Project PPT Notes

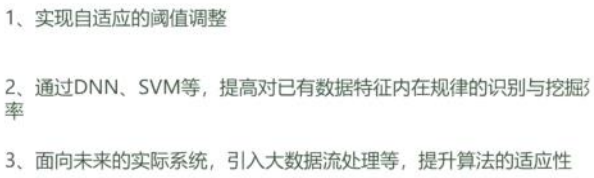
2020年11月30日 17:27

Tsinghua
- KDE for Anomalies Detection
- Topology for anomaly node location
- They added personalized rules to their models and its not told

H3C
- Locate which 网元 has problem. Example D1 calling D2 takes long. Then D3 call D2 takes long too. Then D2 has a problem. Similar for others
- Offline Preprocessing

BOCOIOPS
- Data -> Filtering -> Multi Threading -> Caching -> Root Cause Detection
- The call time that takes the longest
- Analysis Call steps relationship and success or not
- Checking on dockers, if dockers in same os has problem, then OS has problem

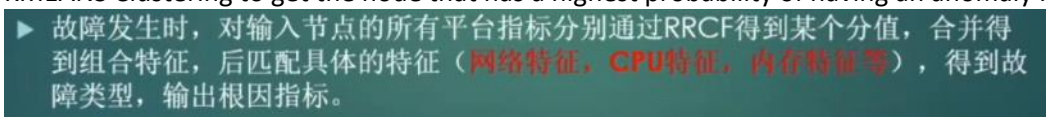
-

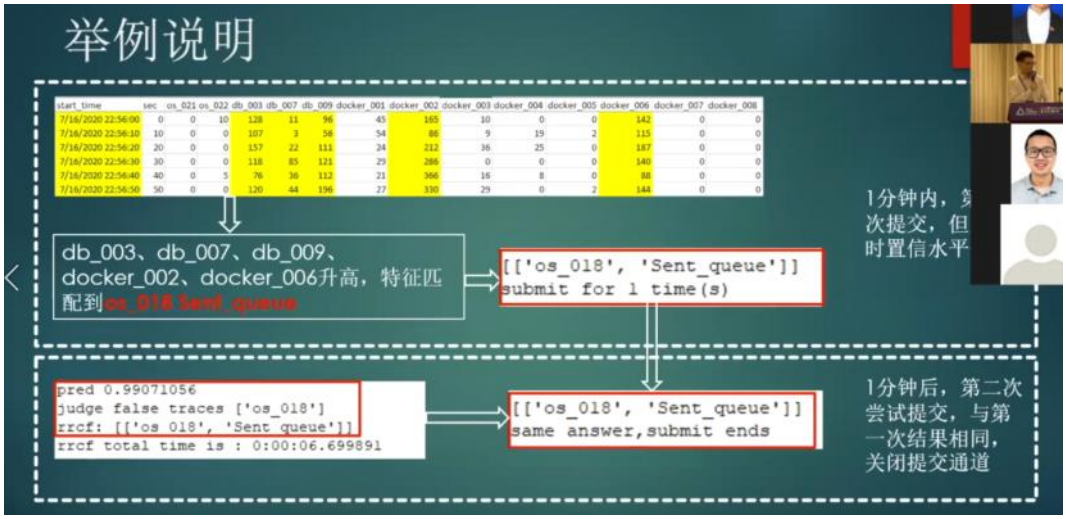


掘金人
- Real time data is moving
- Anomalies might be shorter than 1 mins
- Data representing the situation might be delayed

- Cut the 1 mins API calling data into 10s interval
  - Evaluate on success parameter
  - Elapsed time of a single node
  - The processing time of each nodes matches a certain trait
- After the 1 mins interval
  - Will evaluate on this 1 min interval
  - Using (OSB) data to train a XGBOOST classification model, real time classification
  - Using the "success" parameter in Service Data to help in analyzing

- 

- First Submit - compare the anomaly to current known anomaly
- Second Submit
  - Processed call features
  - Compare processed one with the characteristics of usual features
  - Use KMEANS to get the node that most probably have a problem
  - Using RRCF model to do feature matching with the normal platform data
  - Second Submit
- Most Submit will trigger a checking if the submit meets a threshold
- Use K=1 KNN algo to get the matching characteristic of data then get its type (Root Cause)
- Keep the call trait when running normally, get its different with the data obtained during anomalies occurred, Then use KMEANS Clustering to get the node that has a highest probability of having an anomaly for RRCF input
- 

- 

亚信科技

Trace calling chain and tree node tree

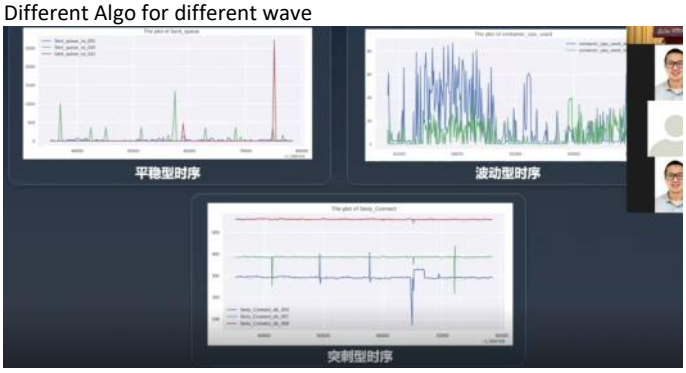
Output Tree

1. Anomaly
2. Seeming Anomaly - every node takes time

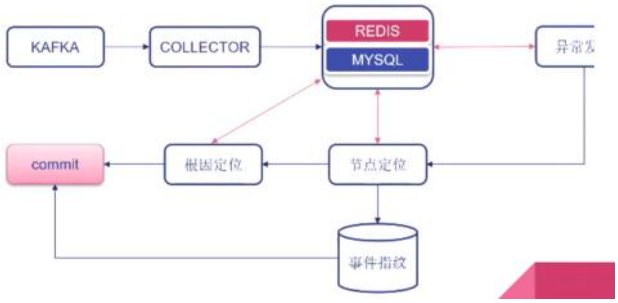
Input: Tree Nodes
Output: node

1. Check every level
2. $$condition: \frac{Tree[depth][status]}{\sum Tree[depth][status]} \geqslant 90\%$$

Different Algo for different wave



微众银行



1. If there are similar historical event, give the root cause
2. Or else locate the node then lo

3.

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

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

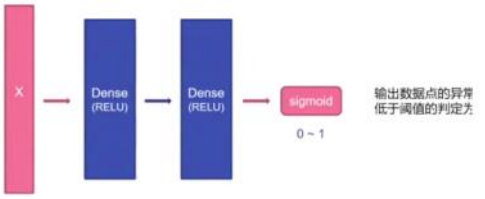
.....

4. cate the root cause

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



- 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