Data Processing Group

writer: 王骏飞, version 2023-05-18

一、文件清单

2
main.py: 动态阈值计算单元以及清洗数据保存单元的入口。

AnomalyDetection.py: 异常检测（静态阈值与多传感器异常检测）的入口。

IO_API.py: 读写数据使用的模块，包括读kafka以及写mysql。

Processor.py: 动态阈值部分数据预测使用的模块，性能较差，但够用。

env.py: 用以同一提供配置信息的模块，可以定时读取文件从而更新配置信息。

二、依赖

见requirments.txt，可使用以下命令安装。

```
pip install -r requirements.txt
```

三、配置文件

配置文件统一放在./configs路径下，同一采用json格式，目前有如下若干个配置文件：

1. config-for-pds.json

   pds为PredictDataSaver的缩写，是将预测数据写入mysql服务器的配置文件，可配置mysql服务器的信息。

2. config-for-wds.json

   wds为WashedDataSaver的缩写，是将清洗后数据写入mysql的配置文件。

3. detector.json

   异常检测组件的配置文件，目前主要包括设备间最大差值、设备各测量值上限阈值以及有效数据期限三部分。暂时没有设置静态的下限阈值。支持热修改。

   max_diff字段以及max_threshold字段，顾名思义，前者是各设备之间的最大数据差值，后者是静态阈值检测。

4. detector-reader.json

   异常检测使用的KafkaReader的配置文件。

5. detector-writer.json

   异常检测需将异常信息写入数据库，此为对应的mysql的配置信息。

6. env-config.json

   对于动态阈值以及wds最重要的环境配置文件，包括了预测时间步数、缓冲区尺寸、动态阈值容忍偏移上限、是否检测计算错误以及有效数据期限等重要的、可能更改的环境参数，支持热修改。

   该配置文件可以被热修改，主要是tolerate字段可能需要修改，即阈值上界偏移预测值的最大差值。输入到数据库中的数据等于预测值+tolerate对应的值。

7. reader-config.json

动态阈值以及wds的读配置文件。之所以与detector-reader区分，是因为它们要使用不同的kafka消费者分组。

四、数据库数据说明

目前建立了三个表，所有表都拥有两个时间戳，一个是insert_timestamp表示插入数据的时间，另一个是timestamp表示数据本身（信息）的时间。

1. washed：保存了清洗后的定时数据
2. upper_bound：保存了动态阈值的上界的数据，配合washed表对照可以判断是否超出阈值。
3. exception：分列保存了异常信息。如果没有发生异常，则不插入数据，如果发生了异常，则出现异常的指标的列上会有异常信息，没有异常的指标的列为NULL。other列目前存储设备掉线信息，如果设备长时间掉线会报出异常。
4. 特别说明一下设备间异常信息的格式：Exception exist in device group(s): [“设备1”,"设备2""]，即两设备之间有异常。如果出现多个设备在一个[]内，可通过其他[]内的数据辅助判断具体哪个设备异常。

   可以结合数据库内的清洗后数据来进一步判断哪个设备出现异常。

五、可能出现的异常

1. env,py通过多线程+定时来更新配置，但没有考虑同步问题，极小概率出现异常。
2. 数据清洗组件重启或本程序重启可能造成数据库中少量数据重复，可以通过insert_timestamp来找到最新数据。
3. 设备掉线超过某个时间（依赖于数据清洗组件的设置）并恢复后的一段时间内，因为样本量不足，无法给出动态阈值。
4. 可能会出现很长的报错信息，是由于arima对于数据有较高的要求导致的（只要不满足就无法计算出结果，于是只能使用最后的值进

行替代）。但只要程序仍在运行，就无需中止。

5. 该程序已经过持续运行测试，也尽可能对可能出现的exception进行捕获，如果出现极特殊的没有考虑到的情况导致运行中断，在排查了服务器（包括kafka以及mysql）的异常后，直接重新运行即可。