典热斑场景下的识别和在有虚警干扰情况下的热斑识别的效果图。

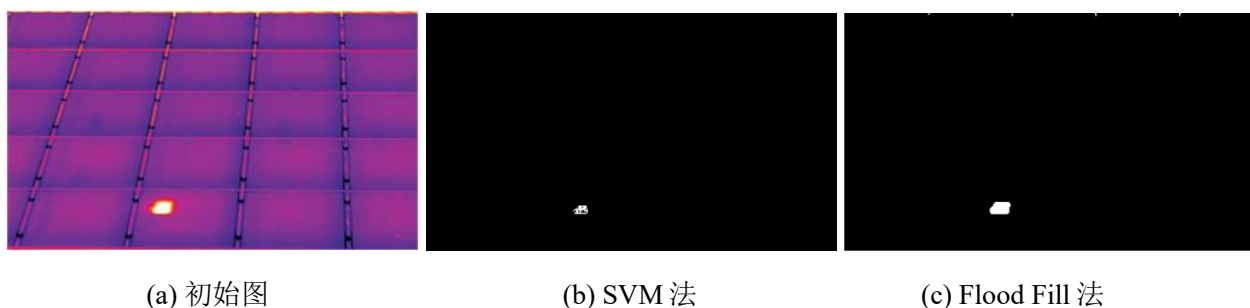


图 4.11 经典场景对比

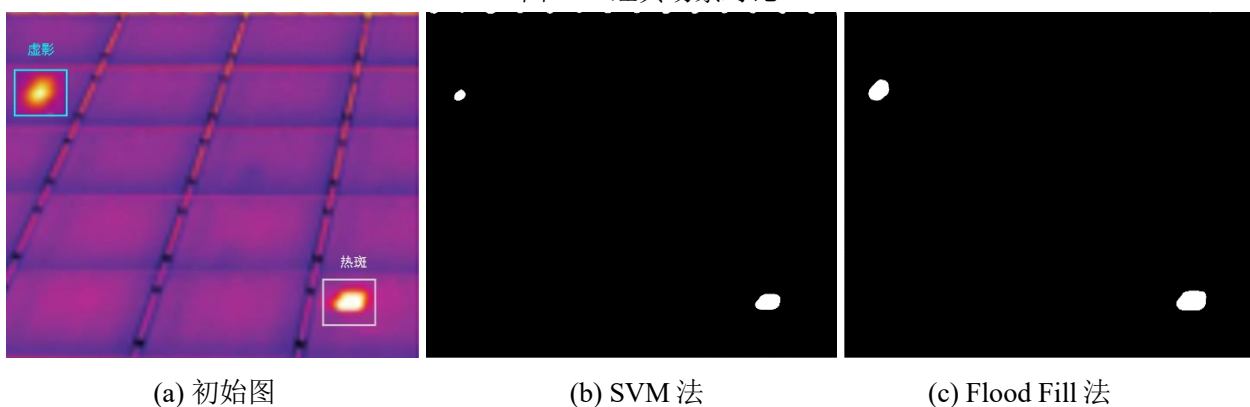


图 4.12 含虚警场合对比

上述测试，从定性角度分析了基于 SVM 优化识别和本文基于 Flood Fill 识别的结果，可以看出两种方法均不能有效识别虚警场合。

通过测试和观察误差数据的原始图像，发现 Flood Fill 和基于 SVM 优化的经典算法在解决热斑问题上存在以下不足，从而导致算法在下述不足场景中可能会产生误判：

- (1) 无法确定灰度化后的热斑区域是否是鸟屎等其他污渍；
- (2) 无法确定灰度化后的热斑是否是光源反射造成的无效光斑；
- (3) 图像处理过程中，可能存在特别干扰值，导致热斑区域被重复计算。

4.4 Flood Fill 优化的热斑检测方法实现

为了解决 Flood Fill 在虚警等干扰情况下识别热斑较差的问题，本文在基于 Flood Fill 的基础上引入了 XGBoost 进一步识别热斑形态。

4.4.1 XGBoost 算法

在传统机器学习算法中，单个学习器对最终结果的预测可能不太准确，XGBoost 使用了集成的思想，它由若干个相对较弱的分类器结合而成，并且 XGBoost 期望每一轮预测后的下一轮预测都会提升预测结果，使得最终预测结果无限逼近真实结果^[55]。

XGBoost 使用决策树作为学习器，首先形成一个基础的学习器，通过对学习器的判断，然后形成新的学习器来弥补前一个学习器的残差，如此反复地形成 T 个学习器，残差也在不断的迭代中变得越来越小，最终形成一个误差小精度高的学习器，再将 T 个学习器组合起来。提高最终分类器的精度是通过训练过程中一次又一次地降低误差得到的，所以 XGBoost 在训练集上有更好的表现。找到了弱学习器后，XGBoost 就要对这 T 个弱学习器进行组合，XGBoost 组合的方式是把这 T 个弱学习器组成一个加法模型^[56]。

XGBoost 作为一个集成算法，假设集成了 t 个弱分类器，每个分类器的决策树模型设为 $f_t(x)$ ，于是经过 t 个分类器作用后的样本预测结果如式 (4.6) 所示：

$$y_i^{(t)} = \sum_{k=1}^t f_k(x_i) = \sum_{k=1}^{t-1} f_k(x_i) + f_t(x_i) = y_i^{(t-1)} + f_t(x_i) \quad (4.6)$$

公式 (4.6) 表明第 t 轮的模型预测结果为，前 $t-1$ 轮预测的结果在集成了第 t 个弱分类器下的预测结果。其中 x_i 为样本 i 自身的特征值， $f_k(x_i)$ 称为第 k 个决策树的叶子结点的权重，即为对应样本 i 的预测值，XGBoost 的最终输出结果就是所有决策树中叶子结点的权重和^[56]。

XGBoost 在遍历决策树时通过加入正则项来进行剪枝，这使得模型更加简单和高效。因此 XGBoost 的目标函数由两部分构成：一是减少模型偏差的损失函数；二是抑制模型复杂度的正则项。XGBoost 算法的目标函数如式 (4.7) 所示：

$$L(\phi) = \sum_{i=1}^n l(y_i, y_i) + \sum_{k=1}^K \Omega(f_k) \quad (4.7)$$

式中，损失值为 $\sum_{i=1}^n l(y_i, y_i)$ ，正则项为 $\sum_{k=1}^K \Omega(f_k)$ ，损失值衡量真实标签 y_i 和预测值 y_i 之间的差异。因此，结合式 (4.7) 可以将目标函数改写为如式 (4.8)：

$$L^{(0)} = \sum_{i=1}^n l(y_i, y_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (4.8)$$

XGBoost 为了提升每轮的预测结果，会使得目标函数进行下降，接下来求让目标函数取得最小值的 f_t ，使用 Taylor 公式展开到平方项来对目标进行近似得到如式 (4.9) 所示：

$$L^{(0)} \cong \sum_{i=1}^n l(y_i, y_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)) + \Omega(f_t) \quad (4.9)$$

式中， g_i 是一阶导数， h_i 是二阶导数，如式 (4.10) 所示：

$$g_i = \partial_{y^{(t-1)}} l(y_i, y^{(t-1)}), \quad h_i = \partial_{y^{(t-1)}}^2 l(y_i, y^{(t-1)}) \quad (4.10)$$

在第 t 个模型下可以认为前 $t-1$ 棵树的模型产生的结果为固定值，故这些预测值和真实值可以当做常数被忽略，于是目标函数被化简为如式 (4.11) 所示：

$$L^{(0)} = \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad (4.11)$$

式中， $\Omega(f_t)$ 即正则化惩罚项，表示为式 (4.12) 所示， T 为决策树叶子单位数量， ω_j 是第 j 个叶子单位的预测值， γ 为惩罚系数。

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T \omega_j^2 \quad (4.12)$$

将 (4.12) 带入目标函数后即将推导得到如式 (4.13)：

$$\begin{aligned} L^{(0)} &= \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) = \sum_{i=1}^n [g_i \omega_{q(x_i)} + \frac{1}{2} h_i \omega_{q(x_i)}^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T \omega_j^2 \\ &= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) \omega_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) \omega_j^2] + \gamma T \end{aligned} \quad (4.13)$$

式中， I_j 为叶子节点 j 的所有样本序列的集合。令 $G_j = \sum_{i \in I_j} g_i$ ， $H_j = \sum_{i \in I_j} h_i$ ，则目标函数表示为如式 (4.14) 所示：

$$L^{(0)} = \sum_{j=1}^T [G_j \omega_j + \frac{1}{2} (H_j + \lambda) \omega_j^2] + \gamma T \quad (4.14)$$

在分类器和树模型确定的情况下，对上式求关于 ω_j 的偏导数，得到的式子为 0 时，可以求得 ω_j 的值如式 (4.15) 所示：

$$\omega_j^* = -\frac{G_j}{H_j + \lambda} \quad (4.15)$$

当偏导数为 0 成立时，可以计算得到对应的目标函数最小，对应的表达式如式 (4.16)：

$$L_{(0)}^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \quad (4.16)$$

在确定了目标函数后，就可确定分割左右子树后得到的损失熵的变化量，即分割前后的损失值变化量，如式 (4.17) 所示：

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (4.17)$$

XGBoost 搜索了全部可能的特征并对它们进行左右子树的分割，在遍历集合的过程中，算法选择了分割前后损失表达式变化量最大的特征来作为分割的节点。XGBoost 的优势体现在使用多个分类器无限逼近最真实的结果，这使得算法有较高的精度，同时由于算法支持对数据进行采样，以及算法遍历过程中存在剪枝操作，这使得 XGBoost 在进行分类时的运行速度较快^[56]。

4.4.2 数据处理

同一光伏板上的热斑，在相同红外成像下的灰度变换图具有如下表 4.2 所示的特征：

表 4.2 热斑特征

特征名称	特征值的解释和计算
像素点总数	热斑区域的像素点个数
相对灰度值和	热斑区域灰度值的和 / <i>edge</i>
相对最大灰度值	热斑区域的最大灰度值 / <i>edge</i>
相对最小灰度值	热斑区域的最小灰度值 / <i>edge</i>
热斑长	热斑区域所在行方向上的最大像素点个数
热斑宽	热斑区域所在列方向上的最大像素点个数

在 Flood Fill 进行深度优先遍历确定热斑区域的过程中，系统使用一个热斑类的集合将单位光伏板下的所有集合全部记录在里面。热斑类 HotSpot 是由若干个像素点的索引组成的列表。对于每个光伏板有唯一的热斑类集合 *Set*。*Set* 中可以记录多个热斑类集合，集合个数为热斑的个数。Flood Fill 的过程中，会获取每个热斑集合并记录在 *Set* 中。在 Flood Fill 进行深度优先遍历结束后，开始进行热斑特征的计算，如图 4.13 为一个热斑特征计算的例子。

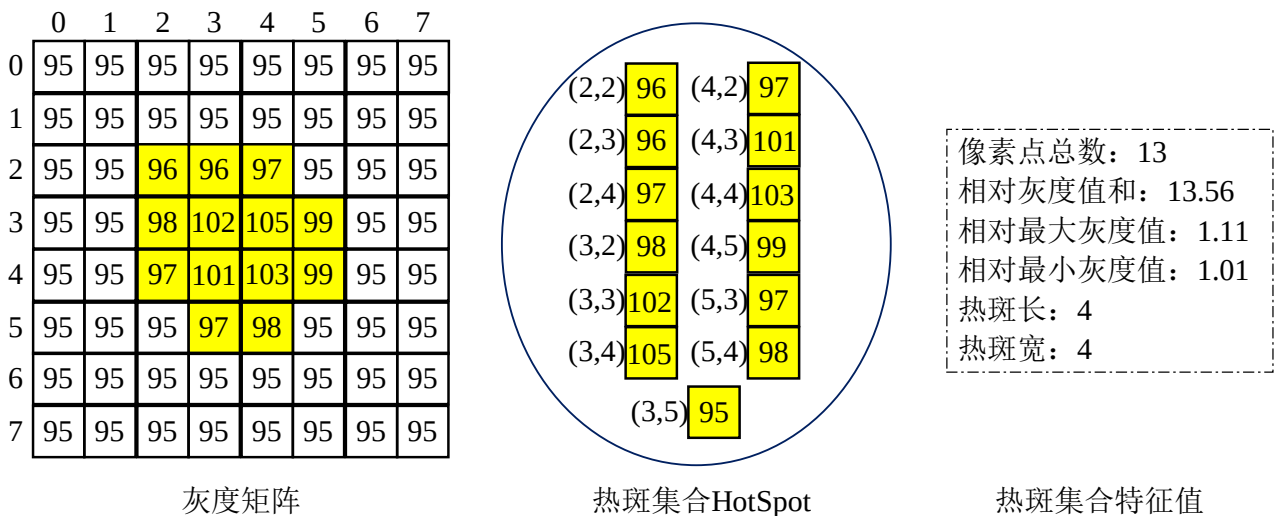


图 4.13 特征值获取

具体的热斑特征计算过程如图 4.14 所示。

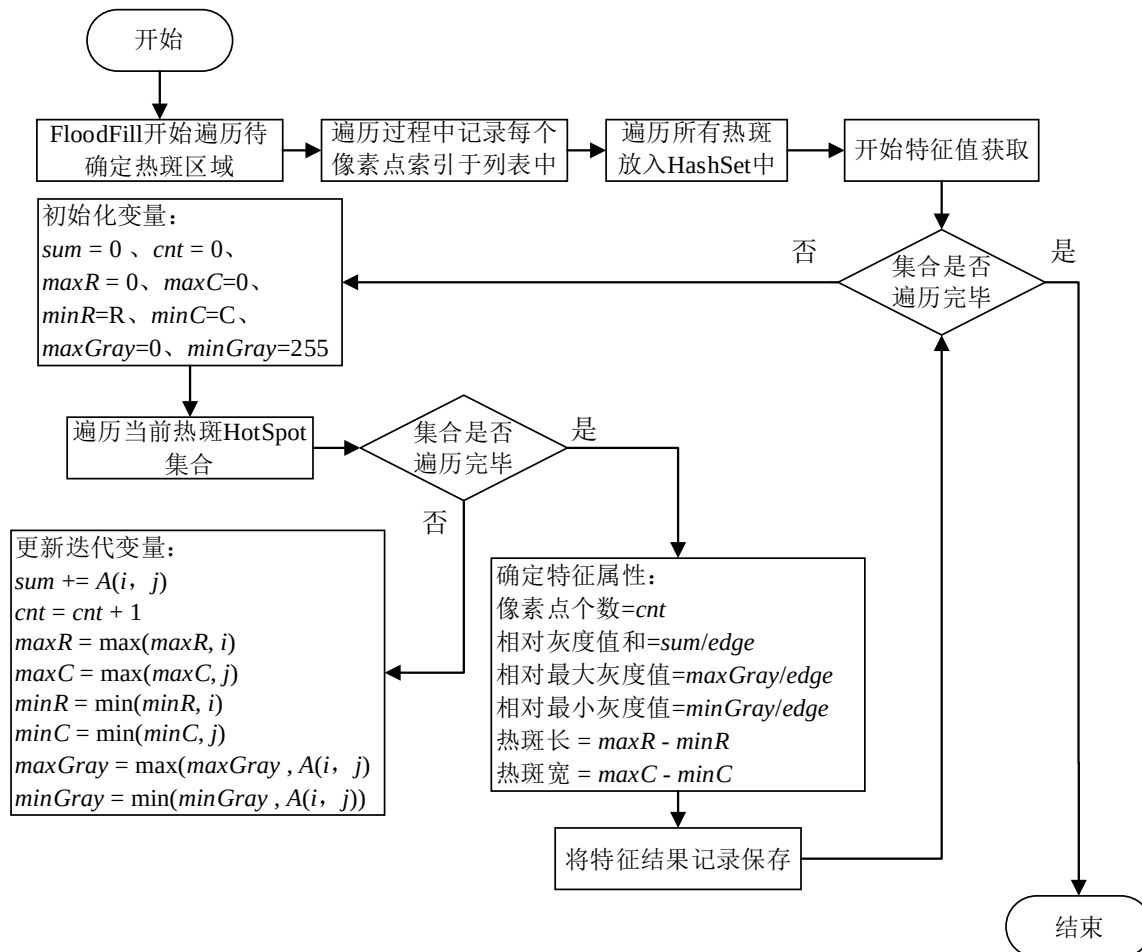


图 4.14 特征值获取流程图

至此，我们将最终的所有特征数据保存了下来，作为 XGBoost 的输入样本。

4.4.3 XGBoost 的参数选择和构建

本文使用 Python 的 Sklearn 库作为框架，经过了简单的数据处理后，算法使用了如表 4.3 所示的参数配置。

表 4.3 XGBoost 参数选择

XGBoost 参数名称	使用的参数值	XGBoost 参数名称	使用的参数值
<i>booster</i>	gbtree	<i>subsample</i>	0.7
<i>objective</i>	binary:logistic	<i>colsample_bytree</i>	0.5
<i>eval_metric</i>	auc	<i>min_child_weight</i>	3
<i>gamma</i>	0.1	<i>silent</i>	0
<i>max_depth</i>	8	<i>eta</i>	0.03
<i>alpha</i>	0	<i>nthread</i>	-1
<i>lambda</i>	0	<i>seed</i>	2022

4.5 模型的测试与评估

4.5.1 模型的评估指标

热斑识别属于二分类算法，评价二分类模型的指标很多，本文使用混淆矩阵来对模型进行评估。混淆矩阵由两个长度为 2 的向量组成，主要统计两类别的真实判别个数和误判个数，即统计以下四组记录的个数：正确判断的肯定记录（ TP ）、错误判断的肯定记录（ FN ）、正确判断的否定记录（ FP ）以及正确判断的否定记录（ TN ）。矩阵形式如表 4.4 所示。

表 4.4 混淆矩阵格式

混淆矩阵		真实值	
		正类（ P ）	负类（ N ）
预测值	假设正（ T ）	真正类（ TP ）	假正类（ FP ）
	假设负（ F ）	假负类（ FN ）	真负类（ TN ）

对于光伏热斑检测领域，还经常使用准确率、误检率和漏检率来描述检测的可靠性。

(1) 精确率：该值越大代表算法的精确性越高，计算方式为：正确预测到的光伏热斑基数除以实际光伏热斑基数。根据混淆矩阵有中的数值有如式（4.18）：

$$precision = \frac{TP}{FN + TP} \quad (4.18)$$

(2) 误检率：指被错分到正样本类型中真实的负样本，占全部负样本基数的比例，如式（4.19）所示该参数可以看出算法的稳定性，在本文中计算方式为：算法把非光伏热斑判断成光伏热斑的基数除以真实非光伏热斑基数得到的百分比。该值越低，表示稳定性越好。

$$FPR = \frac{FP}{FP + TN} \quad (4.19)$$

(3) 漏检率：同样是检验事件中的一个重要指标，是指检验事件中未发现的目标个数占据了当前检验总个数的比例，在本文中指的是检测失败的真实热斑基数与实际热斑基数的百分比，该值越小，说明稳定性越好，不容易发生误判，根据混淆矩阵有中的数值得到如式（4.20）：

$$\begin{cases} miss_rate = \frac{FN}{FN + TP} \\ miss_rate + precision = 1 \end{cases} \quad (4.20)$$

4.5.2 模型测试

为了对模型进行评估，本实验获取了 400 张光伏电站的红外图像，并选择了 150 张作为训练集，50 张作为验证集，剩余 200 张作为测试集。使用准确率、误检率和漏检率以及混淆矩阵作为系统的评估指标，并和文献[22]基于 SVM 的经典识别法以及没经过 XGBoost 优化的 Flood Fill 识别法进行了对比，得到了下面所示的相关检测结果。

如图 4.15 所示，为 XGBoost 优化后的效果对比图，可以看出优化后的算法可以识别出热斑形态，并过滤掉虚警这类非热斑带来的干扰。

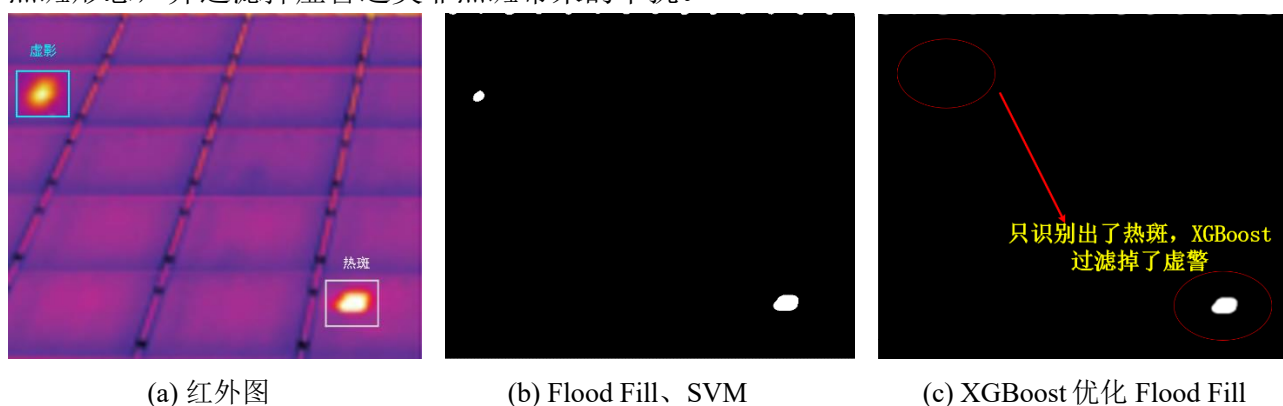


图 4.15 虚警场合对比

优化算法在 200 张测试集中对每个可疑热斑样本的检测，其中样本的总数为热斑个数与非热斑可疑目标个数的和。表 4.5 为测试结果得到的混淆矩阵。

表 4.5 实验结果的混淆矩阵

混淆矩阵		真实值	
		是热斑	不是热斑
预测值	是热斑	144	10
	不是热斑	12	110

进一步地在测试结果的基础上，根据上一小节的公式计算了精确率、漏检率、误检率，并和文献[22]中基于 SVM 优化以及没有经过 XGBoost 的 Flood Fill 检测两种方进行了对比，得到了如表 4.6 所示的检测结果。

表 4.6 热斑检测参数对比

指标 检测方法	精确率	漏检率	误检率
SVM 检测法	89.54	10.46	16.34
纯 Flood Fill 法	88.83	11.17	15.91
XGBoost 优化 Flood Fill	92.31	7.69	8.33

有上表可以看出，经过 XGBoost 优化后的 Flood Fill 在热斑检测的精确率上有所上升，在漏检率和误检率上下降明显。本方法和传统的基于 SVM 进行热斑检测的方法相比，其漏检率和误检率均有明显的降低，可以更好地区分虚警等干扰光斑。

对漏检样例排查发现，极少数位于光伏板边缘的热斑在识别时会发生形态上的干扰，但该类情形在热斑总数的占比极少。对误检样例进行排查发现，极少数光伏板上的热源物体（如昆虫等）有时会被误识别为光伏热斑。实际使用中，算法在虚警的区分上有较高准确率，

在大多数场合下均能对热斑进行准确识别。

4.6 本章小节

本章首先对光伏热斑的形成原理进行了简单的讲解，根据相关原理阐述了光伏热斑识别中的难点和痛点。首先介绍了基于 Flood Fill 实现热斑检测的算法流程。根据传统识别无法有效区分虚警的问题，本文提出了一种基于 XGBoost 优化 Flood Fill 识别热斑属性的算法，该算法可以有效的区分干扰物体的成像，最后通过测试实验得到该方法的精确率在 92.31%，误检率为 8.33%，漏检率为 7.69%，在误检率、漏检率上低于传统的识别方法，能够降低光伏电站后台维护的成本和人力，大大提高光伏系统的稳定性，能够让员工及时发现热斑，并针对性地进行维护。

第5章 总结与展望

5.1 总结

光伏能源是我国大力发展的一种清洁能源，如何高效地为光伏系统进行运维和清洁一直是工业界的一大热门话题。本文调研了工业界光伏电站的运维、控制中的需求痛点以及技术不足，设计了一种基于物联网技术和机器学习的光伏电站运维系统，系统具备智能清洁、功率预测、热斑检测等功能。为了解决上述痛点和不足，本文提出了一种基于 KNN 来预测短期光伏功率的方案，以及提出了一种基于 XGBoost 优化 Flood Fill 的热斑检测方法。本文的主要工作如下：

(1) 对光伏电站远程运维系统中的硬件控制部分的各个功能模块进行了设计介绍。硬件部分通过以 ARM Cortex-M3 为 MCU 的控制板和网关板来实现整体机器人的清洁和远程运维的命令通信以及监控点的上报。其中使用霍尔传感器来标识机器人的起点和终点，通过直流有刷电机来控制机器人的清扫。控制板和通信板通过 485 总线进行通信。

(2) 后台系统通过设备云和边缘终端设备进行通信。后台采用微服务架构，设计了远程更新服务、分布式缓存服务、天气分析服务、数据分析服务、业务服务、监控服务等重要功能服务和基础服务。远程更新使用并行 HTTP 请求和多线程任务编排的设计来优化了整体响应时间。分布式缓存采用三级缓存架构大大加快了后端接口调用的速度，并且为热斑检测服务和功率预测服务提供了数据存储的空间。

(3) 本文主要研究了温度、光辐射、倾斜角度对光伏输出的具体影响，为后台分析算法提供了理论依据。同时详细地介绍了 KNN 算法用于预测光伏输出的具体实现。实验测试中，将 KNN 和经典的 BP 神经网络进行了对比，得出 MAPE 为 7.83% 和 RMSE 为 0.5053 均优于传统的 BP 神经网络方法，并且 KNN 算法执行速度较快。

(4) 光伏板的热斑会造成系统耗能，并且会造成安全隐患。本文设计并实现了一种高效地热斑识别方法。首先基于 Flood Fill 识别出热斑的形态，在使用 XGBoost 进行热斑形态的学习，算法在预测地准确性上远高于其他方法，同时可以区分非热斑造成的干扰。最终经过实验测得精确率为 92.31%，漏检率为 7.69%，误检率为 8.33%。

5.2 展望

光伏电站的远程运维系统运用了物联网技术、机器学习、分布式、云原生等前沿技术，提供了一个功能强大多样的运维平台。但是由于本人水平有限，并且系统运行的时间还不够长，本系统还有许多不足和有待进一步改进的地方：

(1) 清洁机器人的清洁效率还不够高，后续可以考虑将单排机器人改成便携式清洁机器人，并对机器人进行路径规划，且将清洁区域设定为目标点，因此只需对目标点进行清洁。可以借鉴一些图论领域的最短路径算法如 Dijkstra 来对机器人的路径进行规划。这样清洁效率将会得到更大的提升。

(2) 对于光伏功率预测方向，KNN 算法在数据区间边缘两端的预测效果较差，并且在数据集较小的情况下整体的预测误差相比数据集大的情况下要高。后续可以考虑使用组合多种时序分析或者分类预测的算法对当前算法进行改进。

(3) 在热斑识别方面，对于相连热斑的检测准确率还不够高，有时受环境和拍摄影响，相邻两块热斑会被识别成一块热斑，对此可以使用相关机器学习算法学习热斑的面积，将连在一起的成像拆分开来，从而解决相邻热斑的识别问题。

参考文献

- [1] 《中国可再生能源发展报告》正式发布[J]. 建设机械技术与管理, 2021,34(04):7.
- [2] 曹林, 王静, 王宝友, 等. 太阳光伏能源技术及其标准化[J]. 信息技术与标准化, 2011(Z1):30-34.
- [3] 李卫民. 浅论中国发展分布式光伏能源的经济效益[J]. 中小企业管理与科技(上旬刊), 2019(05):71-72.
- [4] 韩熠. 太阳能光伏板清洁机器人路径规划研究与应用[D]. 兰州: 兰州理工大学, 2017.
- [5] 董伯先. 浅析光伏电站清洁机器人的应用现状[J]. 太阳能, 2020(03):46-50.
- [6] 海涛, 梁挺兴, 黄曲达, 等. 一种高效的光伏监控方案及发电量预测[J]. 计算机测量与控制, 2015,23(08):2637-2639.
- [7] 臧建彬, 王亚伟, 王晓东. 灰尘沉积影响光伏发电的理论和试验研究[J]. 太阳能学报, 2014,35(04):624-629.
- [8] 宋爱国, 许春山, 徐宝国, 等. 复杂环境下远程巡检机器人关键技术及应用[Z]. 2015.
- [9] Supe H, Avtar R, Singh D, et al. Google Earth Engine for the Detection of Soiling on Photovoltaic Solar Panels in Arid Environments[J]. Remote Sensing, 2020,12(9):1466.
- [10] 刘旭, 王硕禾, 黄晓波, 等. 多模态LoRa通信技术在光伏机器人的应用[J]. 石家庄铁道大学学报(自然科学版), 2020,33(03):58-64.
- [11] 薛家祥, 龚普, 吴坚. 基于无线通信和Android手机的光伏监控系统的实现[J]. 自动化与仪表, 2016,31(10):28-31.
- [12] 庄清霖. 基于ZigBee协议的分布式光伏电站运行数据实时监控系统[J]. 农村电气化, 2021(07):57-59.
- [13] Elizalde J, Cruces C, Sandoval M S, et al. Self-Powered Photovoltaic Bluetooth® Low Energy Temperature Sensor Node[J]. IEEE Access, 2021,9:111305-111314.
- [14] 张俊蔚. 基于物理模型的光伏电站输出功率预测[J]. 甘肃水利水电技术, 2015,51(01):46-49.
- [15] 杨磊, 高德东, 孟广双, 等. 基于BP神经网络的故障光伏组件输出功率预测[J]. 青海大学学报, 2021,39(02):91-97.
- [16] 王倩, 张智晟, 王帅, 等. 基于PSO-RNN的光伏发电功率预测研究[J]. 青岛大学学报(工程技术版), 2014,29(04):40-43.
- [17] Wu Z, Hu Y, Wen J X, et al. A Review for Solar Panel Fire Accident Prevention in Large-Scale PV Applications[J]. IEEE Access, 2020,8:132466-132480.
- [18] Kim K A, Krein P T. Reexamination of Photovoltaic Hot Spotting to Show Inadequacy of the Bypass Diode[J]. IEEE Journal of Photovoltaics, 2015,5(5):1435-1441.
- [19] Dhimish M, Mather P, Holmes V. Novel Photovoltaic Hot-Spotting Fault Detection Algorithm[J]. IEEE

- Transactions on Device and Materials Reliability, 2019,19(2):378-386.
- [20] Hong F, Song J, Meng H, et al. A novel framework on intelligent detection for module defects of PV plant combining the visible and infrared images[J]. Solar Energy, 2022,236:406-416.
- [21] Niazi K A K, Akhtar W, Khan H A, et al. Hotspot diagnosis for solar photovoltaic modules using a Naive Bayes classifier[J]. Solar Energy, 2019,190:34-43.
- [22] 陈文勤. 一种基于红外图像识别的光伏组件热斑检测系统的研究与实现[D]. 南京: 南京邮电大学, 2020.
- [23] Al-Debagy O, Martinek P. Extracting Microservices' Candidates from Monolithic Applications: Interface Analysis and Evaluation Metrics Approach, 2020[C]. IEEE, 2020.
- [24] Popovic R S. Hall devices for magnetic sensor microsystems, 1997[C]. IEEE, 1997.
- [25] Li M, Amin A, Torrea R, et al. Processor based 20Mhz 4×4 Cat-5 LTE MIMO receiver with advanced detectors, 2013[C]. IEEE, 2013.
- [26] Zhamanov A, Sakhiyeva Z, Suliyev R, et al. IoT smart campus review and implementation of IoT applications into education process of university: 2017 13th International Conference on Electronics, Computer and Computation (ICECCO), 2017[C]. IEEE, 2017.
- [27] Yassein M B, Shatnawi M Q, Aljwarneh S, et al. Internet of Things: Survey and open issues of MQTT protocol, 2017[C]. IEEE, 2017.
- [28] Haselbock S, Weinreich R, Buchgeher G. An Expert Interview Study on Areas of Microservice Design, 2018[C]. IEEE, 2018.
- [29] Zuo X, Su Y, Wang Q, et al. An API gateway design strategy optimized for persistence and coupling[J]. Advances in Engineering Software, 2020,148:102878.
- [30] Lubis R, Sagala A. Multi-thread performance on a single thread in-memory database, 2015[C]. IEEE, 2015.
- [31] Ji Z, Ganchev I, O'Droma M, et al. A Distributed Redis Framework for Use in the UCWW, 2014[C]. IEEE, 2014.
- [32] Chowdhury S A, Sapra V, Hindle A. Client-Side Energy Efficiency of HTTP/2 for Web and Mobile App Developers, 2016[C]. IEEE, 2016.
- [33] Ding Z, Li R, Wang X. Research and Implementation of Jetspeed2-Based SSO, 2009[C]. IEEE, 2009.
- [34] Kieseberg P, Schrittwieser S, Fruhwirt P, et al. Analysis of the Internals of MySQL/InnoDB B+ Tree Index Navigation from a Forensic Perspective, 2019[C]. IEEE, 2019.
- [35] 邱纯. 任意辐照度与温度条件下光伏系统输出特性建模[D]. 湖北: 华中科技大学, 2011.
- [36] Mukisa N, Zamora R. Optimal tilt angle for solar photovoltaic modules on pitched rooftops: A case of low latitude equatorial region[J]. Sustainable Energy Technologies and Assessments, 2022,50:101821.
- [37] Liu L, Liu D, Sun Q, et al. Forecasting Power Output of Photovoltaic System Using A BP Network Method[J]. Energy Procedia, 2017,142:780-786.

- [38] Yunbo Z, Zhiguo L, Shengzhu L, et al. Research on BP neural network based grid-connected photovoltaic inverter, 2016[C]. IEEE, 2016.
- [39] Chen H, Chang X. Photovoltaic power prediction of LSTM model based on Pearson feature selection[J]. Energy Reports, 2021,7:1047-1054.
- [40] Park N, Ahn H K. Multi-Layer RNN-based Short-term Photovoltaic Power Forecasting using IoT Dataset, 2019[C]. AEIT, 2019.
- [41] Sansa I, Boussaada Z, Mazigh M, et al. Solar radiation prediction for a winter day using ARMA model, 2020[C]. IEEE, 2020.
- [42] Eskandari A, Milimonfared J, Aghaei M. Optimization of SVM classifier using Grid Search Method for Line-Line Fault Detection of Photovoltaic Systems, 2020[C]. IEEE, 2020.
- [43] Rastin N, Taheri M, Jahromi M Z. A stacking weighted k-Nearest neighbour with thresholding[J]. Information Sciences, 2021,571:605-622.
- [44] Nasibov E, Kandemir-Cavas C. Efficiency analysis of KNN and minimum distance-based classifiers in enzyme family prediction[J]. Computational Biology and Chemistry, 2009,33(6):461-464.
- [45] Roumpakias E, Zogou O, Stamatelos A. Correlation of actual efficiency of photovoltaic panels with air mass[J]. Renewable Energy, 2015,74:70-77.
- [46] Pena J C, Nápoles G, Salgueiro Y. Normalization method for quantitative and qualitative attributes in multiple attribute decision-making problems[J]. Expert Systems with Applications, 2022,198:116821.
- [47] Li G, Wang W, Zhang W, et al. Grid search based multi-population particle swarm optimization algorithm for multimodal multi-objective optimization[J]. Swarm and Evolutionary Computation, 2021,62:100843.
- [48] Chandola D, Gupta H, Tikkiwal V A, et al. Multi-step ahead forecasting of global solar radiation for arid zones using deep learning[J]. Procedia Computer Science, 2020,167:626-635.
- [49] Dhimish M. 70% Decrease of Hot-Spotted Photovoltaic Modules Output Power Loss Using Novel MPPT Algorithm[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2019,66(12):2027-2031.
- [50] Chen J, Li Y, Ling Q. Hot-Spot Detection for Thermographic Images of Solar Panels, 2020[C]. IEEE, 2020.
- [51] Ali H, Khan H A. Evaluation of Low-Voltage Loss Under Partial Shading Conditions in Solar Photovoltaic Systems, 2018[C]. IEEE, 2018.
- [52] Gutiérrez Galeano A, Bressan M, Jiménez Vargas F, et al. Shading Ratio Impact on Photovoltaic Modules and Correlation with Shading Patterns[J]. Energies, 2018,11(4):852.
- [53] Belhadj C A, Banat I H, Deriche M. A detailed analysis of photovoltaic panel hot spot phenomena based on the bishop model: 2017 14th International Multi-Conference on Systems, Signals & Devices (SSD), 2017[C]. IEEE, 2017.
- [54] Zhou G, Zhou J, Li Y. Filling depressions based on sub-watersheds in raster digital elevation models: 2017

- IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017[C]. IEEE, 2017.
- [55] Jidong L, Ran Z. Dynamic Weighting Multi Factor Stock Selection Strategy Based on XGboost Machine Learning Algorithm, 2018[C]. IEEE, 2018.
- [56] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System, 2016[C]. ACM, 2016.

致谢

光阴似箭，日月如梭，三年已过，纵观即将画上句号的研究生生涯，我有过懵懂好奇的时候，也有被挫折打击自信心的时候，也不缺少解决科研和工程问题时欣喜若狂的时候。母校杭州电子科技大学有着浓厚的学术氛围，电子信息学院的科研能力和工程能力不断地吸引我，让我在三年地求学路上学到了很多，这必然是我人生中的一段宝贵经历和财富，由衷地感谢学校给我们提供的机会和优质的教育，感谢每一个人。

首先，感谢我的导师陈科明副教授，老师扎实的学术功底和深厚的工程能力让我学到了很多专业上的知识和技巧。陈老师高尚的品德和实事求是地态度深深地影响了我，让我励志做一个踏实上进的人。本次写作，在选题、构思、撰写上，陈老师都给了我悉心的指导和热情的帮助。在此谨向我最敬爱的陈老师表达我最诚挚的敬意和感谢。

感谢我的同门刘剑、吴文楷、邢豪蔚、张百军、蔡坤、黄盼盼、胡晓双以及我的师弟师妹们，他们是我工程实践上的战友，在各个方面上给我我很大的技术帮助和建议，很荣幸拥有这么一帮优秀的同学。

感谢我的室友闻勇强、金千航、祝丰豪，我们是本科的同学，同时也是研究生室友，深厚的友谊让我们能够一起在科研和未来的工程道路上走的更远。感谢他们在不同专业方向上的帮助和指导。

感谢我的父母及妹妹，感谢你们对我学习上的支持和生活上的帮助，你们是我生活上的导师，是在外漂泊的我的温暖的港湾，让我顺利的走完了研究生生涯。

感谢论文评阅和答辩委员会的老师，感谢你们的耐心，我一定会悉心听取你们宝贵的意见并积极改正。

最后，衷心地祝愿所有人，万事如意，身体健康，心想事成，让我们一起迎接更美好的明天！

附录

作者在读研期间发表的学术论文及参加的科研项目

参加的科研项目：

- [1] 光伏电站远程运维与智能分析系统开发一期. 横向课题. 2019.12-2021.1. 项目编号: KYH043121088M
- [2] 光伏电站远程运维与智能分析系统开发二期. 横向课题. 2021.6-2022.12. 项目编号: KYH043120060M

发表的专利：

- [1]陈科明，徐志杰.一种基于K最近邻分类的光伏功率预测方法[P]. 浙江省, 202210214610.6; 2022-03-07