

**环境要求：**

Hadoop 2.7 + Spark 2.3

Python 3.6

**Python库：**

pyspark、kafka、pandas、scipy、sklearn、tsfel

**模块名：读Kafka**

依赖库：pyspark、kafka

功能：Kafka消费者，保存1小时内的数据。如果读到速率异常告警数据，则获取速率异常时间戳，4分钟后，获得异常时间点前后4分钟的实时解码参数数据，启动后续模块。如果读到多个速率异常告警，则并行处理后续模块。

输入：json格式消息队列

输出：速率异常时间点、异常时间点前后4分钟内带有时间戳的实时解码参数数据

测试环境地址：192.168.1.102:26667

参数消息队列：（共16个有效参数）

topic:fvgiis-wireless-Message

{

"uuid": "31baaadc81f64b6984372964c0172de0",

"device\_sn": "1000000082784e2b",

"nodetype": 1,

"datasource": 1,

"testtime": "2022-06-16 14:06:28",

"ss\_rsrp": -91.350000,

"ss\_rsrq": -13.510000,

"ss\_sinr": 1.680000,

"pucch\_txpower": 22.000000,

"pusch\_txpower": 0.000000,

"dlmcsavg": 1.800000,

"dlmcsbest": 3,

"dlmcsmost": 1,

"ulmcsavg": 17.080000,

"ulmcsbest": 30,

"ulmcsmost": 19,

~~"macthr\_dl": 12.032000,~~

~~"macthr\_ul": -10000.000000,~~

~~"phythr\_dl": -10000.000000,~~

~~"pdsch\_slots": 0,~~

~~"pusch\_slots": -10000,~~

~~"dl\_grants": -10000,~~

~~"ul\_grants": -10000,~~

"cqi\_avg": 11.070000,

"cqi\_best": 13,

"cqi\_most": 10,

"pdsch\_bler": 0.000000,

"pusch\_bler": 0.000000,

~~"band": -10000,~~

~~"bandwith": -10000.000000,~~

~~"earfcndl": -10000,~~

~~"pci": 448,~~

~~"tac": -10000,~~

~~"cellid": -10000,~~

~~"slottimingpattern": -10000,~~

~~"specialslotsymbolpattern": -10000,~~

~~"referencesubcarrierspacing": -10000,~~

~~"networktype": 80,~~

~~"networkstate": -10000,~~

"timesec": 1655359588,

"device\_id": 201,

"scene\_id": 20,

"dt": "20220616",

"dt\_hour": "2022061614",

"dt\_min": "202206161406",

"dt\_sec": "20220616140628",

"net\_type": "5G",

"p\_scene\_id": 19

}

速率异常告警消息队列：

topic：fvgiis-ai-alarm

{

"uuid": "dc17c0ce2386484caddaeef72407ce86",

"deviceSn": "1000000084d49210",

"dataType": 5,（数据类型：1=高时延 5=ftp 2=灌包 0=无线参数）

"dtSec": "20220418140239"

}

数据需要处理成为含有以下字段的pandas.DataFrame：

dt\_sec,

SS\_RSRP,SS\_RSRQ,SS\_SINR,

PUCCH\_TxPower,PUSCH\_TxPower,

Pathloss,

MCSAvg\_DL,MCSBest\_DL,MCSMost\_DL,MCSAvg\_UL,MCSBest\_UL,MCSMost\_UL,

CQI\_Avg,CQI\_Best,CQI\_Most,

PDSCH\_BLER~~,PDSCH\_iBLER,PDSCH\_rBLER~~,PUSCH\_BLER~~,PUSCH\_iBLER,PUSCH\_rBLER~~

另外还要带上异常信息相关的3个参数，Kafka写入时会用到：

uuid，

device\_sn，（由deviceSn改名）

abnormal\_sec（由dtSec改名）

**模块名：时序数据采样**

Python脚本文件：aidiag\_ts\_sample.py

依赖库：pandas

功能：将带有时间戳的实时解码参数数据整理成按秒采样的参数数据，范围为异常时间点前后4分钟。如果该秒没有数据则留空；如果该秒有多个数据则取平均值。该模块的输出可以保存到数据库作为模型离线训练的数据集。

接口名：def ts\_sample(df\_raw: DataFrame) -> DataFrame（pandas.core.frame.DataFrame）

输入：速率异常时间点、异常时间点前后4分钟内带有时间戳的实时解码参数数据

输出：速率异常时间点、异常时间点前后4分钟内按秒采样的参数数据（DataFrame对象）

时序数据字段：

sec,（按秒采样的时间）

SS\_RSRP,SS\_RSRQ,SS\_SINR,

PUCCH\_TxPower,PUSCH\_TxPower,

Pathloss,

MCSAvg\_DL,MCSBest\_DL,MCSMost\_DL,MCSAvg\_UL,MCSBest\_UL,MCSMost\_UL,

CQI\_Avg,CQI\_Best,CQI\_Most,

PDSCH\_BLER,PDSCH\_iBLER,PDSCH\_rBLER,PUSCH\_BLER,PUSCH\_iBLER,PUSCH\_rBLER

DataFrame固定为480行（8分钟）

**模块名：Pandas数据处理**

Python脚本文件：aidiag\_ts\_transform.py

依赖库：pandas、scipy

功能：对按秒采样的参数数据进行进一步数据转换处理，如线性插值

接口名：def ts\_transform(df\_param: DataFrame) -> DataFrame

输入：速率异常时间点、异常时间点前后4分钟内按秒采样的参数数据（DataFrame对象）

输出：转换后的参数数据（DataFrame对象）

**模块名：tsfel时序特征提取**

Python脚本文件：aidiag\_ts\_feature.py

依赖库：tsfel、pandas

功能：从转换后的参数数据中，提取统计特征与时序特征。该模块代码和离线训练时的代码相同，可以复用。

接口名：def ts\_feature(df\_data: DataFrame) -> numpy.ndarray

输入：转换后的参数数据（DataFrame对象）

输出：时序特征（numpy矩阵）

**模块名：模型预测**

Python脚本文件：aidiag\_predict.py

依赖库：sklearn

功能：将时序特征输入已经训练好的模型，得到预测的故障原因

接口名：def predict(feature: numpy.ndarray, model: model) -> dict

输入：时序特征（numpy矩阵）、已经离线训练好的模型（model文件）

输出：模型预测的故障原因概率分布（覆盖/干扰/其他）（dict）

输出数据样例：

{

"predict\_reason": “弱覆盖”,

"predict\_reason\_prob": {

“弱覆盖”: 0.6,

“干扰”: 0.3,

“其他”: 0.1,

}

}

**模块名：写Kafka**

依赖库：pyspark、kafka

功能：Kafka生产者，将模型预测的故障原因与相应的异常时间点一起写入消息队列。

输入：速率异常时间点、模型预测的故障原因（覆盖/干扰/其他）（dict）

输出：json格式待定

输出数据样例：

topic：fvgiis-ai-predict

{

"uuid": "31baaadc81f64b6984372964c0172de0",（与异常告警发生信息一致）

"device\_sn": "1000000082784e2b",（与异常告警发生信息一致）

"abnormal\_sec": "20220616140628",（与异常告警发生信息一致）

"predict\_reason": “弱覆盖”, （模型预测的故障原因）

"predict\_reason\_prob": { （模型预测的故障原因概率分布）

“弱覆盖”: 0.6,

“干扰”: 0.3,

“其他”: 0.1,

}

}