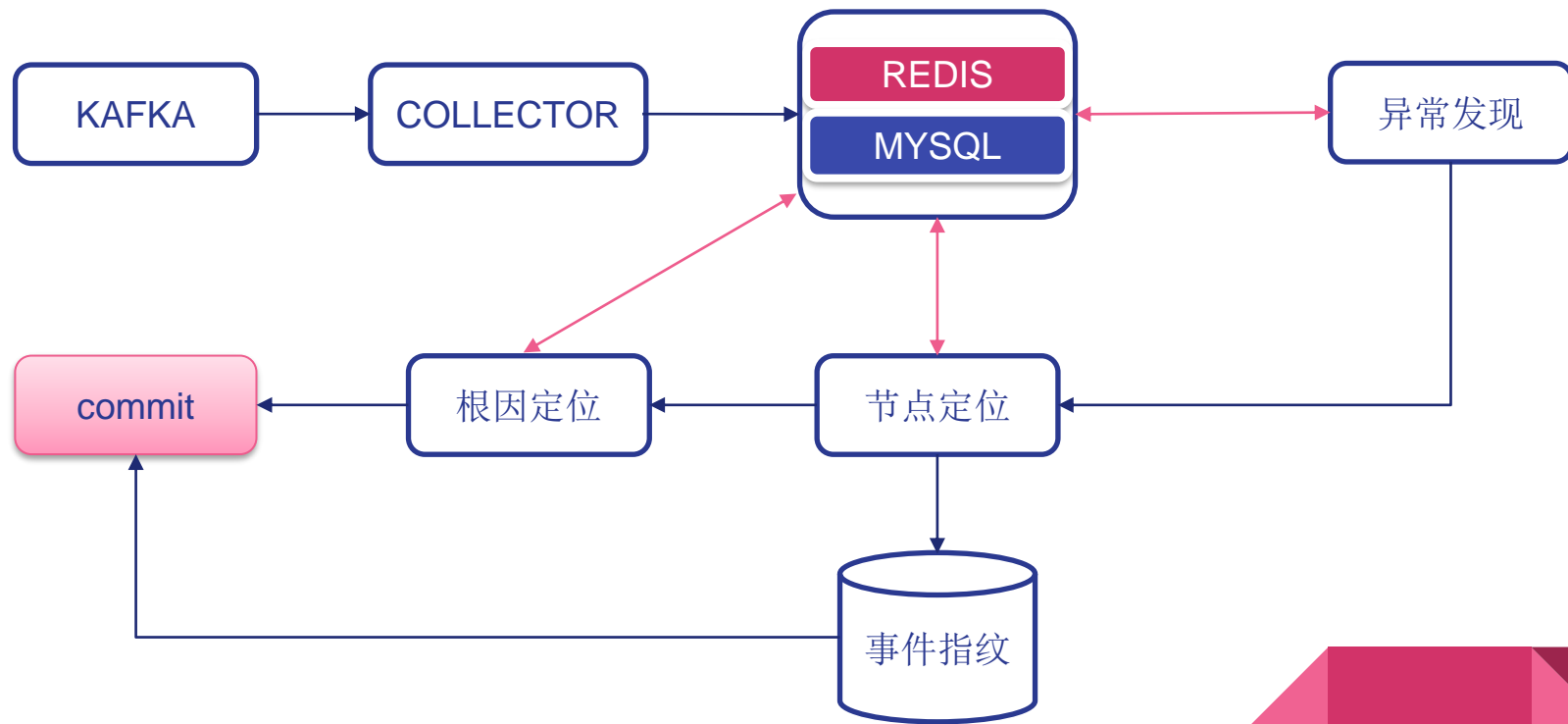


Benjili战队

August 2020, hangzhou

应用架构

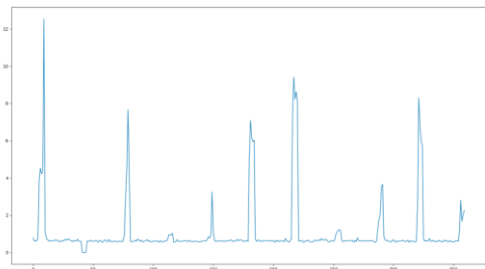
架构图



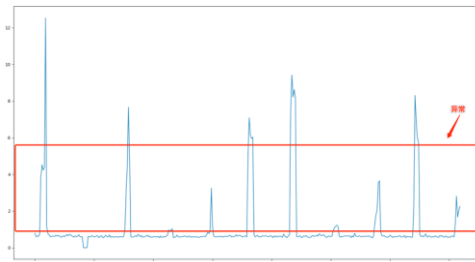
异常发现

异常发现-时延&量

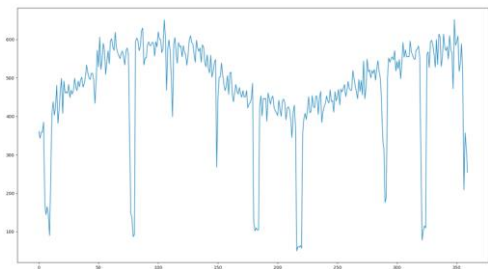
时延



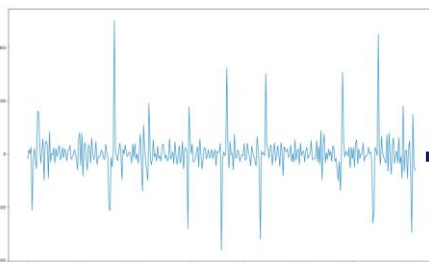
3sigma



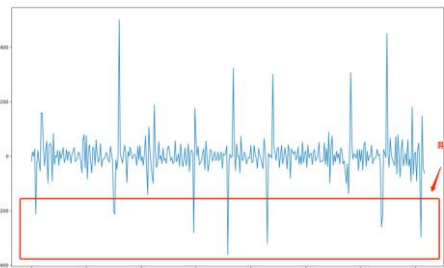
量



diff



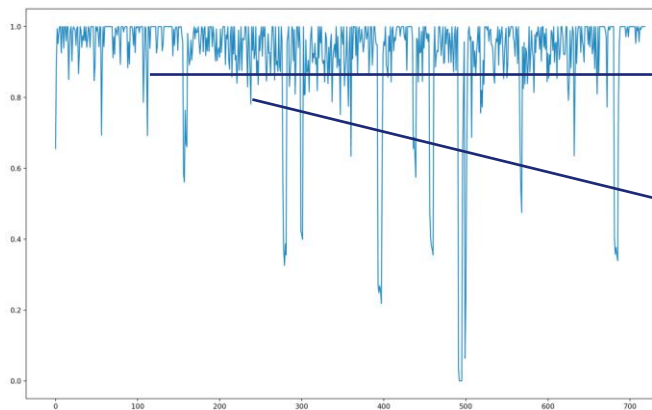
3sigma



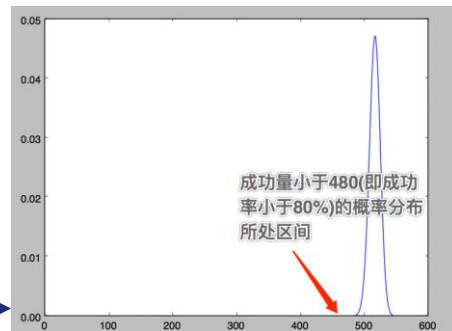
异常发现-率

率

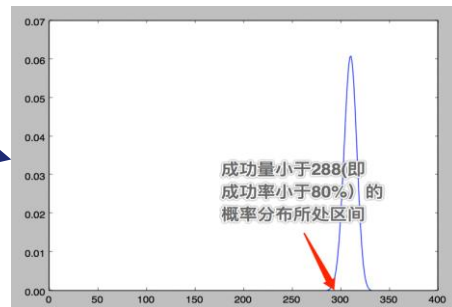
- 在交易量不同的情况下，同样的成功率异常程度不一样



交易量600的
伯努利分布

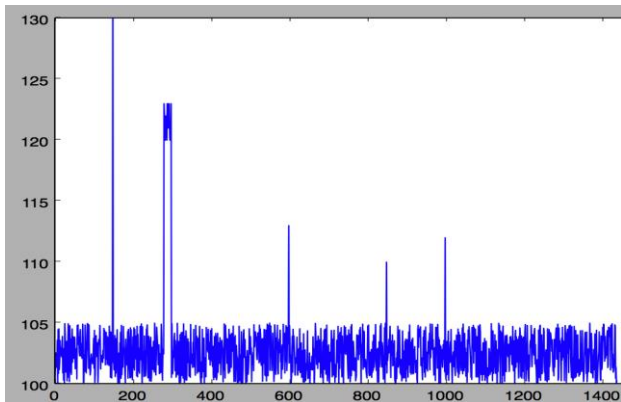


交易量360的
伯努利分布

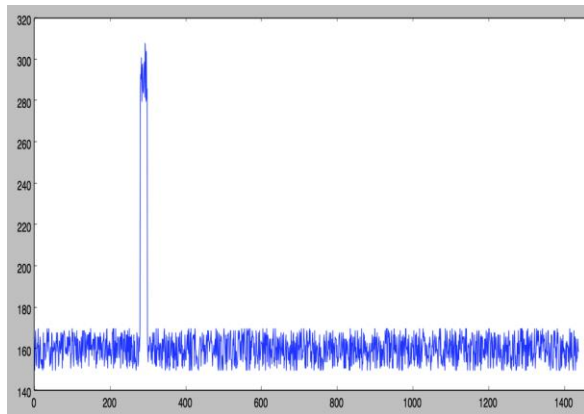


异常收敛

准确的汇聚方式，会放大异常的程度。如下图，假如只用耗时平均值(图一)，会产生很多噪音。而如果我们转换一下，用超出历史耗时中位数的交易笔数分析(图二)，则很多噪音会被过滤掉



图一



图二

节点定位

异常链路收集-耗时、成功率、链路中断

os_021_osb->docker_001

docker_001_local_method->docker_007

docker_007_jdbc->db_003

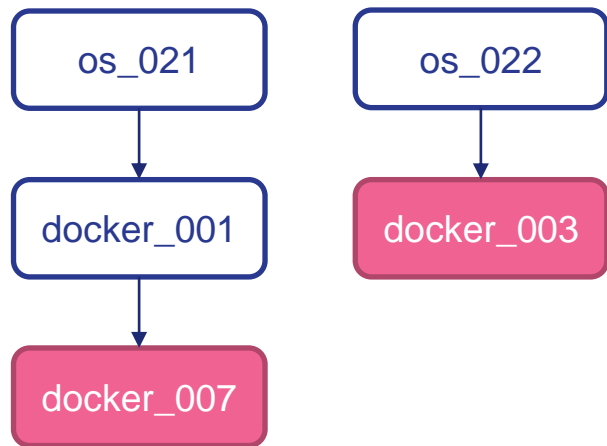
os_022_osb->docker_003

os_022_osb->docker_002

os_021_osb->docker_004

docker_008_jdbc->db_007

.....



事件指纹

Tanimoto系数: $E_j(A, B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B}$ (A 当前异常事件向量 B 对比异常事件向量)

节点id(onehot)

异常节点上下游

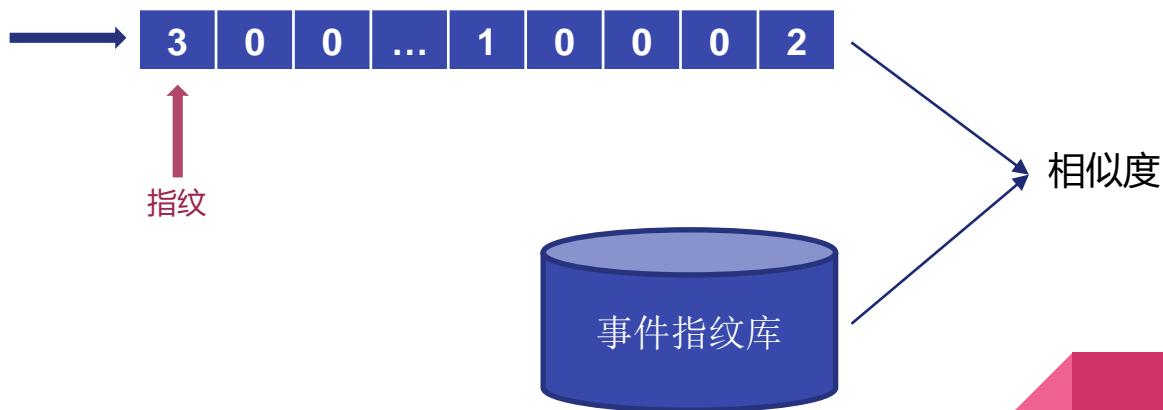
链路存在中断

存在共同上层容器

时延上涨

成功率下降

.....



根因定位

数据采集

采集异常样本

- 预赛各阶段以及复赛第一轮中出现的性能指标异常

正负样本不平衡

- 对负样本进行过采样以达到正负样本 1:1

样本权重增强

- 通过过采样来增加异常区间起始点的权重



特征提取

单个数据点特征

- 原始值
- 分位数
- 异常程度（基于高斯分布）

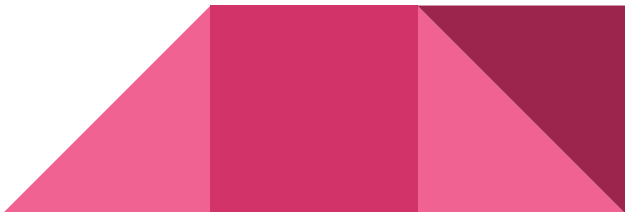
时间序列特征

- 一阶差分
- 异常程度（基于高斯分布）
- 变化比例
- 分位数

时间窗口特征

- 多种窗口宽度
- 均值、方差
- 异常点所占比例

事件指纹特征

- 节点id
 - 异常节点上下游
 - 链路存在中断
 - 存在共同上层容器
 - 时延上涨
 - 成功率下降
- 

模型介绍

尝试过的模型

➤ 逻辑回归

- 可解释性强 速度快 实际表现一般

➤ 随机森林

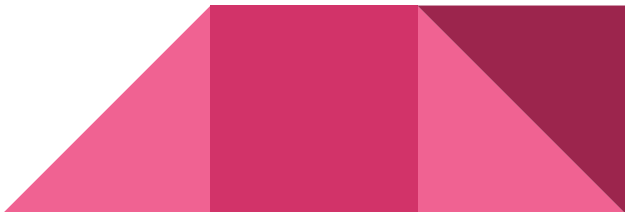
- 实际表现稳定 泛化能力好 速度较快

➤ LSTM

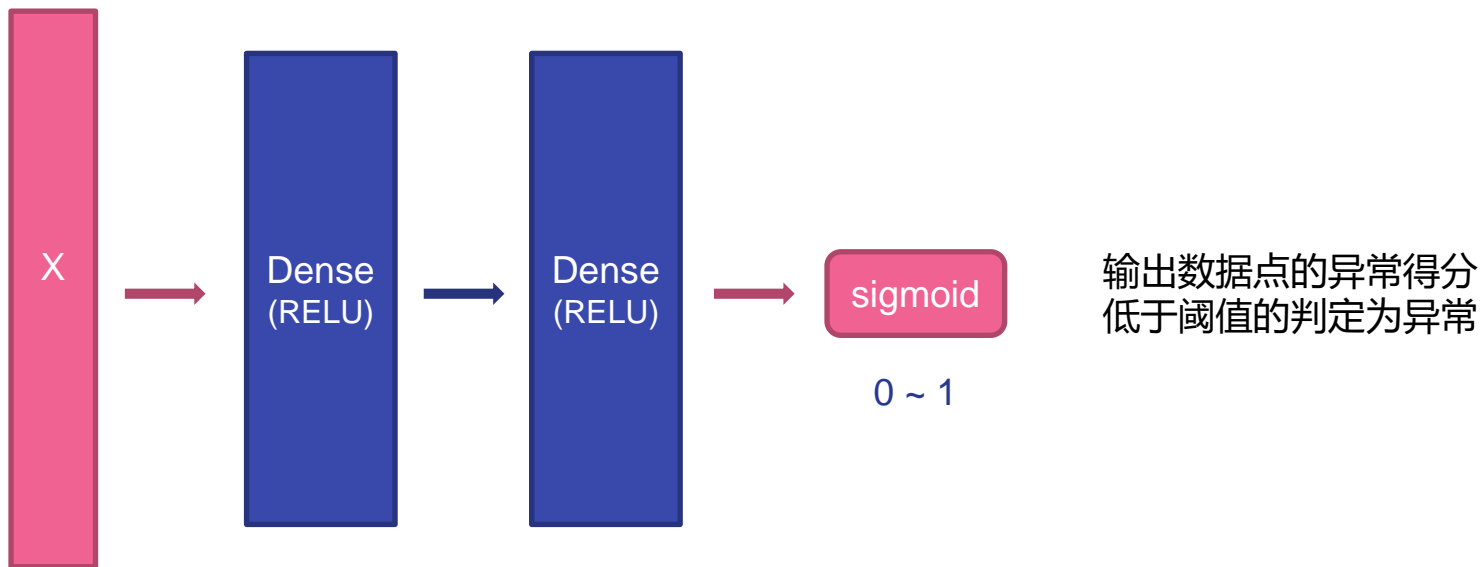
- 实际表现一般 速度较慢

➤ DNN

- 泛化能力强 实际表现最好



模型选择-DNN




- 两个64维的全连接层和一个sigmoid输出层
- dropout和L1正则化项减少模型的过拟合

总结与思考

总结

- 自适应不同类型的KPI，自动提取特征，通用性强
- 系统架构合理，算法鲁棒性强，表现稳定
- 事件指纹帮助故障提前发现

思考

- 模型参数的优化
 - 丰富异常检测维度
 - 融入运维专家知识
- 



WeSmartOps



williamxue666



谢谢大家