

方案：基于机器学习/深度学习的配网终端设备健康度评估方案

1. 目标

利用FTU设备定期上报的遥信(200项)和遥测(10+项)数据,构建一个基于机器学习/深度学习的健康度评估模型。该模型输出一个量化的健康度分数(例如0-100分),分数越低表示设备健康状况越差,风险越高,旨在实现:

- **早期预警**: 识别潜在的设备故障风险。
- **精准运维**: 指导运维人员优先处理状态最差的设备。
- **主动维保**: 替代传统的周期性维保,实现状态修,降本增效。

2. 数据准备与预处理

- **数据源**: 云平台存储的FTU遥信、遥测时间序列数据。
- **数据整合**: 按设备ID和时间戳关联遥信和遥测数据。
- **时间窗口**: 由于数据是定期上报(如15分钟),需要定义一个或多个时间窗口(例如:过去1小时、6小时、24小时、7天)来聚合特征,捕捉短期波动和长期趋势。
- **数据清洗**:
 - **缺失值处理**:
 - 遥信: 可使用前一个状态填充(forward fill)或标记为特殊类别。
 - 遥测: 可使用均值/中位数填充、插值法(线性、样条)或模型预测填充。对于序列模型,可能使用Masking机制。
 - **异常值处理**: 对遥测数据进行异常值检测和处理(如盖帽法、IQR法)。
 - **时间戳对齐**: 确保数据按时间顺序排列,处理可能的延迟或乱序数据。
- **数据编码与标准化**:
 - 遥信: 将状态量(如“合/分”、“正常/异常”)转换为数值(如0/1)或进行独热编码(One-Hot Encoding),特别是对于多状态指标。
 - 遥测: 进行标准化(Standardization,均值为0,方差为1)或归一化(Normalization,缩放到0-1),消除量纲影响,便于模型训练。

3. 特征工程(重点解决疑问点2 & 3)

这是决定模型效果的关键步骤。我们需要从原始数据中提取能有效反映设备健康状况的特征。

- **(疑问点2) 遥信指标特征抽取**: 200多个遥信指标数量较多,直接输入可能导致维度灾难和噪声干扰。**强烈建议进行特征抽取和选择,而不是直接使用原始状态。**
 - **基于时间窗口的衍生特征**:
 - **状态变化频率**: 特定关键遥信点(如电源、通信、关键告警)在窗口内变化的次数。
 - **异常状态持续时间/占比**: 各种告警(电池、通信、硬件故障等)状态在窗口内的累计持续时间或时间占比。
 - **特定事件计数**: 如PT断线、操作失败、装置重启等事件在窗口内的发生次数。
 - **状态稳定性指标**: 计算某些关键状态序列的熵、或者连续保持某个状态(尤其是异常状态)的最大时长。
 - **时间距离特征**: 距离上次特定告警(如交流失电、硬件故障)发生的时间。
 - **特征选择**:
 - **领域知识筛选**: 首先,与电网运维专家合作,根据经验判断哪些遥信指标与FTU的常见故障模式(如电源问题、通信问题、主板/模块故障、测量异常等)直接相关,优先关注这些指标及其衍生特征。例如,电源模块状态、电池相关告警、通信异常、板卡故障等看起来就是高价值指标。
 - **统计方法筛选**: 计算特征的方差(低方差特征可能信息量少)、与目标(如果暂时有粗略标签,如“近期是否发生故障”)的相关性。
 - **模型筛选**: 使用如随机森林、XGBoost等模型输出的特征重要性评分进行筛选。或者使用L1正则化(Lasso)让模型自动选择特征。
 - **结论(Q2)**: 不建议直接用200+原始遥信状态。应先基于领域知识初步筛选,然后重点对筛选后的指标及其他可能有用的指标,计算时间窗口内的衍生统计特征,最后再用统计或模型方法进一步精炼特征集。
- **(疑问点3) 遥信与遥测的联合表现力**: 是的,遥信和遥测指标在设备健康度不佳时,具有很强的联合表现力。很多故障场景是两类数据异常的组合。
 - **联合场景示例**:
 - **电源问题**: "交流失电信号"(遥信)发生 + "电池电压低"(遥信)告警 + 遥测"电池电压"持续下降(遥测特征:趋势为负)。
 - **PT/CT问题**: "PT断线"(遥信) + 对应相的遥测"计量电压U"突变为0或异常值(遥测特征:突变检测)。
 - **通信模块故障**: "本机goose通信异常"(遥信) + 相关"遥测"数据(如功率、电流)全部变为0或无效值(遥测特征:值恒定或为无效)。
 - **过负荷/内部短路风险**: 遥测"计量IA/IB/IC"持续偏高(遥测特征:均值/最大值超阈值) + "装置硬件故障"(遥信)。
 - **特征工程体现**:
 - **直接组合**: 模型(尤其是深度学习和树模型)可以自动学习这种交互。
 - **显式创建交互特征(可选)**: 例如,创建一个特征表示"交流失电且电池电压低于阈值"。但这会增加特征数量,通常让模型自己学更好。
 - **结论(Q3)**: 遥信和遥测的结合是预测健康度的关键。模型设计需要能有效融合两类信息。

4. 模型设计与选择(重点解决疑问点1)

- **(疑问点1) 输入层设计**:
 - **方案A: 特征拼接 (Concatenation - 推荐首选)**:
 1. 对预处理后的遥信数据计算衍生特征 (Feature Set YX)。
 2. 对预处理后的遥测数据计算衍生特征 (Feature Set YC)。
 3. 将 YX 和 YC **拼接**成一个长的特征向量。
 4. 将此拼接后的向量作为模型的统一输入层。
 - **适用模型**:
 - **树模型**: XGBoost, LightGBM, Random Forest (对表格类特征数据效果通常很好,能自动处理特征交互,对特征缩放不敏感)。
 - **前馈神经网络 (MLP/FNN)**: 可以学习复杂的非线性关系。
 - **优点**: 结构相对简单,实现直接,大多数成熟的ML/DL框架都支持。模型可以在内部直接学习遥信和遥测特征之间的交互关系。
 - **缺点**: 如果时间序列的动态非常重要(而不仅仅是窗口统计特征),此方法可能丢失部分时序信息。
 - **方案B: 双通道输入与融合 (Dual-Channel Input & Fusion)**:
 1. 设计一个子网络(如MLP, 1D CNN, 或 LSTM/GRU)处理遥信衍生特征序列(或原始序列)。
 2. 设计另一个子网络处理遥测衍生特征序列(或原始序列)。
 3. 将两个子网络的输出(例如,最后隐藏层的表示)进行融合(如拼接、加权平均、注意力机制)。
 4. 融合后的表示再通过一个或多个全连接层进行最终预测。
 - **适用模型**: 定制的深度学习网络。
 - **优点**: 可以为不同类型的数据(状态序列 vs. 数值序列)设计最优的子网络结构,可能更好地捕捉各自的模式。
 - **缺点**: 模型结构更复杂,训练和调优难度更大。融合策略的设计是关键。
 - **结论(Q1)**: **建议首先尝试“方案A: 特征拼接”**,配合XGBoost/LightGBM或MLP。这通常是效果和复杂度的良好平衡点。如果发现时序动态性非常关键,且窗口统计特征不足以捕捉,再考虑更复杂的“方案B: 双通道融合”模型,可能结合LSTM/GRU来处理时间序列信息。
- **模型选择建议**:

- **基线模型**：从 XGBoost 或 LightGBM 开始。它们在表格数据上的表现通常非常出色，训练快，自带特征重要性评估，且对数据缩放不敏感。
- **深度学习模型**：
 - **MLP**：作为基于拼接特征的深度学习选项。
 - **LSTM/GRU（如果时序性重要）**：可以考虑用于方案B，或者直接将拼接后的特征序列输入LSTM（如果特征本身是按时间步组织的）。
 - **Autoencoder（用于无监督/异常检测）**：见下一节。

5. 样本不平衡处理 (重点解决疑问点4)

电网设备故障是小概率事件，健康样本远多于不健康/故障样本，这是典型的不均衡学习问题。

- **数据层面策略**：
 - **过采样 (Over-sampling) 少数类**：
 - **SMOTE (Synthetic Minority Over-sampling Technique)**: 生成与少数类样本相似的合成数据。注意可能引入噪声或导致过拟合，需要谨慎使用并配合后续策略。变种如ADASYN, Borderline-SMOTE等。
 - **随机过采样**：简单复制少数类样本。容易过拟合。
 - **欠采样 (Under-sampling) 多数类**：
 - **随机欠采样**：随机丢弃多数类样本。可能丢失多数类的重要信息。
 - **Tomek Links / Edited Nearest Neighbors (ENN)**: 删除多数类中与少数类靠近或分类不清的样本。
 - **NearMiss**: 保留与少数类样本最近的多数类样本。
 - **组合采样**：结合过采样和欠采样，如SMOTE + Tomek links。
- **算法层面策略**：
 - **代价敏感学习 (Cost-Sensitive Learning)**:
 - 在模型训练时，为少数类样本分配更高的错误分类代价（权重）。许多模型（如XGBoost, LightGBM, Scikit-learn中的SVM、逻辑回归等）支持设置 `class_weight` 参数为 'balanced' 或手动指定权重。**这是常用且推荐的方法。**
 - **集成方法**：
 - **Balanced Random Forest**: 在构建每棵树时，对抽取的样本进行平衡（过采样或欠采样）。
 - **EasyEnsemble / BalanceCascade**: 通过多次欠采样多数类，训练多个基学习器并集成。
- **评估指标**：
 - **放弃Accuracy！** 它在不均衡数据上具有误导性。
 - **使用**：
 - **Precision (精确率)**
 - **Recall (召回率/敏感度)** - 对少数类尤其重要！
 - **F1-Score (Precision和Recall的调和平均)**
 - **AUC-ROC (ROC曲线下面积)**
 - **AUC-PR (Precision-Recall曲线下面积)** - 对于严重不均衡数据通常比AUC-ROC更具信息量。
- **改变问题视角**：
 - **异常检测 (Anomaly Detection)**:
 - **思路**：假设大部分数据是“健康”的，训练模型学习“健康”模式。偏离此模式的设备即为“异常”或“不健康”。
 - **模型**：Isolation Forest, One-Class SVM, Autoencoder（重构误差大的为异常）。
 - **优点**：对“不健康”样本数量要求低，甚至可以在只有健康样本的情况下训练（One-Class SVM, Autoencoder）。异常得分可以反向映射为健康度分数。
 - **缺点**：可能无法区分不同类型的“不健康”，只知道它“不正常”。
- **结论(Q4)**：
 1. **首选策略**：在模型训练中使用代价敏感学习（调整类别权重）。
 2. **评估**：必须使用Recall, F1-Score, AUC-PR等不均衡指标。
 3. **备选/补充策略**：谨慎尝试SMOTE过采样；或者，如果“故障”样本极度稀疏且难以定义，将问题转换为**异常检测**可能更有效，使用Autoencoder或Isolation Forest。

6. 模型训练与评估

- **目标变量定义**：
 - **分类**：定义健康状态等级（如：健康、关注、告警、危险），需要历史数据标签（如基于历史故障记录、专家标注）。
 - **回归**：直接预测健康度分数（0-100），需要定义分数的连续标签（可能基于故障发生时间远近、异常指标严重程度等）。
 - **异常检测**：输出异常分数，无需显式标签，但需要定义分数到健康度的映射。
- **数据划分**：采用**时间序列划分**，例如使用较早的数据做训练集，后续数据做验证集和测试集，避免数据泄露。
- **训练过程**：使用验证集进行超参数调优（如学习率、树的数量/深度、网络结构、正则化系数等）。选择在验证集上表现最优（根据不均衡指标）的模型。
- **最终评估**：在独立的测试集上评估模型性能。

7. 健康度评分与应用

- **分数生成**：
 - **分类模型**：使用预测为“不健康”类别的概率，转换成健康分，如 $\text{Health Score} = (1 - P(\text{unhealthy})) * 100$ 。
 - **回归模型**：直接使用预测值。
 - **异常检测模型**：将异常分数归一化并反转，如 $\text{Health Score} = (1 - \text{Normalized Anomaly Score}) * 100$ 。
- **阈值设定**：结合业务需求和模型在验证集/测试集上的表现，设定不同的健康度阈值，对应不同的运维响应级别（如：<30分立即排查，30-60分列入观察计划等）。
- **可视化与集成**：在云平台或监控系统上展示设备健康分地图、列表排名、历史趋势图。将低分设备告警信息推送到运维工单系统。

8. 技术方案总结与挑战

- **总结**：本方案建议采用特征拼接的方式融合遥信和遥测的衍生特征，优先选用XGBoost/LightGBM或MLP作为模型。针对样本不平衡问题，采用代价敏感学习或异常检测方法，并使用合适的评估指标。特征工程和领域知识是关键。
- **挑战**：
 - **数据质量**：原始数据的准确性、完整性至关重要。
 - **标签获取（如果做监督学习）**：获取准确的历史故障或健康状态标签可能困难且耗时。
 - **特征工程复杂度**：需要深入理解业务和数据，设计有效的衍生特征。
 - **模型可解释性**：复杂模型（尤其是深度学习）的决策过程不易解释，可能需要SHAP等工具辅助。
 - **概念漂移**：设备老化、环境变化、固件更新可能导致数据模式变化，模型需要定期重新训练或在线更新。

下一步建议：

1. **数据探索与分析 (EDA)**：深入分析遥信、遥测数据特性，识别关键指标和潜在关联。
2. **初步标签定义**：与专家合作，尝试根据历史记录或规则定义初步的“不健康”标签或时间段。
3. **特征工程实践**：实现基于时间窗口的特征提取。
4. **模型快速验证**：使用代价敏感的XGBoost/LightGBM进行初步建模，评估效果。
5. **迭代优化**：根据初步结果，调整特征工程、尝试不同模型（包括异常检测）、优化超参数。