

AiCon

全球人工智能与机器学习技术大会

机器学习加持下的 时序类数据异常智能监控

刘彪

高级工程师@腾讯



极客时间

重拾极客精神·提升技术认知

下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新



扫一扫下载极客时间App

人工智能基础课

“通俗易懂的人工智能入门课”

王天一
博士
副教授



扫一扫，免费试读

AI技术内参

你的360度人工智能信息助理

洪亮劫
Etsy
数据科学主管



扫一扫，免费试读



关注落地技术，探寻AI应用场景

- 14万AI领域垂直用户
- 8000+社群技术交流人员，不乏行业内顶级技术专家
- 每周一节干货技术分享课
- AI一线领军人物的访谈
- AI大会的专家干货演讲整理
- 《AI前线》月刊
- AI技能图谱
- 线下沙龙



扫码关注带你涨姿势

QCon

全球软件开发大会

成为软件技术专家的 必经之路

[北京站] 2018

会议：2018年4月20-22日 / 培训：2018年4月18-19日
北京·国际会议中心

8折 购票中, 每张立减1360元
团购享受更多优惠



识别二维码了解更多

ArchSummit

全球架构师峰会

2018 · 深圳站

从2012年开始算起，InfoQ已经举办了9场ArchSummit全球架构师峰会，有来自Microsoft、Google、Facebook、Twitter、LinkedIn、阿里巴巴、腾讯、百度等技术专家分享过他们的实践经验，至今累计已经为中国技术人奉上了近千场精彩演讲。

限时**7折**报名中，名额有限，速速报名吧！

- 2012.08.10-12 深圳站
- 2018.07.06-09 深圳站
- 会议：07.06-07.07
- 培训：07.08-07.09



TABLE OF CONTENTES

背景

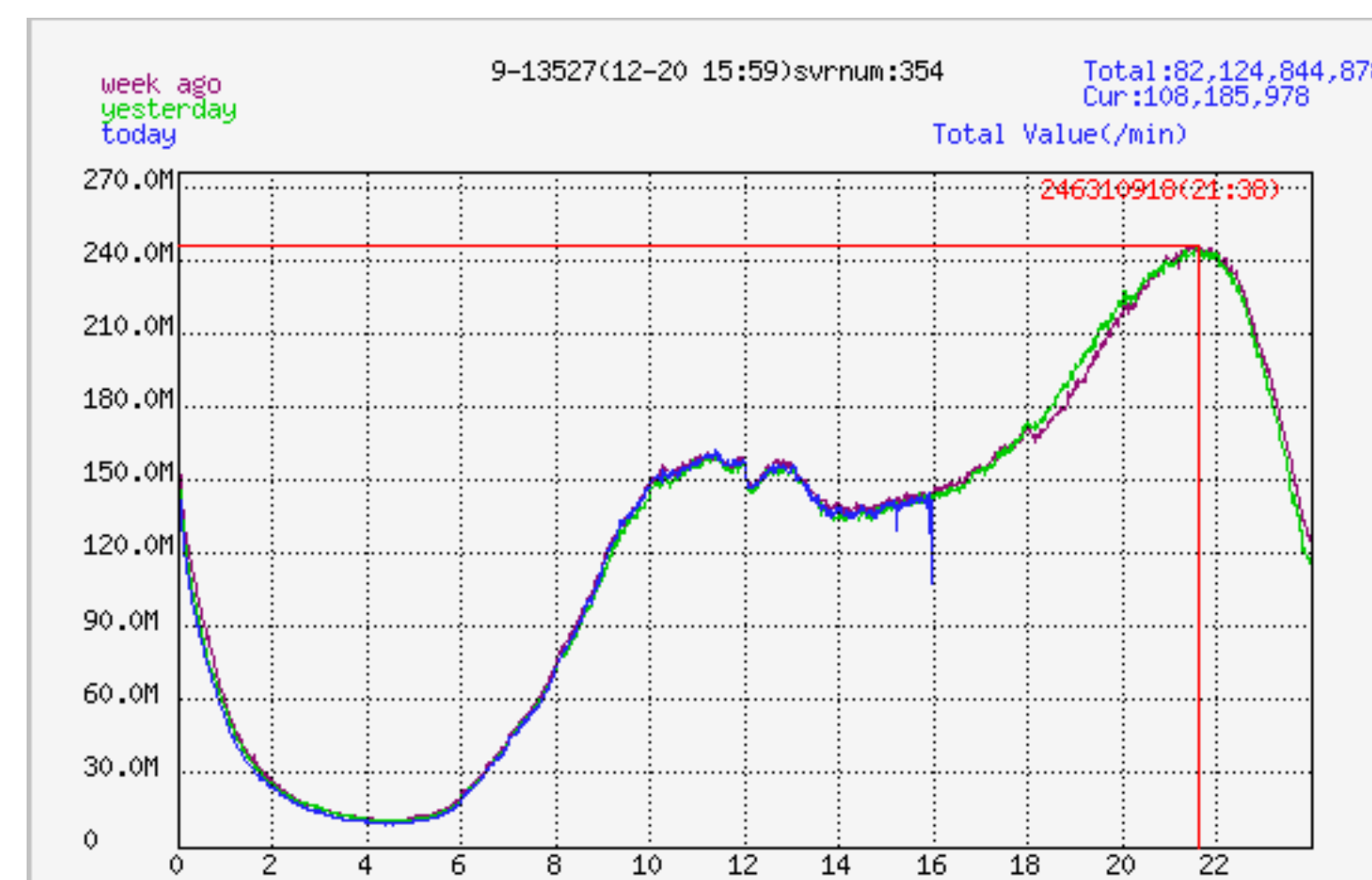
算法探索

落地

下一步

监控系统: Monitor

250万属性
20人



Monitor的问题

运维成本高

4万/天, 200条/人

准确率极低、6%

理想中的监控

一个零

- 触发条件“0”维护

两个九

- 准确率：90%
- 召回率：90%

TABLE OF CONTENTES

背景

算法探索

落地

下一步

GBDT初探

初探：

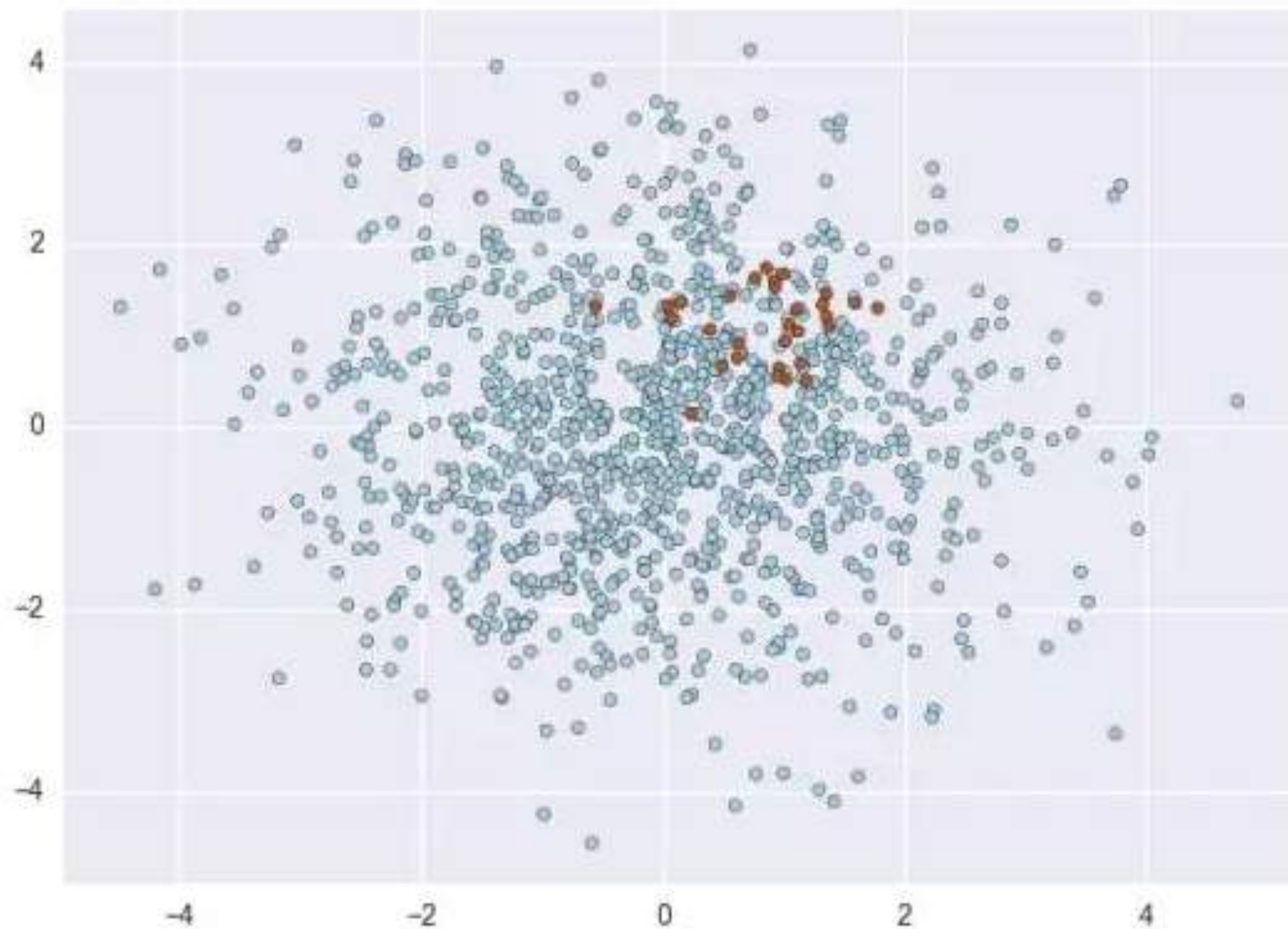
第几轮	样本	正类	负类	正类占比	召回率	准确率	精确率	11个视图, 8024个属性	备注
第一轮	1383	541	842	39%	81%	96%	86%		未达预期
第二轮	5707	5285	422	92%	96%	96%	98%	04月30日：13%异常率	过多异常
第三轮	20298	5285	15013	26%	99%	92%	96%	04月30日：0%异常率 05月10日：0%异常率 04月02日：0%异常率	无异常



1、赵建春-AI浪潮下的高效运维思考与实践.pdf

<https://pan.baidu.com/s/1jIDwZTO#list/path=%2F&parentPath=%2F>

GBDT“失败”总结



原因分析

- 样本不全面，11/8000
- 正负样本不均衡，比例为10000 : 1

结论

- 监控行业负样本稀疏
- 负样本获取困难

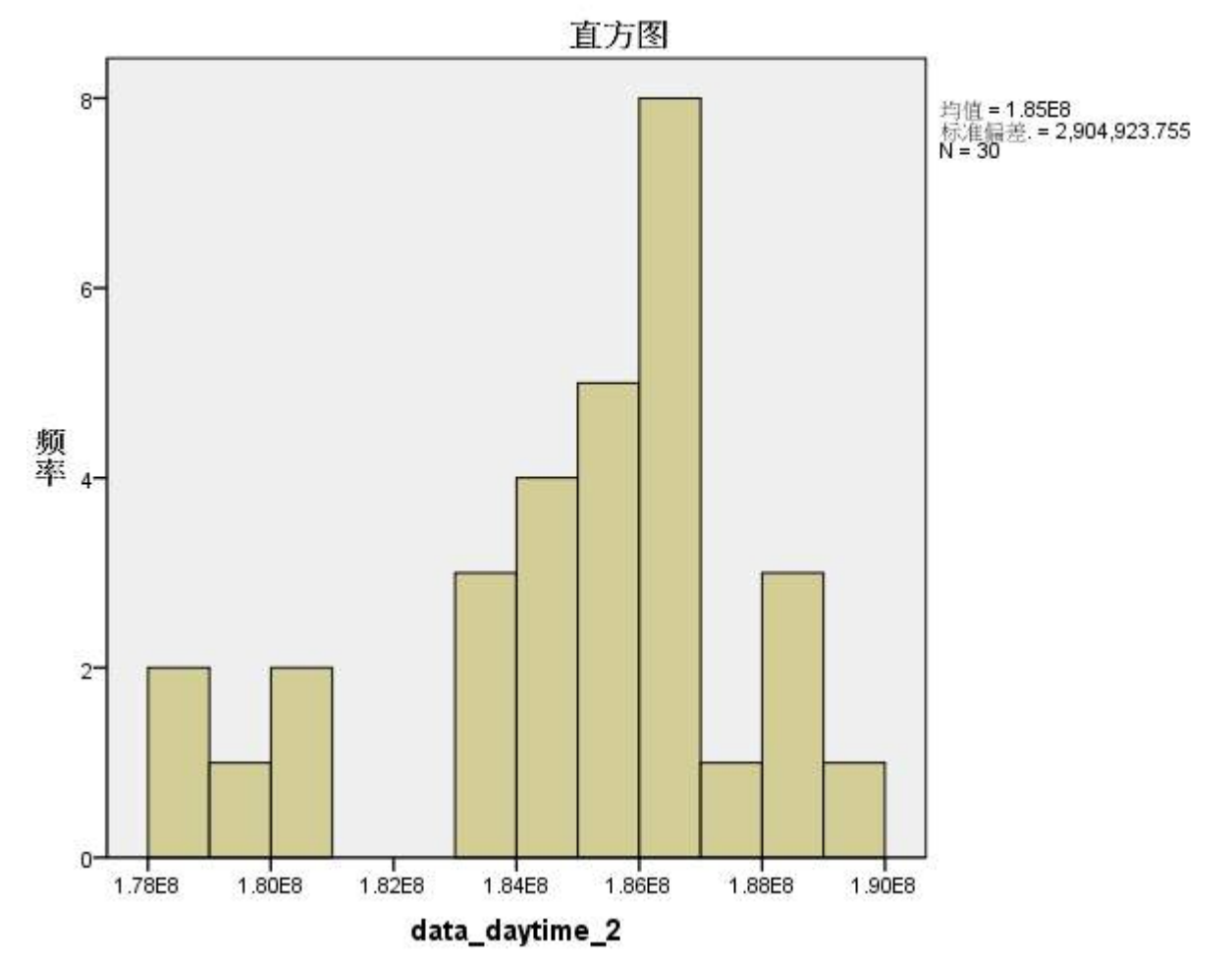
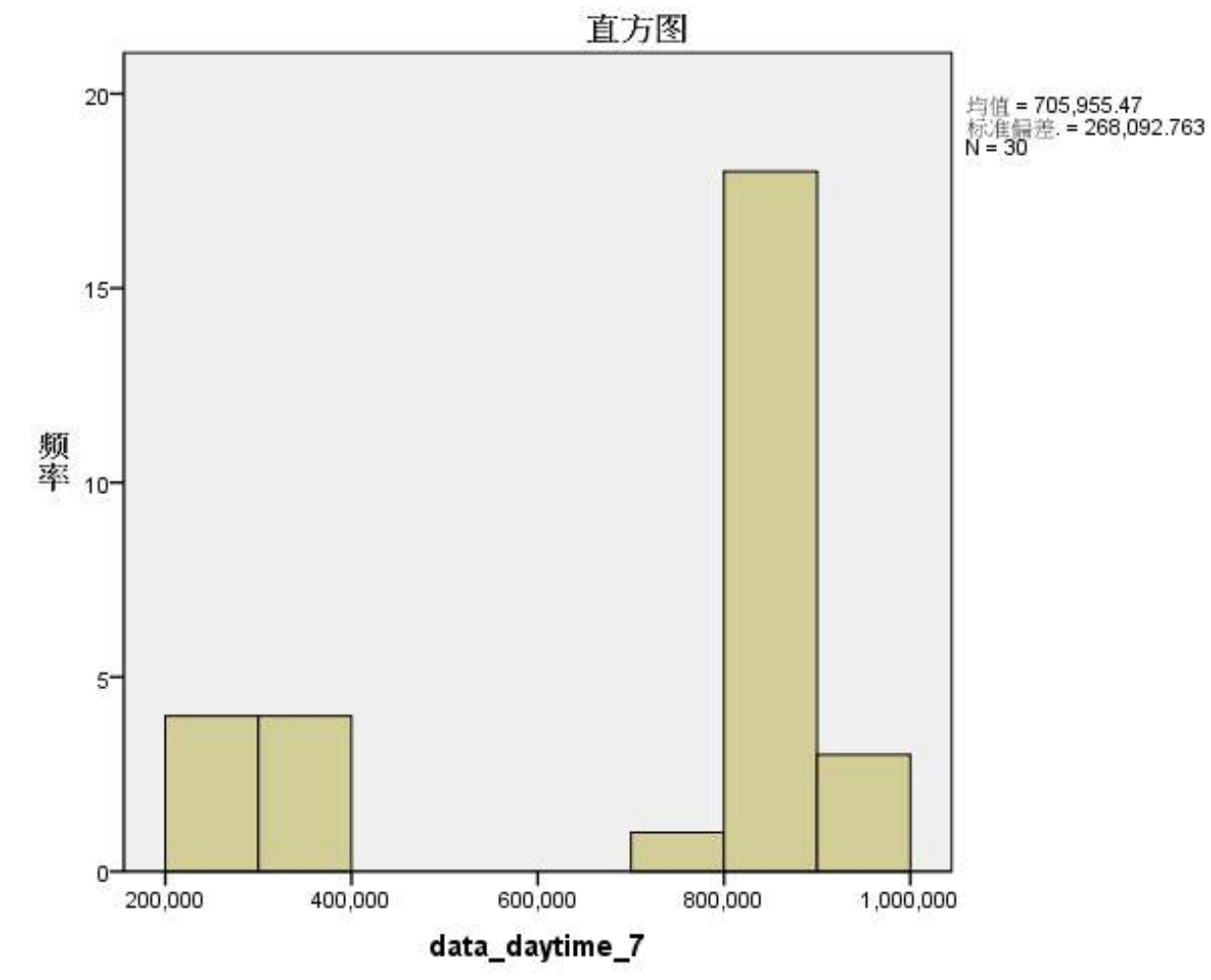
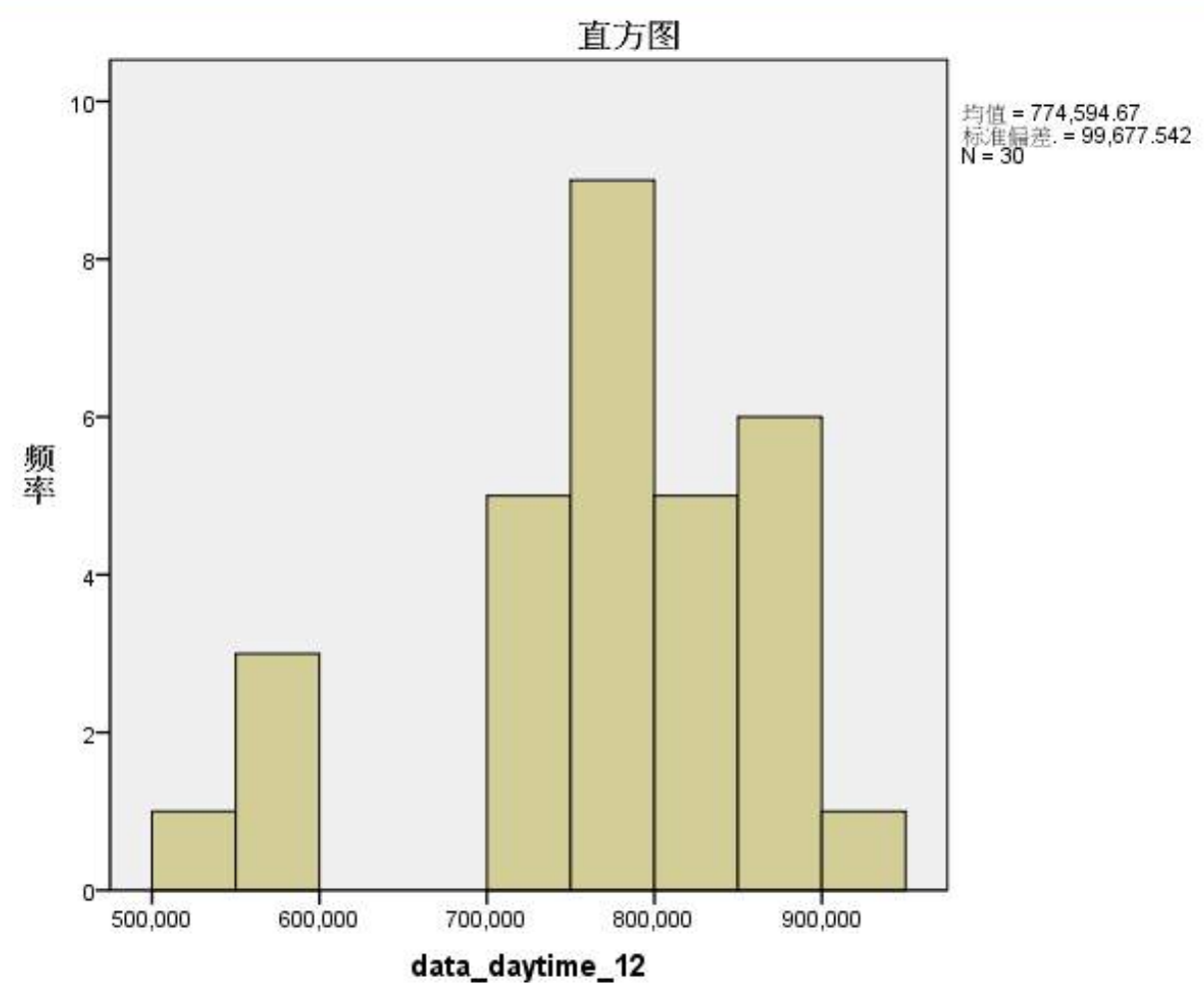
解决方案

- ✓ 过滤掉大量正常样本
 1. 统计判别法
 2. 无监督

统计判别法

准则	显著性检测	公式	优缺点
Pauta	正态分布	$ x_i - u > 3 * \sigma$	使用简单，样本较少时误差比较大
Chauvenet	正态分布	$ x_i - u > Z_c * \sigma$	$Z_c < 3$ 且与n相关，比Pauta更加准确，可用于n < 10的粗大判定
Grubbs	正态或接近正态	$ x_i - u > T * \sigma$	T与n和概率 α 有关，判断标准更加严格，在n=20-100时，效果较好
Dixon	不需要	相对复杂	应用于同组数据的一致性检验，但计算相对复杂
T-Test	不需要	$ x_i - u > K * \sigma$ 或者 $ x_i - u > K * \sigma$	与Dixon类似

正态性检测



定性结果：同一时刻的上报值，长时间窗口看趋于正态分布

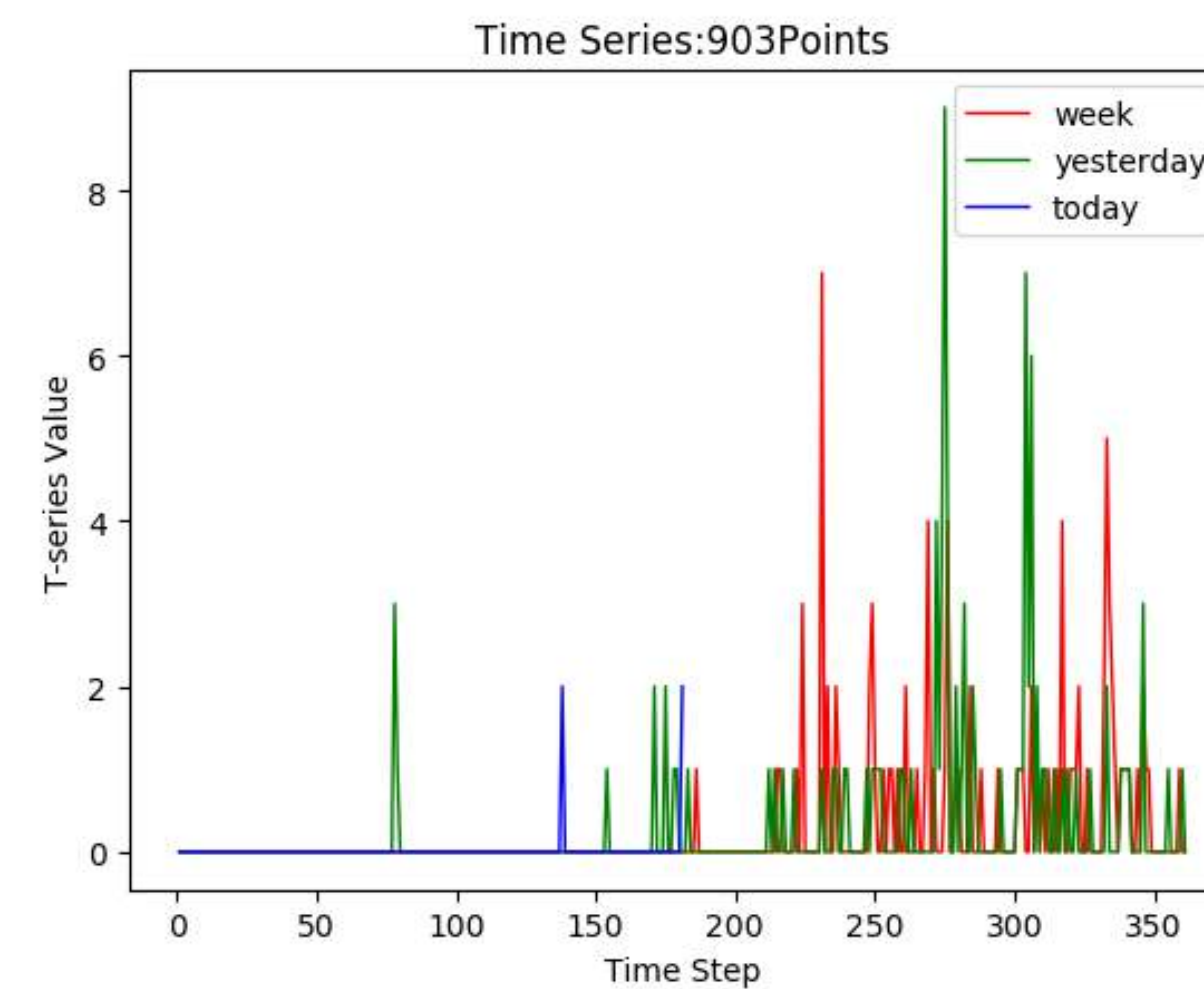
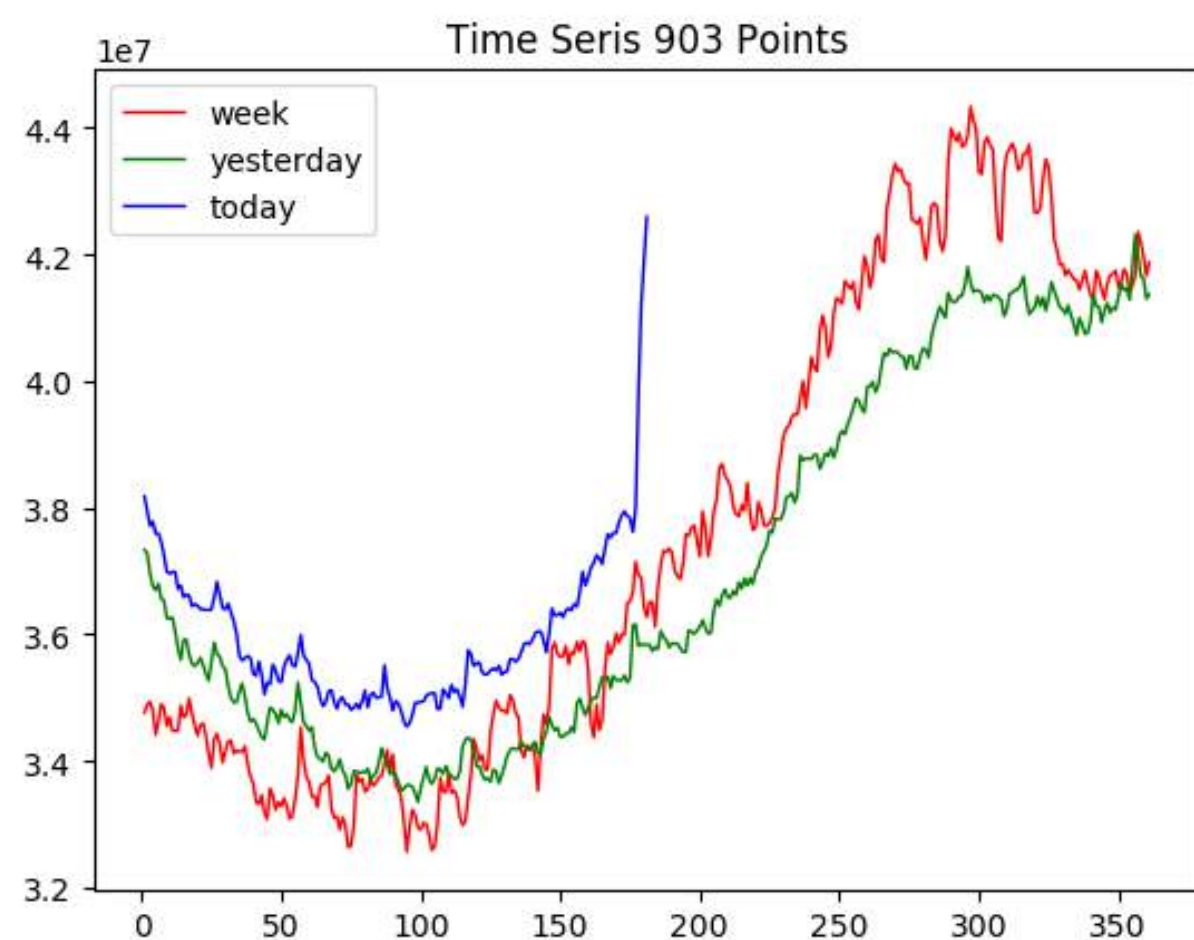
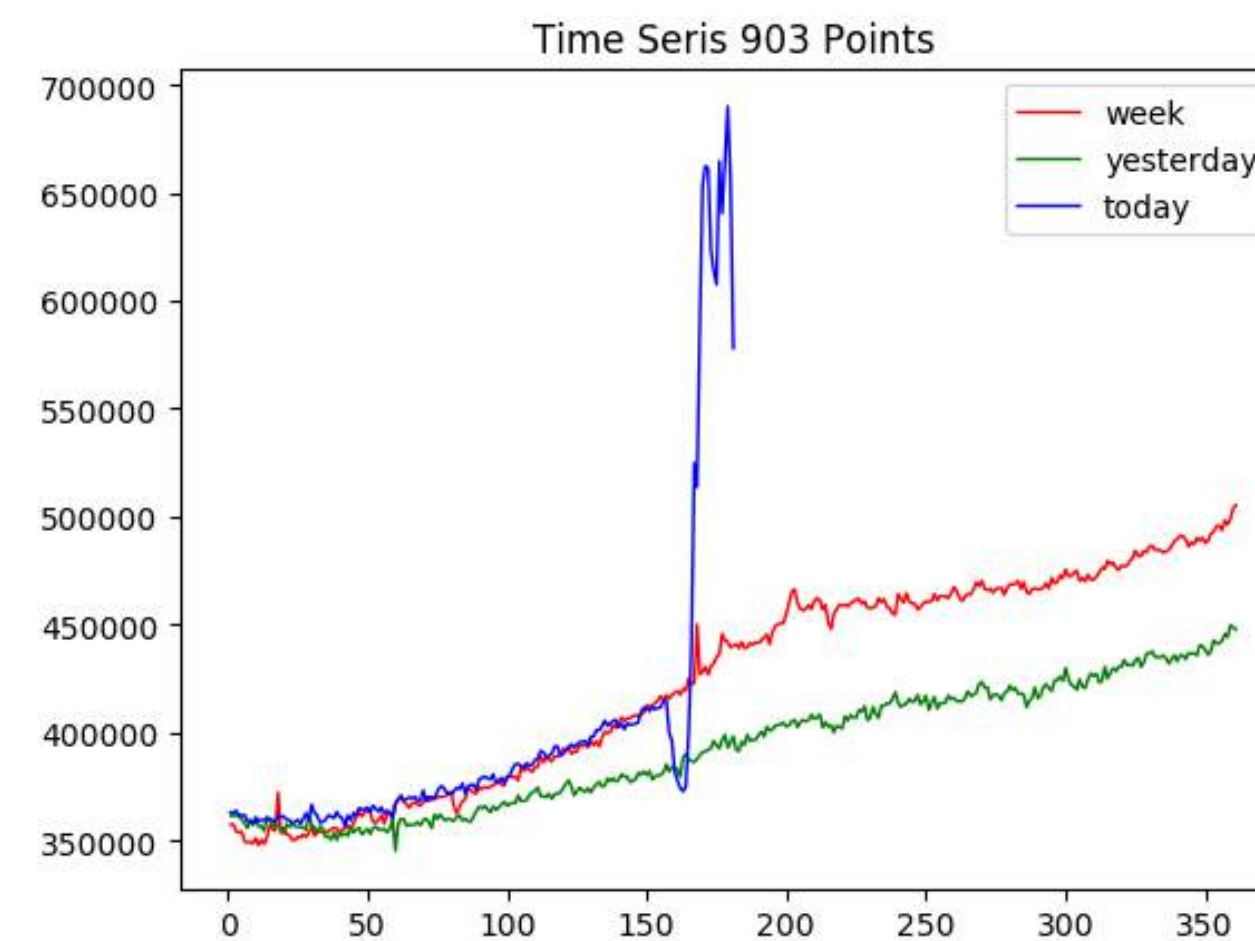
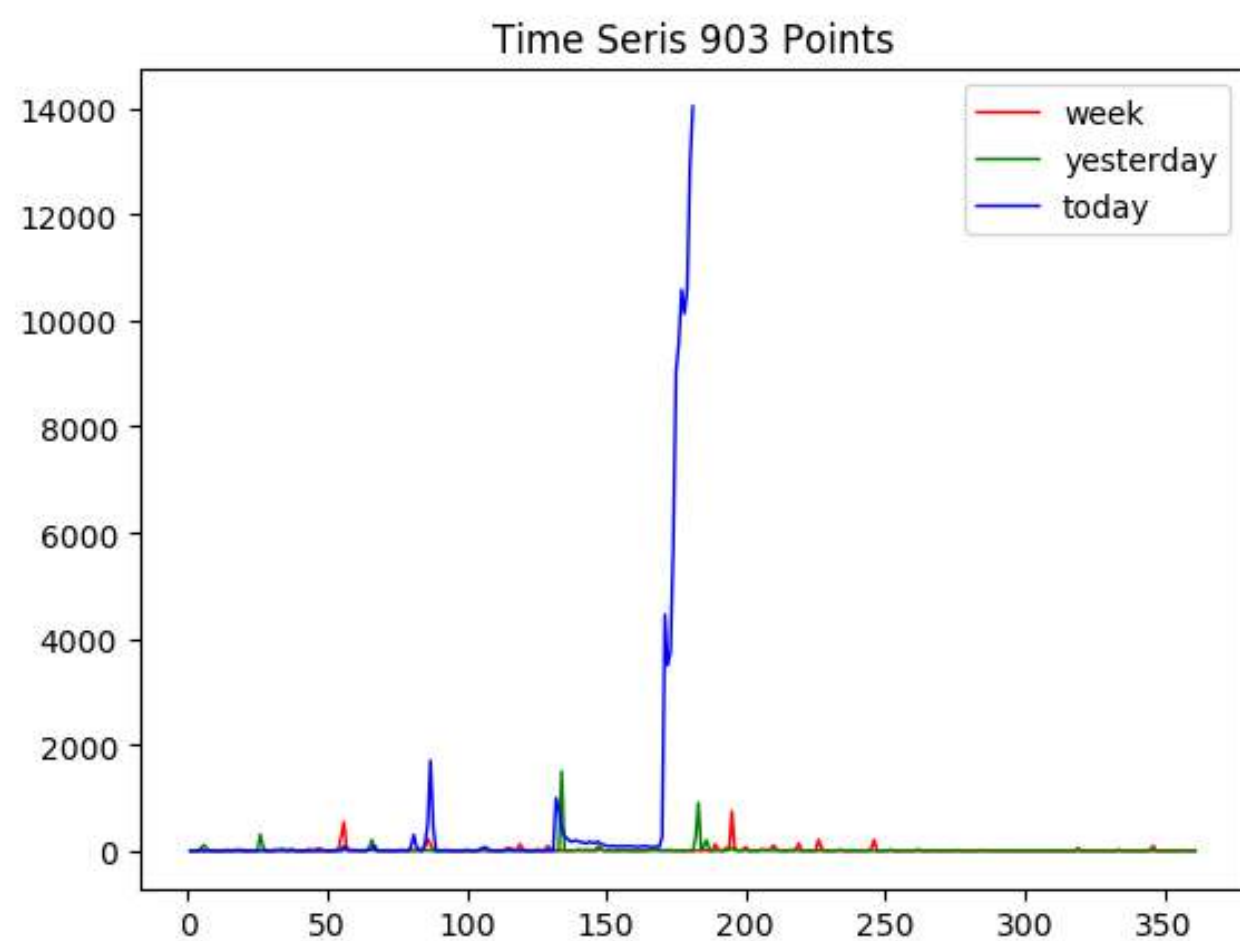
统计判别法： Grubbs

```
def calculate_statistic(self, value_list):  
    return list(pandas.DataFrame(value_list).describe()[0].values[1:])  
  
def grubbs_predict(self, check_value, ref_value_list):  
    t_index = self.index  
    ref_desc = self.calculate_statistic(ref_value_list)  
    if abs(check_value - ref_desc[0]) > t_index * ref_desc[1]:  
        return -1  
    else:  
        return 1
```

算法

1. 值：当前时刻点
2. 参考值：前后N分钟的环比、同比、周同比值
3. 均值、标准差
4. $|x_i - u|$ 与 $Z * \sigma$ 比大小
5. 异常： $|x_i - u| > Z * \sigma$

Grubbs算法结果



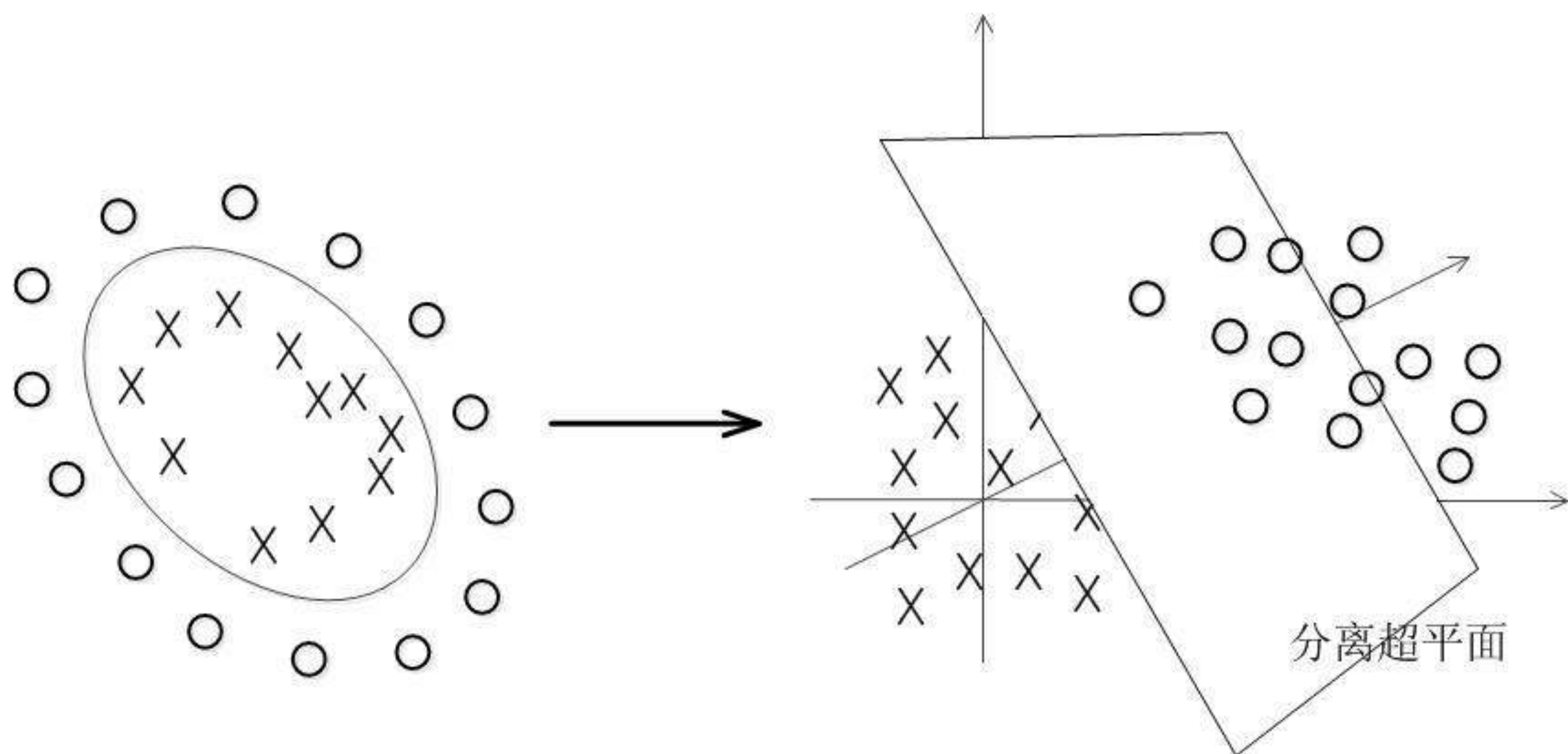
Grubbs分析优化

第一轮	总样本	3h_3	1h_3	3h_or_1h_3
正样本	19	17	17	17
负样本	208	153	75	158
准确率		74.9%	40.5%	77.1%
召回率		73.6%	36.1%	76.0%
8000 正样本误召回率		0.0%	NA	0%

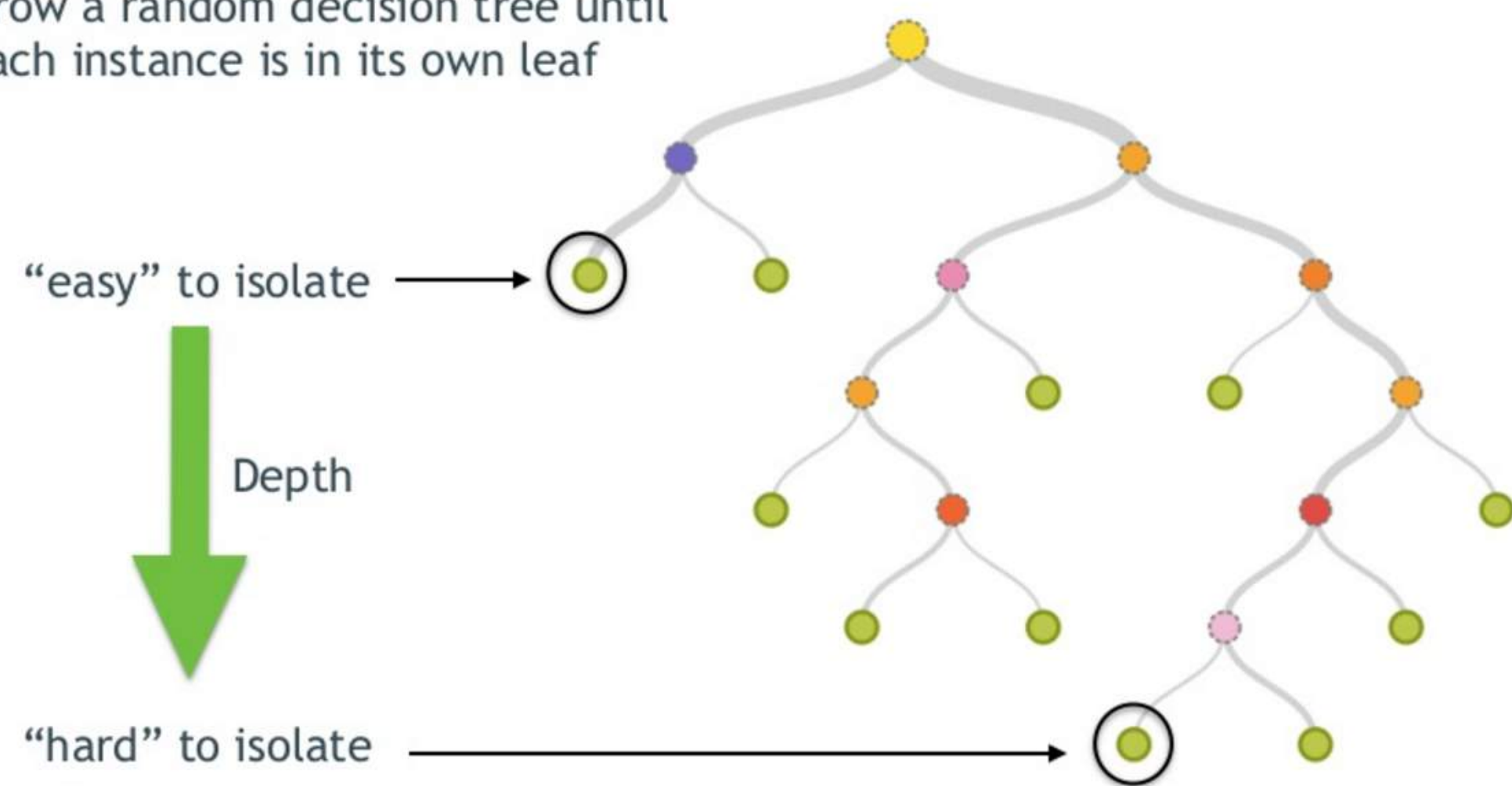
第二轮	总样本	1h_2	1h_2.5	1h_3	1h_3.5	3h_2	3h_2.5	3h_3	3h_3.5	1h_or_3h_2	耗时
正样本											4ms/个
负样本	1328	991	725	542	465	1122	1038	889	690	1147	
准确率											
召回率		74.6%	54.6%	40.8%	35.0%	84.5%	78.2%	66.9%	52.0%	86.4%	
8000 正样本误召回率		NA	NA	NA	NA	2.0%	0.4%	0.0%	0.0%	<2.0%	

优化点：1, $|x_i - u|$ 与 $Z * \sigma$, Z调整；2, Window大小

无监督算法



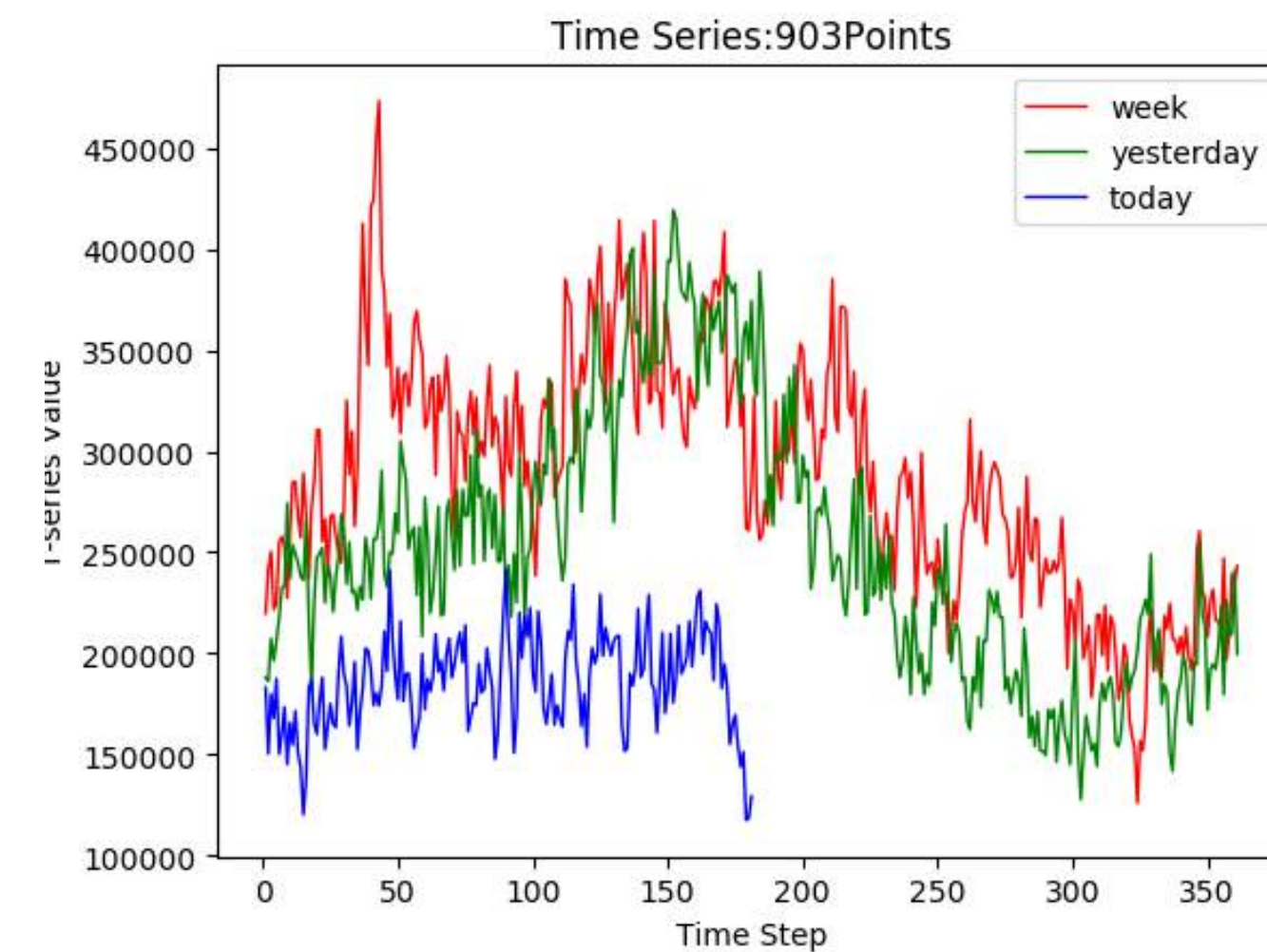
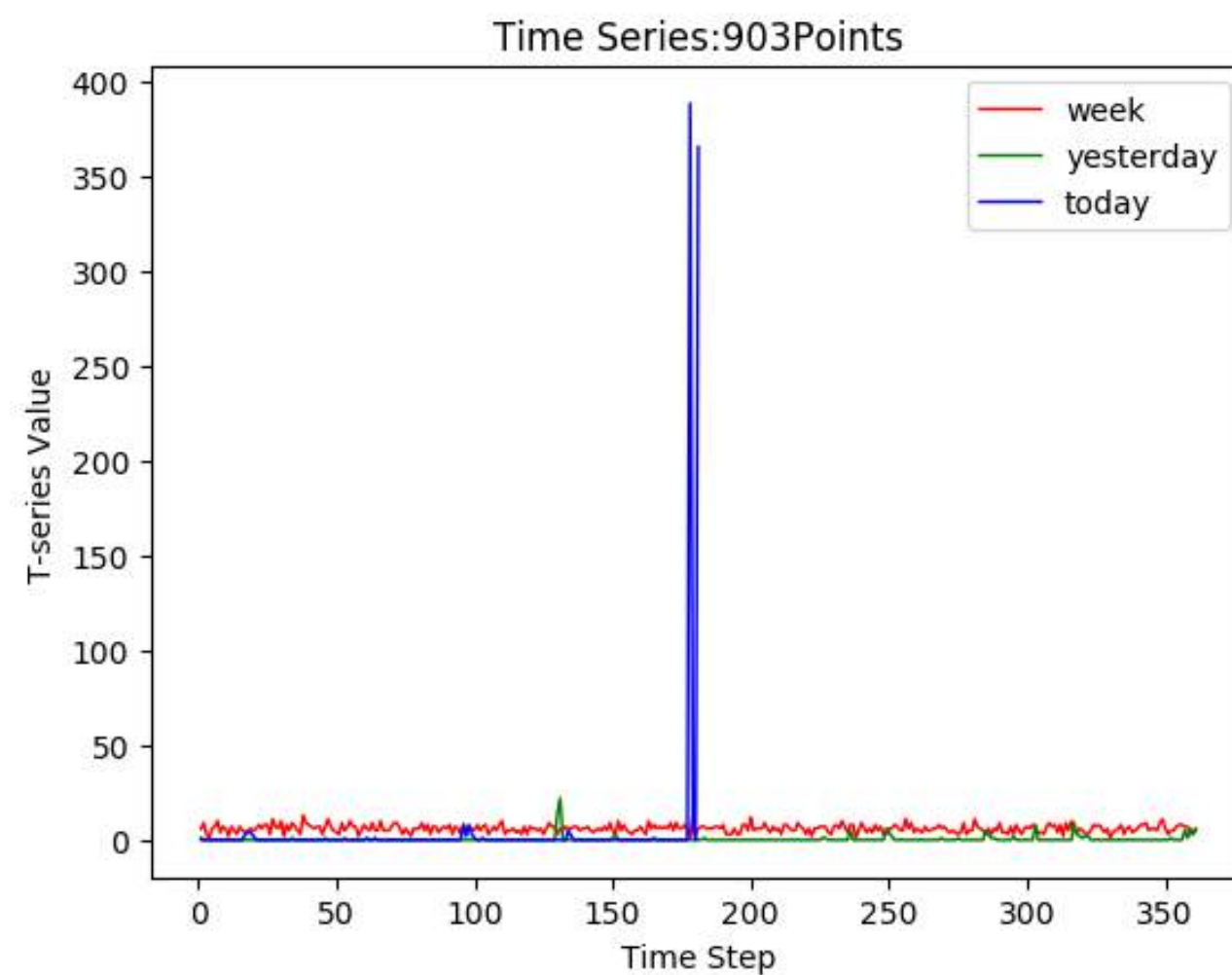
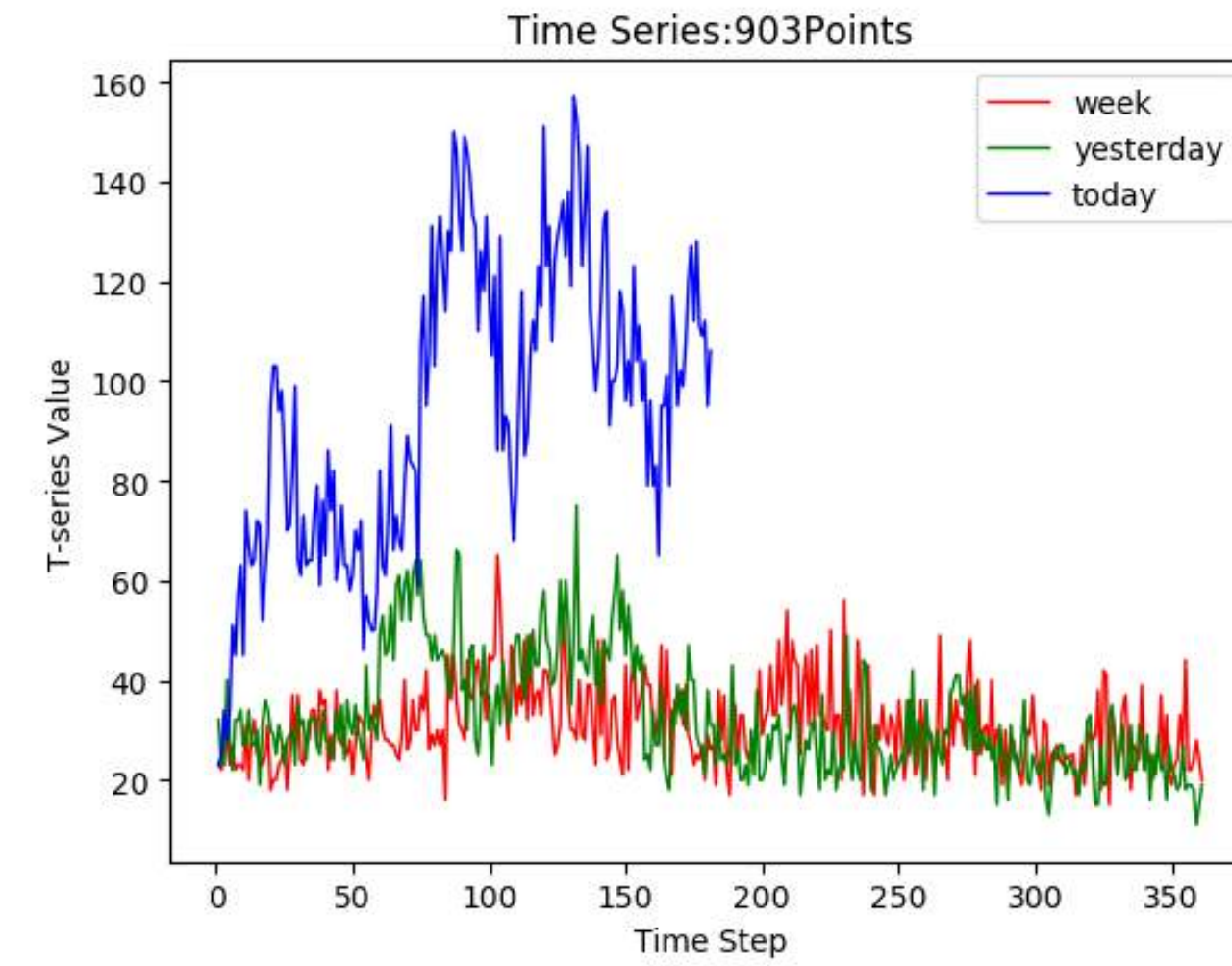
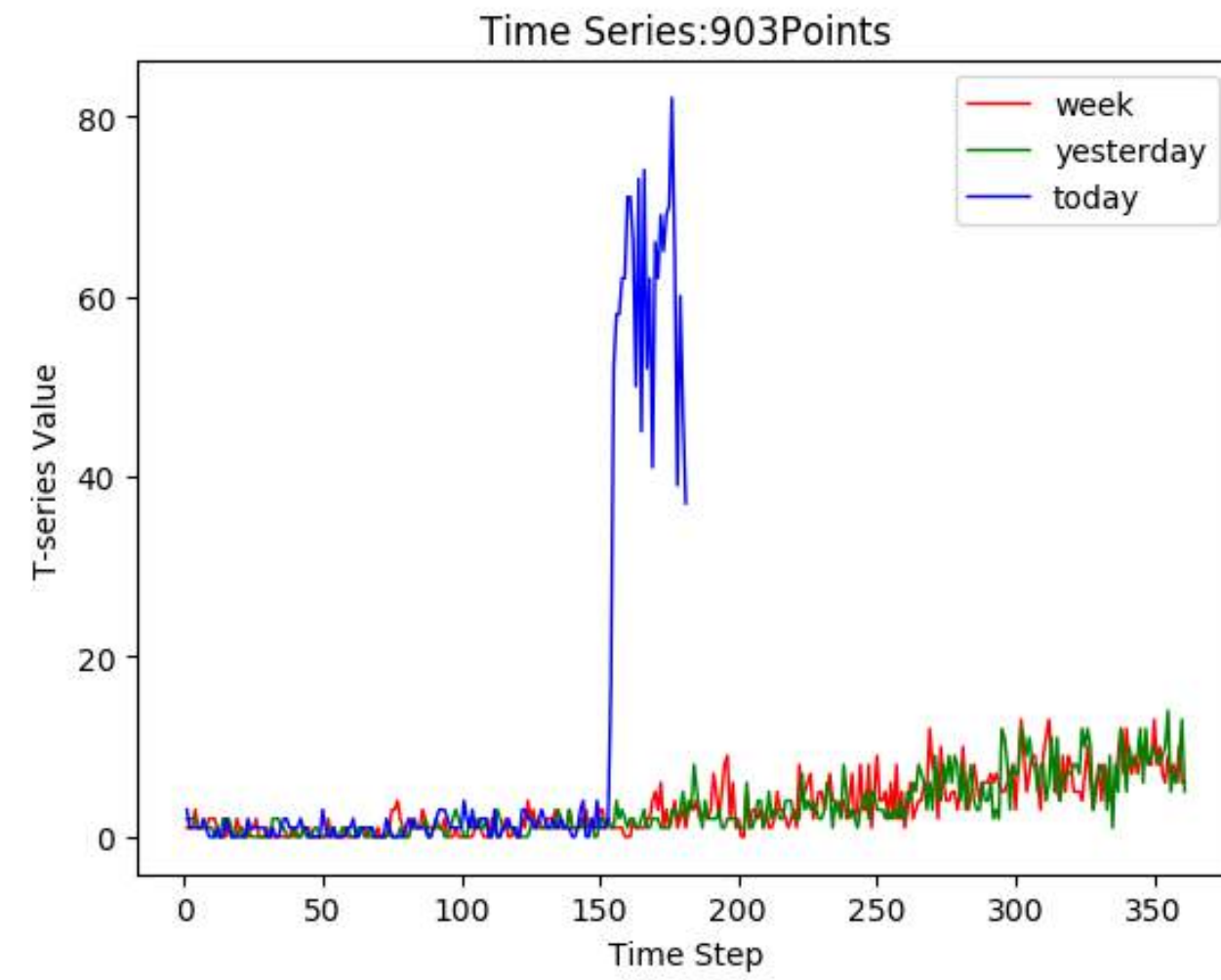
Grow a random decision tree until
each instance is in its own leaf



无监督算法: Isolation Forest

```
def if_predict(self, sample_features):  
    clf = IsolationForest(n_estimators=3, max_samples='auto', contamination=0.15)  
    # clf = IsolationForest()  
    clf.fit(sample_features)  
    y_pred_test = clf.predict(sample_features)  
    # print y result  
    return y_pred_test[-1]
```

Isolation Forest 算法结果



Isolation Forest分析优化

第二轮	总样本	1h_2	1h_2.5	1h_3	1h_3.5	3h_2	3h_2.5	3h_3	3h_3.5	1h_or_3h_2	耗时
正样本	8000										
负样本	1328	991	725	542	465	1122	1038	889	690	1147	
准确率											
召回率		74.6%	54.6%	40.8%	35.0%	84.5%	78.2%	66.9%	52.0%	86.4%	4ms/个
8000 正样本误召回率		NA	NA	NA	NA	2.0%	0.4%	0.0%	0.0%	<2.0%	

第二轮	总样本	Default	n_estimators=3 max_samples='auto' contamination=0.15
正样本	8000		
负样本	1328	1287	1192
准确率			
召回率		96.9%	89.8%
8000 正样本误召回率		5%	11.6%
耗时		243ms/个	10ms/个

优化：参数调整，n_estimators=3, max_samples='auto', contamination=0.15

Grubbs & Isolation Forest

	总样本	3h_2	3h_3	1h_or_3h_2	n_estimators=3 max_samples='auto' contamination=0.15	3h_2 or n_estimators=3 max_samples='auto' contamination=0.15
正样本	8000					
负样本	1328	1122	889	1147	1192	1252
准确率						
召回率		84.5%	66.9%	86.4%	89.8%	95%
8000 正样本误召回率		2.0%	0.0%	<2.0%	11.6%	7%
耗时		4ms/个	4ms/个	4ms/个	10ms/个	

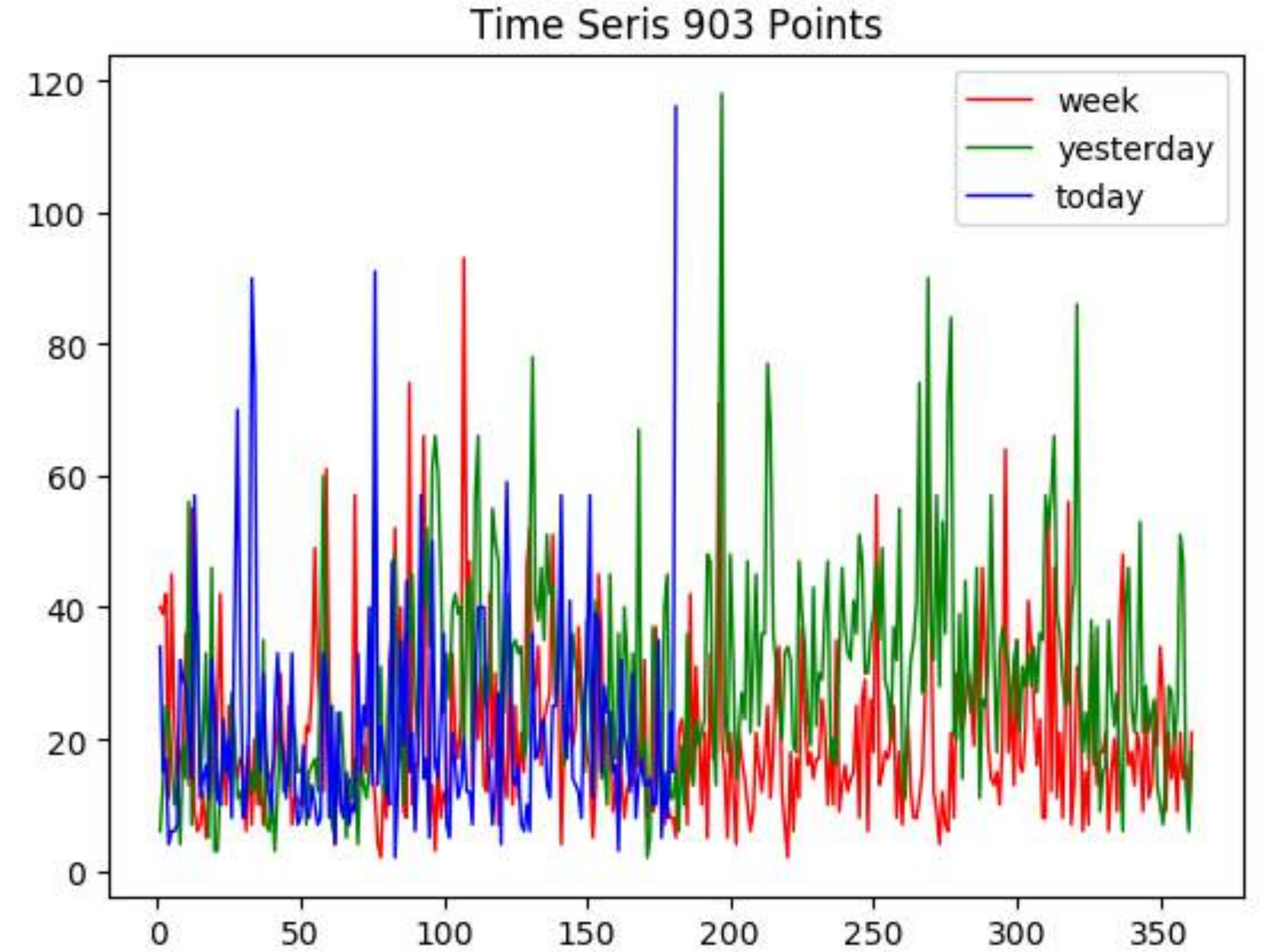
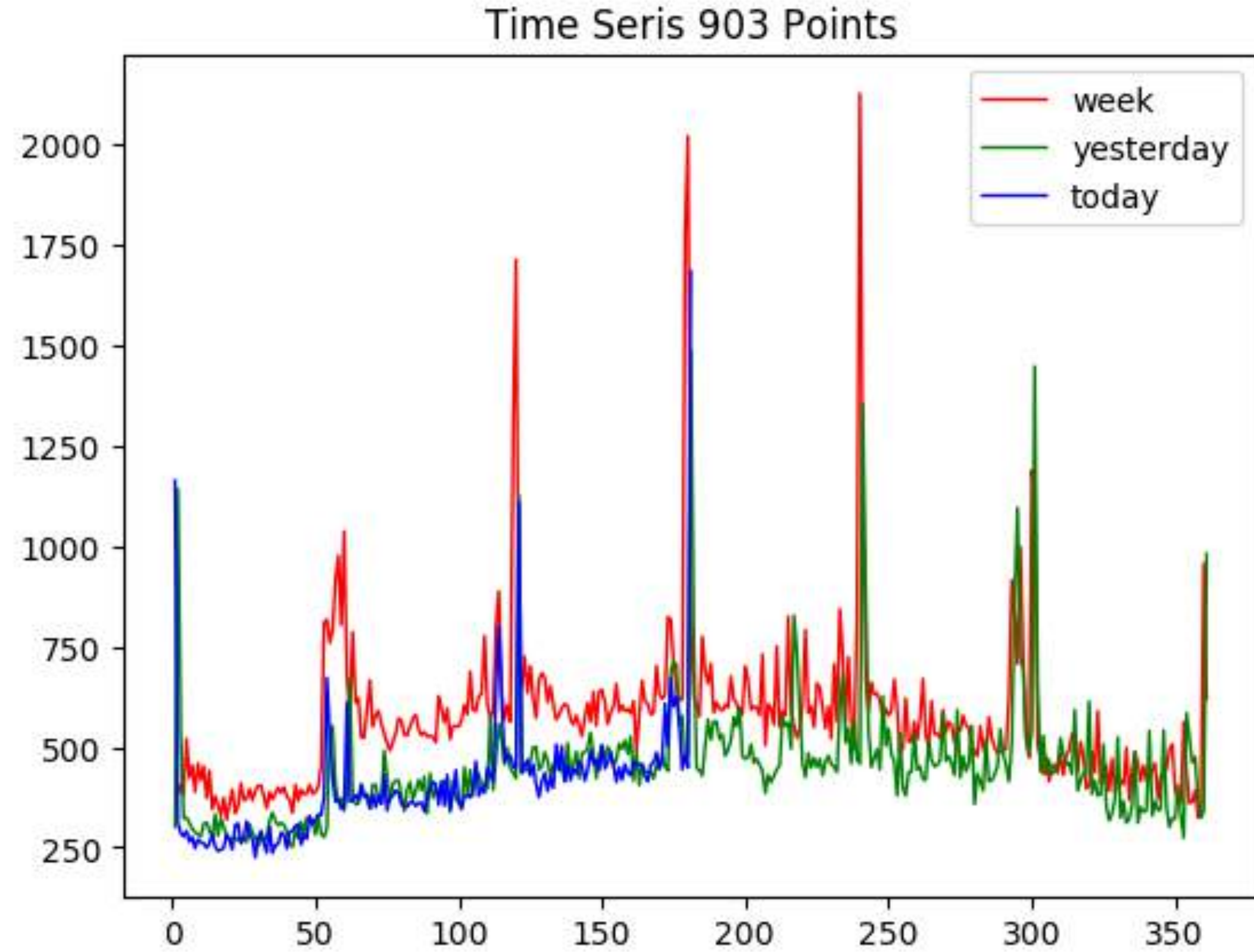
优化：算法整合

有监督算法:

```
def gbd_train(self, train):  
    # Get train/test set random  
    sample_random = random.sample(train, len(train))  
    trainset = np.array(sample_random[0:int(len(sample_random)*0.9)])  
    testset = np.array(sample_random[int(len(sample_random)*0.9):])  
    y = trainset[:, -1]  
    x = trainset[:, (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)]  
    y_test = testset[:, -1]  
    x_test = testset[:, (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)]  
    # Training  
    grd = GradientBoostingClassifier(n_estimators=300, max_depth=10, learning_rate=0.05)  
    grd.fit(x, y)  
    preds = grd.predict_proba(x_test)[:, 1]  
    gbd_auc = roc_auc_score(y_test, preds)  
    model_name = "gbd_model" + "_" + time.strftime('%Y-%m-%d', time.localtime(time.time()))  
    pickle.dump(grd, open(model_name, "wb"))  
    print "AUC:", gbd_auc  
        标记为负样本
```

```
def gbd_predict(self, x_test):  
    testset = np.array([x_test]) #注意此处中括号, 默认样本位多样本数据  
    x_test = testset[:, (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31)]  
    loaded_model = pickle.load(open("./gbd_model_2018-01-03", "rb"))  
    preds_new = loaded_model.predict_proba(x_test)[:, 1]  
    return preds_new[0]
```


GBDT解决的问题



GBDT分析优化

第一轮	总样本	GBDT
正样本	404	323
负样本	96	64
召回率	NA	66.6%
正确率	NA	77.4%

第二轮	总样本	GBDT	GBDT_OPT
正样本	2029	1980	1991
负样本	5464	5350	5410
召回率	NA	98%	99%
正确率	NA	97%	98%

优化：1, 新增样本；2, 新增特征

TABLE OF CONTENTES

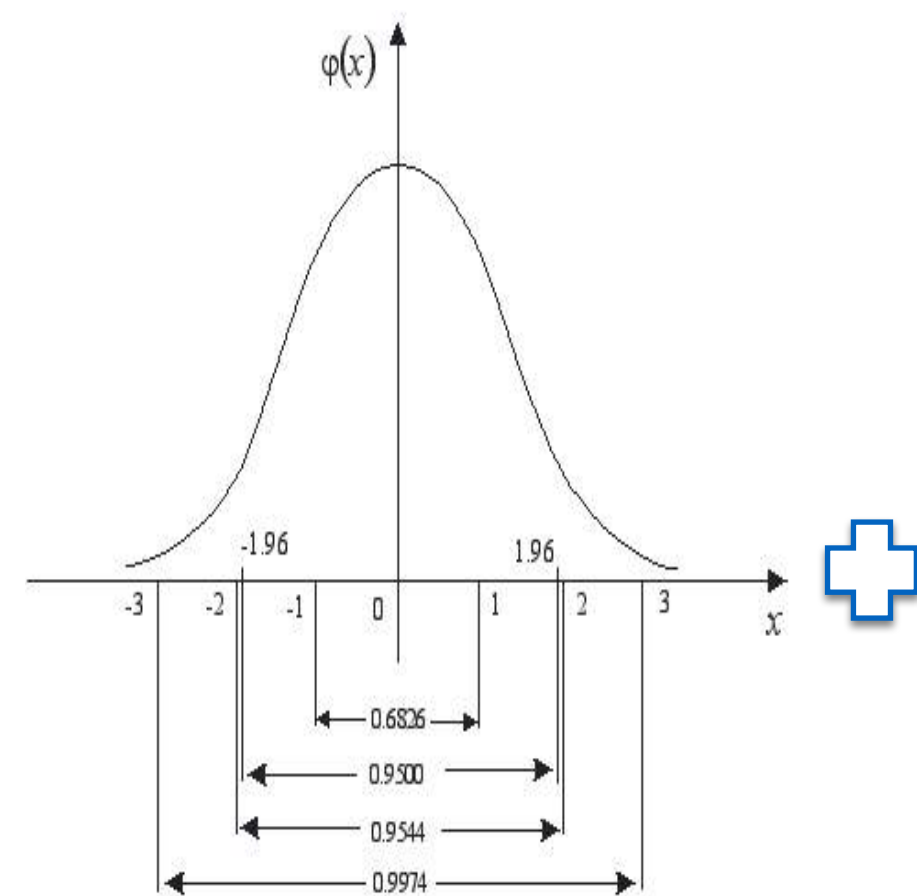
背景

算法探索

落地

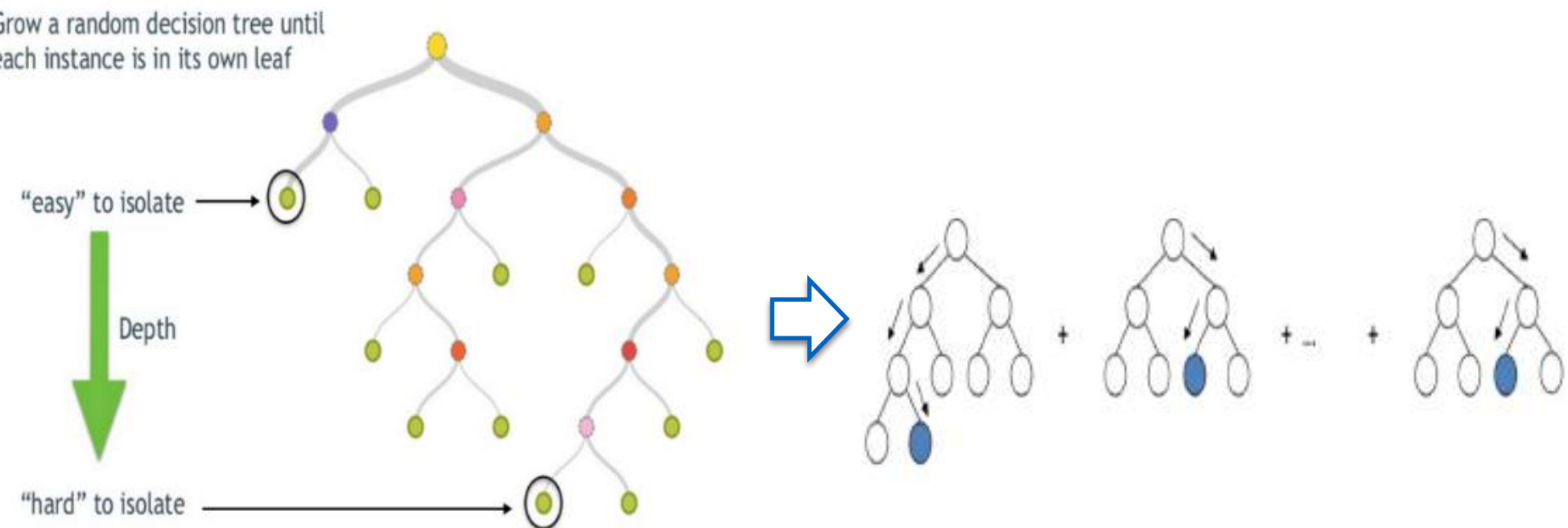
下一步

算法方案



● Grubbs

Grow a random decision tree until each instance is in its own leaf



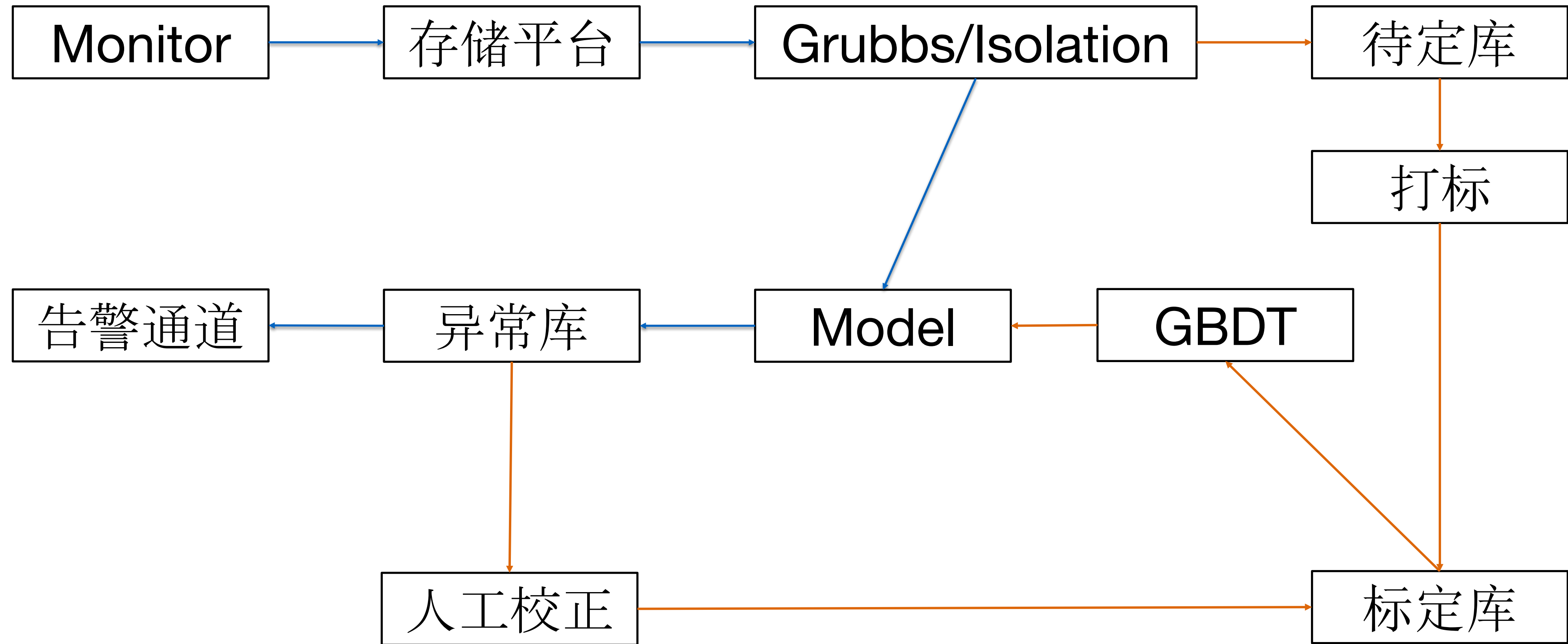
● Isolation Forest

● GBDT

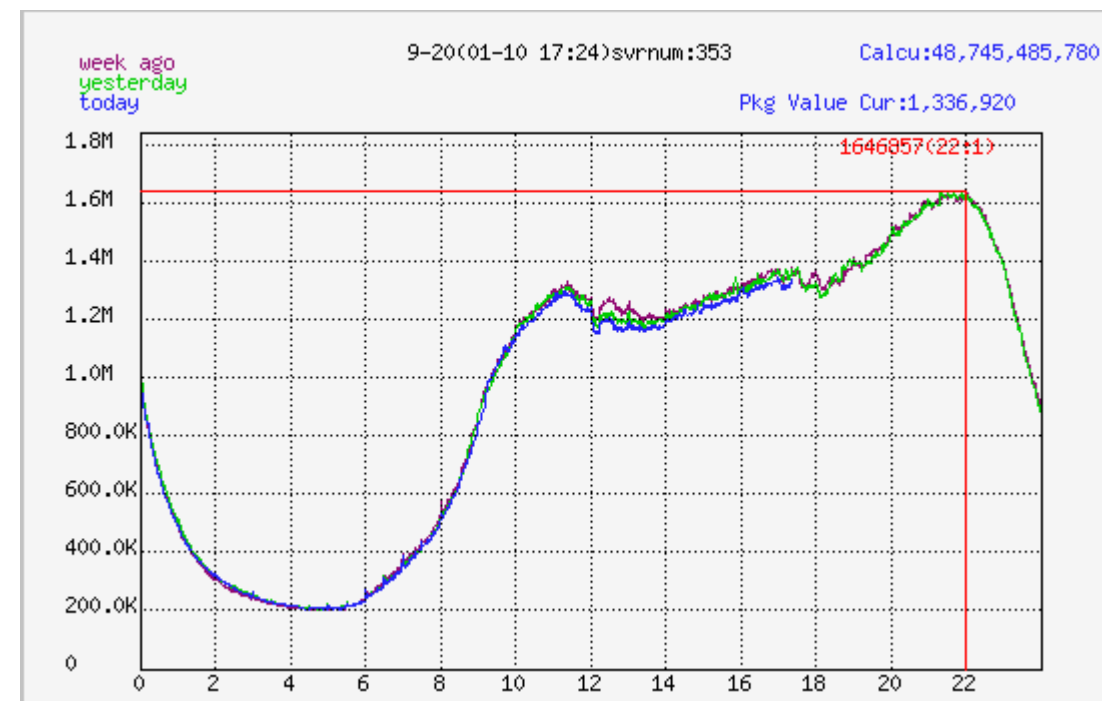


● 逻辑策略

工程方案



数据与计算“坑”

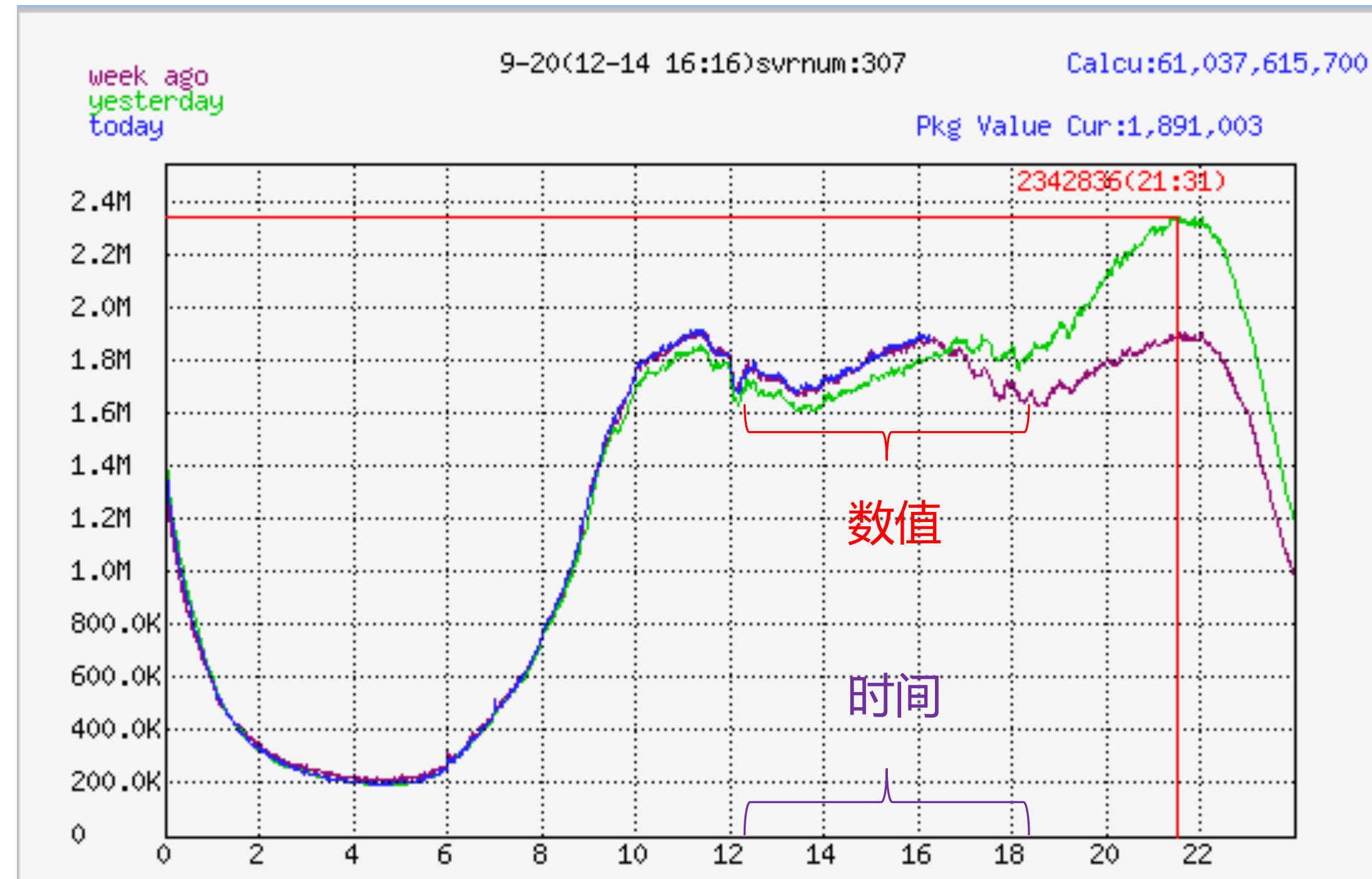


- 1.数据缺失
- 2.“零”值
- 3.数据变更



- 1.计算延时
- 2.“雪崩”

特征“坑”



时间与值对应

系统-展示

异常视图

异常查询

异常统计

特别关注

全部异常

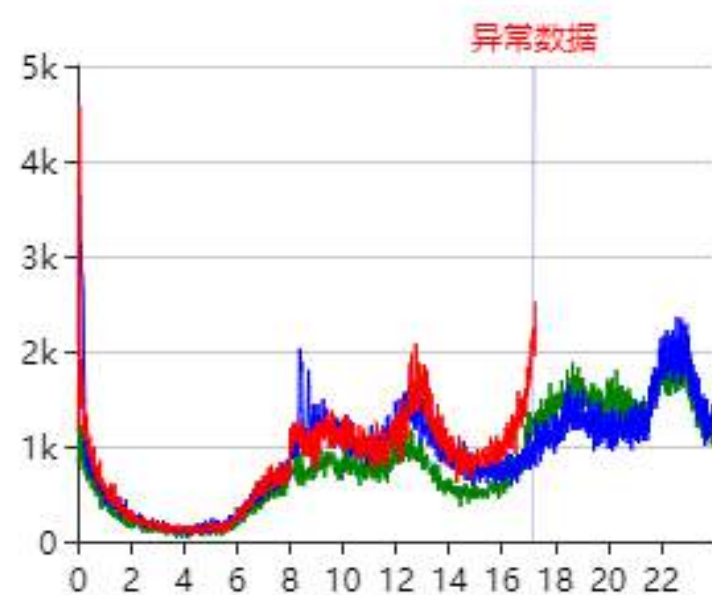
高级搜索

视图名/ID



[4542] (新.登录类CS) 主从DB_sh - [428593]AppName长度...
2017-12-29 17:10:10 - 2017-12-29 17:15:11

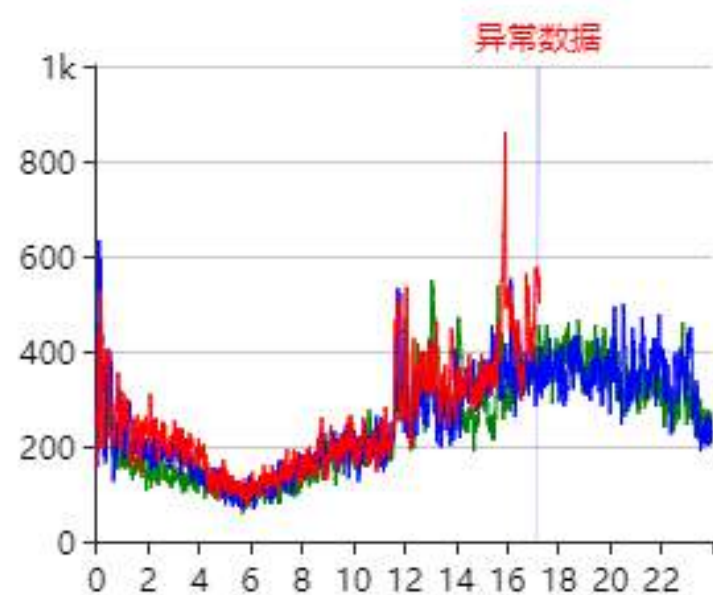
今日 昨日 一周前



标记为正样本
标记为负样本

[2433]c2_conn新 - [88271]AppID 0x200c01(XGame)
2017-12-29 17:10:03 - 2017-12-29 17:15:03

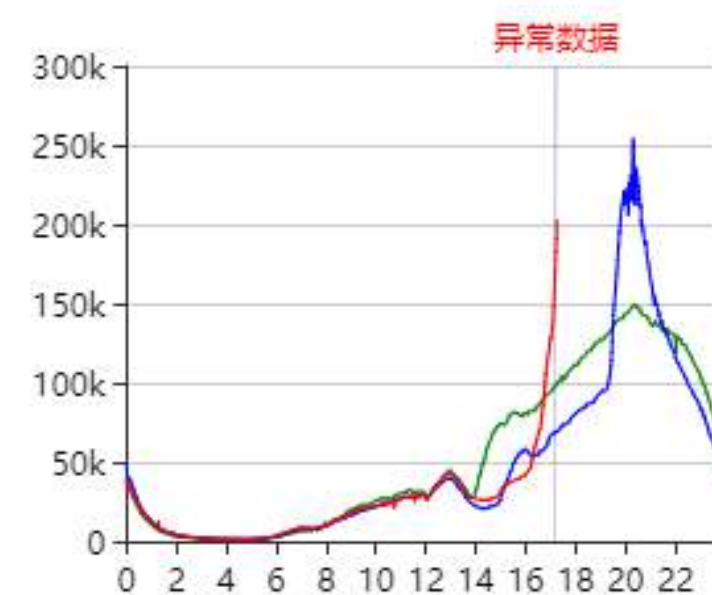
今日 昨日 一周前



标记为正样本
标记为负样本

[16539]机器人消息预处理 - [2607980]oidb_0xeed小冰回包...
2017-12-29 17:10:30 - 2017-12-29 17:15:29

今日 昨日 一周前



标记为正样本
标记为负样本

[4428]ptlogin_websvr_天津 - [13077]post登录请求 (所有输...
2017-12-29 17:10:08 - 2017-12-29 17:15:07

今日 昨日 一周前



标记为正样本
标记为负样本

系统-接入

AIM

异常视图

推送配置

关于AIM

推送配置

视图订阅

属性屏蔽

新增订阅

1 配置群信息

2 选择模块/视图

订阅名:

推送群号:

当前用户QQ号为 1064375898! 请确保该QQ在该群中为管理员! 如需更改请点击[此处](#)

负责人:

系统-输出



智能检测

【智能告警】视图:【udpconn_天津移动】, 属性:【sched--老重定向逻辑忽视ISP要求返回本IDC】在时间点: 2018-01-09 09:06发生异常。

<http://monitor.server.com/link/graph/viewid:5654/pointer:282415/compare>

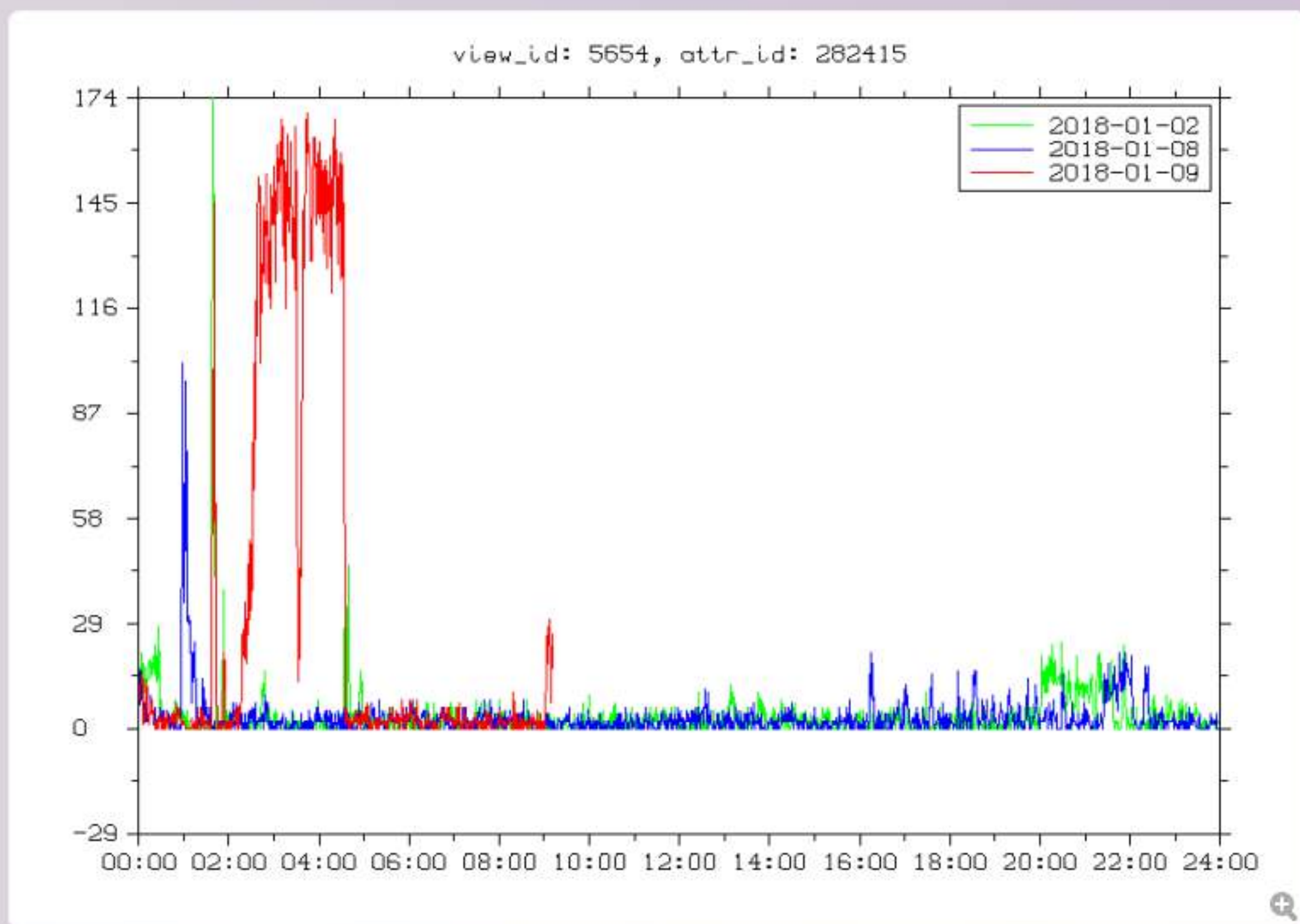


TABLE OF CONTENTES

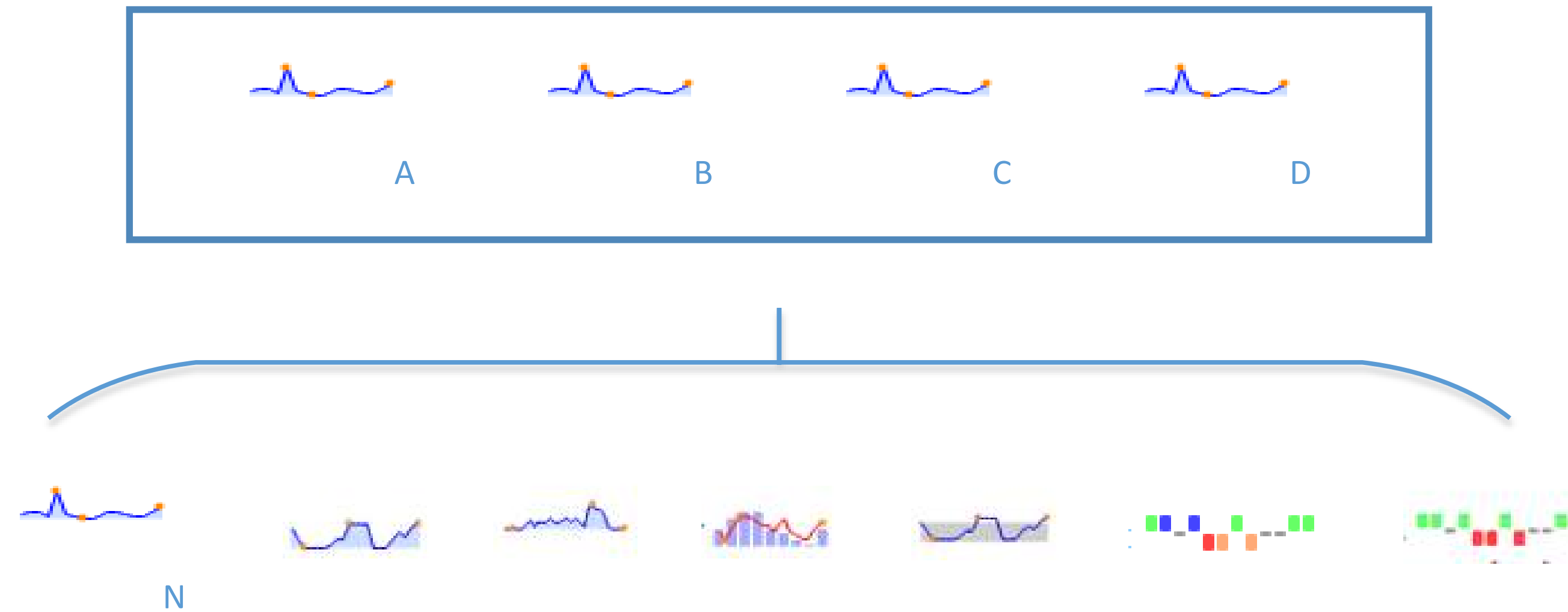
背景

算法探索

落地

下一步

下一步：关联分析和根因分析



关联分析

1. 时间关联
2. 空间关联
3. 逻辑关联

根因分析

1. 时间、空间与逻辑关联的因、果分析
2. “自愈”

Thanks!