Detect API

作者：周泽龙
邮箱：zzl850783164@163com
日期：2019年11月1日

运行环境

- Language: python 3.6.6
- Package:
  - Flask: 1.1.1
  - elasticsearch: 5.3.0
  - numpy: 1.17.3
  - requests: 2.22.0
  - scikit-learn: 0.21.3

文件结构

```
ApiDetect
|-- AnomalyDetection.py      # 执行异常检测算法类
|-- app.py                   # flask app
|-- ElasticSearch.py         # Elastic 搜索类
|-- Global.py                # 全局变量类
|-- MyThread.py              # 多线程类
```

> `app.py`
>
> > flask app，运行后监听本地端口：5000
> >
> > 目前包含接口 URL: /api/v1/detect

- 向接口发送 request 请求后，后台操作如下：
  - 解析 request 请求，得到：数据集提取范围，异常检测算法名称
  - 根据数据集提取范围，从 elasticssearch 中分别提取训练数据集和检测数据集
  - 根据异常算法名称，分别执行异常检测
  - 若有超过 `threshold` 数量的算法返回异常，则最终判断有异常
  - 向 "http://47.95.199.184/api/v1/report" 发送检测报告

> `ElasticSearch.py`
>
> > 执行 elastic 数据搜索

- 当创建一个 `ElasticSearch` 类时，传入参数：`index` 和 `type`

- 执行 `getSearchResult()` 函数
  - 参数为 `json` 格式的 `body`，如下：

```
{
    "query": {
        "range": {
            "timestamp": {
                "lte": timeNow,
                "gte": time60SecondsBefore
            }
        }
    }
}
```

  - 使用 scroll 分页
  - 返回值：所有的结果集列表

> `AnomalyDetection.py`
>
> > 执行异常检测，目前支持 “Isolation Forest” 和 “Three Sigma” 两种算法

- 当创建一个 `AnomalyDetection` 类时，需要传入 `json` 格式的算法说明，如下：

```
{
    "name": "iForest",                      #string字符串
    "param": "{\"threshold\": 0.1}",        #sting格式的json字符串
}
```

- 执行 `run()` 函数
  - 参数为训练数据集和检测数据集，它将新建一个线程执行异常检测算法
  - 返回值：`True` — 有异常；`False` — 无异常
- 后续扩展，添加新的异常检测算法
  - 在 `AnomalyDetection.py` 中实现新算法，或在其他文件下也可
  - 在 `Global.py` 中加入新算法名称
  - 在 `AnomalyDetection.run()` 函数中，根据算法名称选择算法，新建线程执行该算法