

## KPI 异常检测科研问题的原始出处

1. 登录 <http://iops.ai/>, 注册账号
2. 进入到“竞赛”模块, 找到 KPI 异常检测页面:  
[http://iops.ai/competition\\_detail/?competition\\_id=5&flag=1](http://iops.ai/competition_detail/?competition_id=5&flag=1)
3. 预览查看问题描述, 并下载数据集

The screenshot displays the 'KPI异常检测' (KPI Anomaly Detection) competition page on the iops.ai platform. The page is divided into several sections:

- Header:** Includes the 'AIops Challenge' logo, navigation tabs for '运维场景' (Operations Scenarios), '数据集' (Data Sets), '竞赛' (Competition), and '科研问题' (Research Problems). A search bar is present with the placeholder text '请输入你想要搜索的内容' (Please enter the content you want to search for). User information for 'jasper' is shown in the top right corner.
- Competition Details:** Displays 'KPI异常检测' (KPI Anomaly Detection) with a subtitle '来自搜狗, 腾讯, eBay, 百度, 阿里巴巴的多条KPI曲线, 有异常标注' (Multiple KPI curves from Sogou, Tencent, eBay, Baidu, and Alibaba, with anomaly annotations). It also shows '353支参赛队伍' (353 participating teams), '竞赛剩余时间-180天' (Competition remaining time - 180 days), and '奖金¥92000' (Prize money ¥92,000).
- Navigation Tabs:** A row of tabs includes '预览' (Preview), '数据集' (Data Set), '讨论' (Discussion), '排行榜' (Ranking), and '规则' (Rules). The '预览' tab is highlighted with a red box.
- Problem Description:** The '描述' (Description) tab is selected. It contains text explaining the importance of web service stability and the role of KPI monitoring. A line graph titled '图1 异常KPI示例' (Figure 1: Example of abnormal KPI) shows a fluctuating line with two red circles highlighting specific anomalies. The x-axis is labeled with days of the week (Mon, Tue, Wed, Thu, Fri, Sat, Sun).
- Footer:** A row of tabs at the bottom includes '描述' (Description), '数据集' (Data Set), '讨论' (Discussion), '排行榜' (Ranking), and '规则' (Rules). The '描述' tab is highlighted with a red box.

<div> <div> </div> <div> <a href="#">运维场景</a> <a href="#">数据集</a> <a href="#">竞赛</a> <a href="#">科研问题</a> </div> <div> <input type="text" value="请输入你想要搜索的内容"/> </div> <div> <a href="#">知识库</a> <a href="#">论坛</a> </div> <div> <a href="#">jasper</a> <a href="#">退出</a> <a href="#">中文</a> </div> </div>				
关联数据集				
数据集名称	文件名称	文件描述	上传时间	操作
KPI异常检测决赛数据集	KPI异常检测决赛数据集.zip		2018-05-24 16:43:34	<a href="#">下载</a>
KPI异常检测预赛测试集	test.zip		2018-01-25 09:44:57	<a href="#">下载</a>
KPI异常检测预赛训练集	train.zip		2018-01-25 09:43:33	<a href="#">下载</a>

## 问题描述

随着互联网，特别是移动互联网的高速发展，web 服务已经深入到社会的各个领域，人们使用互联网搜索，购物，付款，娱乐等等。因此，保障 web 服务的稳定已经变的越来越重要。

Web 服务的稳定性主要靠运维来保障，运维人员通过监控各种各样的关键性能指标（KPI）来判断 Web 服务是否稳定，因为 KPI 如果发生异常，往往意味着与其相关的应用发生了问题。图 1 中展示了一个 KPI 异常的例子——某互联网公司的网页访问量数据发生了异常，其中红色圆圈标示了 KPI 发生的异常。

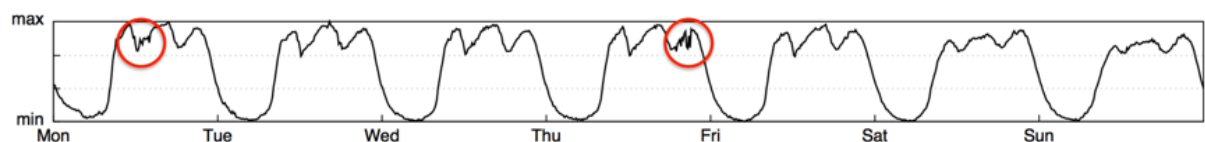


图 1 异常 KPI 示例

这些 KPI 大致分为两种类型：服务 KPI 和机器 KPI。服务 KPI 是指能够反映 Web 服务的规模、质量的性能指标，例如，网页响应时间，网页访问量，连接错误数量等。机器 KPI 是指能够反映机器（服务器、路由器、交换机）健康状态的性能指标，例如，CPU 使用率，内存使用率，磁盘 IO，网卡吞吐率等。

KPI 异常检测指的是通过算法分析 KPI 的时间序列数据，判断其是否出现异常行为。这里的难点主要有：

- 1 异常发生的频率很低。在实际的运维场景中，业务系统很少发生异常，因此可供分析的异常数据很少。
- 2 异常种类的多样性。因为实际的业务系统很复杂，并且会不断更新升级，所以故障的类型多种多样，从而导致了异常种类的多样性。
- 3 KPI 的多样性。KPI 有表现为周期型的，有表现为稳定型的，有表现为不稳定的，持续波动型，如图 2，3，4 所示。

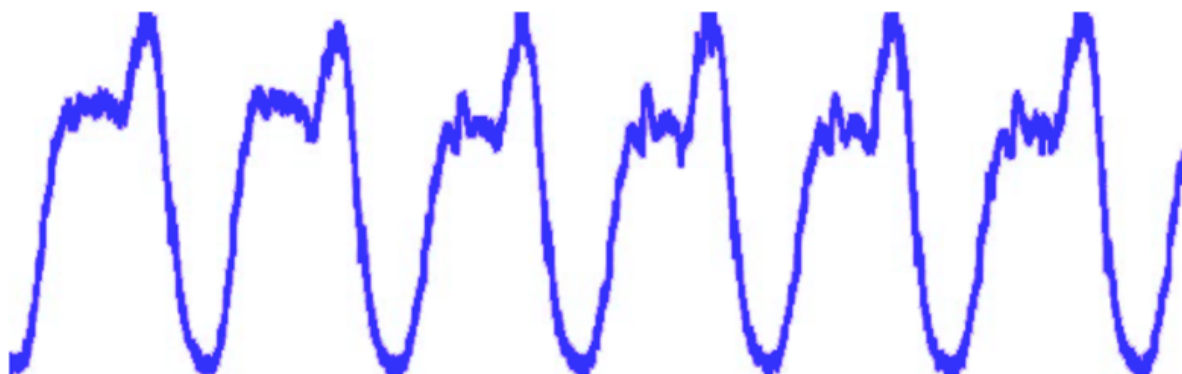


图 2 周期型 KPI



图 3 稳定型 KPI

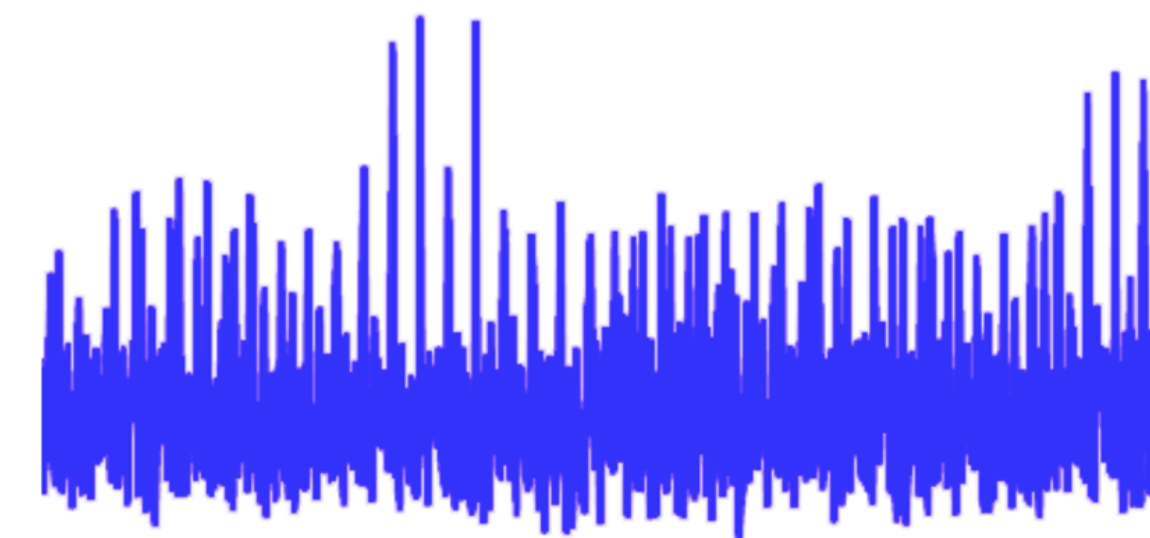


图 4 不稳定型 KPI

正是因为这些难点，导致现有的异常检测算法的准确率（precision）和召回率（recall）都不高，存在大量的误报和漏报。这不仅增加了运维人员的工作量，而且导致运维人员无法及时准确地发现异常的 KPI。

因此，为了提高异常检测算法的准确率和召回率，我们收集了来自众多互联网公司的真实场景的 KPI 数据，经过脱敏处理后提供出来，希望更多的学术界和工业界的人员参与进来，一起提高异常检测算法的性能。

## 异常检测的性能评价指标和评估程序

设计的异常检测算法需要在每一个时间点进行一次异常检测。根据异常检测结果，输出判定结果 1（异常）或 0（正常）。在此基础上，本挑战赛计算异常检测结果的 F-score。

通常情况下，运维人员往往只关心异常检测算法能否检测到某一连续异常区间，而非检测到该异常区间的每一个异常点。因此，我们以如下方式计算 TP, TN, FP, FN:

1 对于一段标记的连续异常区间:

如果异常检测算法在该连续异常区间开始后的 不晚于  $T$  个时间点内检测到了该连续异常区间，本挑战赛认为此异常检测算法成功地检测到了整段连续异常区间，因此该异常区间内的每一个异常点都算作一次 true positive (TP)；否则，该连续异常区间内的每一个异常点都算作一次 false

negative (FN)。

2 对于一个没有标记异常的时间点：

如果异常检测算法输出了异常，计做一次 false positive (FP)；否则，计做一次 true negative (TN)。

精度 (precision)，召回率 (recall) 和 F-score 计算方式：

精度 =  $TP / (TP + FP)$

召回率 =  $TP / (TP + FN)$

F-score =  $(2 * 精度 * 召回率) / (精度 + 召回率)$

评价指标示例：

标注	0	0	1	1	1	0	0	1	1	1
算法输出	1	0	0	1	1	1	0	0	0	1
调整后的算法输出	1	0	1	1	1	1	0	0	0	0

如图所示，当  $T=1$  时，上图中异常检测算法成功地检测到了第一个连续异常区间，而未能成功检测到第二个连续异常区间。因此，本挑战赛认为算法的准确率为 0.6，召回率为 0.5。

但是这种评价指标检测报出的区间异常是存在一定延迟 (delay) 的，比如在上图中，第一个异常区间的  $delay=1$ ，第二个异常区间的  $delay=2$ 。考虑到在实际场景中，运维人员希望报出异常的延迟不要太高，所以这里我们定义一个 delay threshold。也就是说，当超过 threshold 个点还没有被报出来时，就不扩展到整个区间。在上图中，假设 delay threshold 设为 1，那么第二个区间被认为没有检测出来异常。

在本次比赛中，我们暂定 delay threshold =7。主办方保留在极特殊情况下修改 delay threshold 取值的权利。

提交结果要求：

1 提交结果必须为 csv 文件，文件名必须为英文。

2 csv 文件格式示例：

KPI ID,timestamp,predict

0,1501475700,1

0,1501475760,1

注意：提交 csv 文件 header 必须为：KPI ID, timestamp, predict

3 KPI ID 及 timestamp 列必须与测试集一一对应。

注：评估脚本及样例文件请见 [Evaluation](https://github.com/iopsai/iops/tree/master/evaluation)  
<https://github.com/iopsai/iops/tree/master/evaluation>

主办方保留对评估指标进行更新的权利。