

异常检测在苏宁的实践

苏宁科技集团云计算研发中心AIOps研发中心
汤永

1. 背景介绍

BACKGROUND INFORMATION

2. 监控开放平台设计

THE DESIGNATION OF MONITOR OPEN PLATFORM

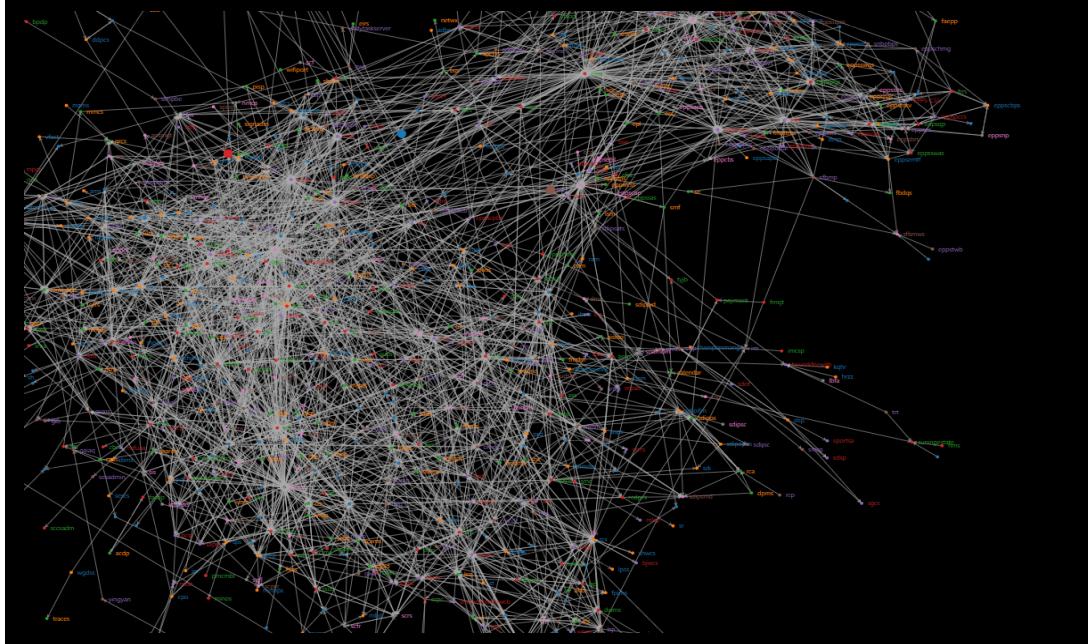
3. 异常检测平台实践

THE PRACTICE OF ANOMALY DETECTION PLATFORM

4. 未来规划

THE FUTURE PLANNING

1. 背景介绍



■ 系统和服务的复杂性：

1. 4000+系统，数量还在增加
2. 系统间调用方式复杂：大部分使用ASF，也有其他的方式如HESSIAN，ESB等
3. 苏宁业务的复杂：既有线上新业务又有线下老业务，这些业务系统之间会有大量的关联。

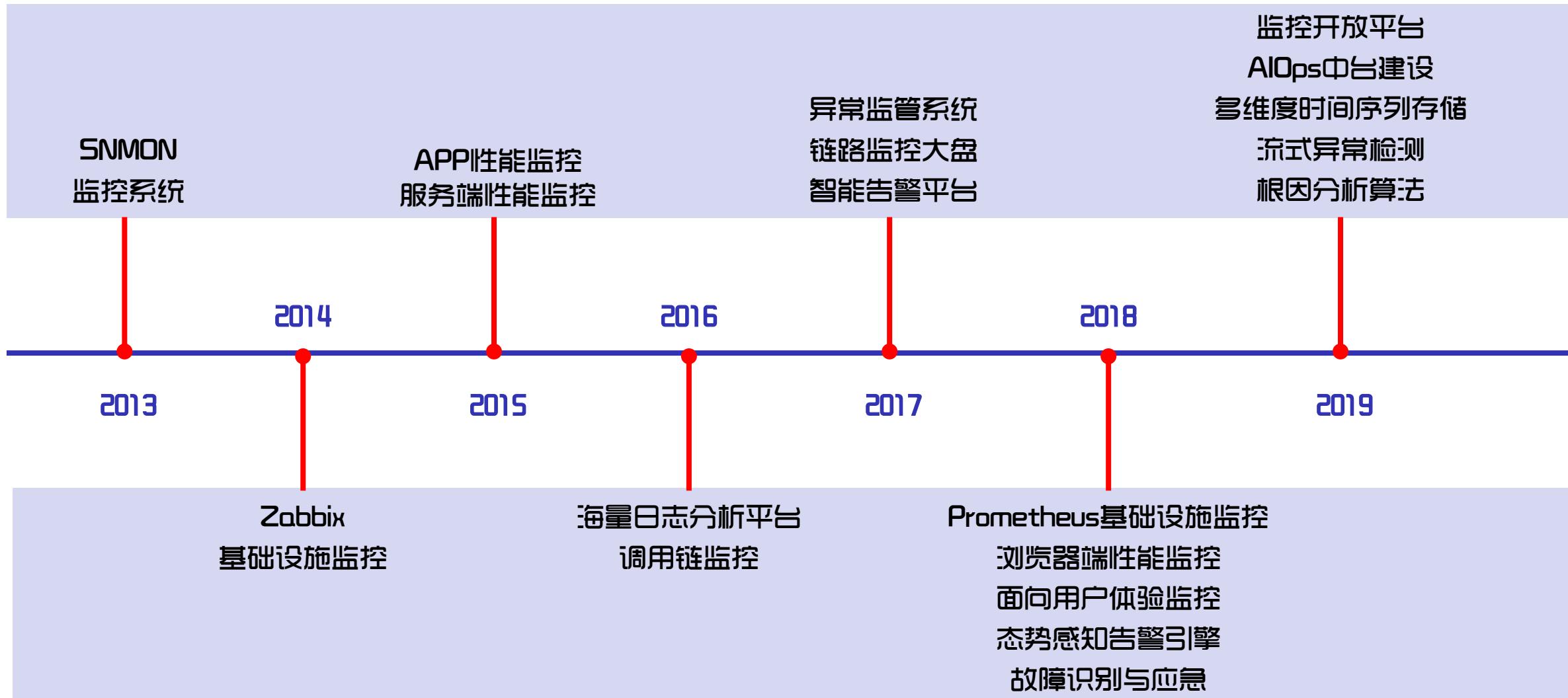


■ 基础环境的复杂性：

1. 多数据中心，每个数据中心会划分多个逻辑机房和部署环境
2. 服务器规模27w+，例如，缓存服务器就有可能有上千台服务器
3. 服务器类型复杂性：cloudstack, openstack, vmware, kvm, k8s下docker, swarm下的docker。

1. 背景介绍: 监控发展历程

数据 + 分析 + 算法 = AIOps



1. 背景介绍

BACKGROUND INFORMATION

2. 监控开放平台设计

THE DESIGNATION OF MONITOR OPEN PLATFORM

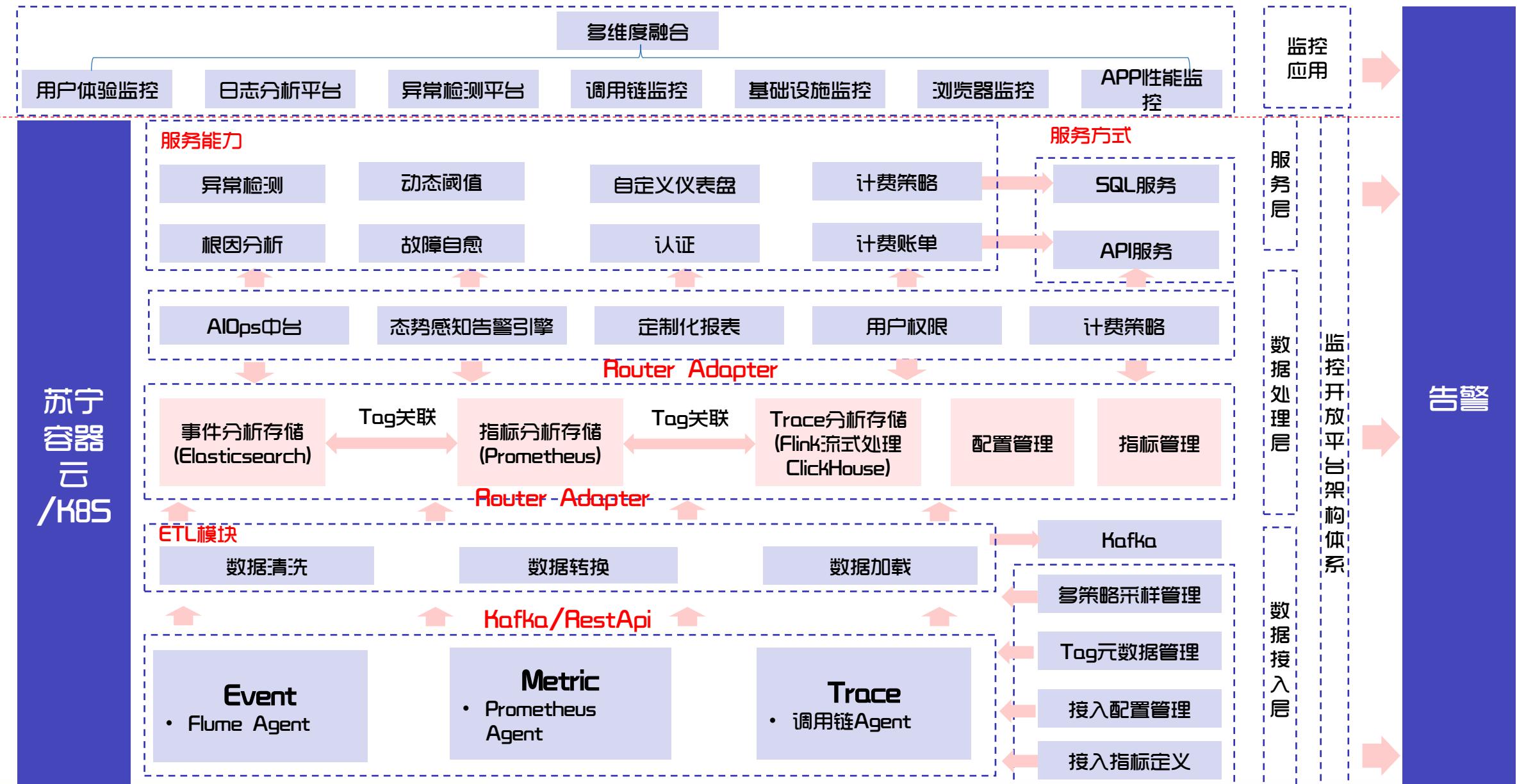
3. 异常检测平台实践

THE PRACTICE OF ANOMALY DETECTION PLATFORM

4. 未来规划

THE FUTURE PLANNING

2. 苏宁监控开放平台设计



1. 背景介绍

BACKGROUND INFORMATION

2. 监控开放平台设计

THE DESIGNATION OF MONITOR OPEN PLATFORM

3. 异常检测平台实践

THE PRACTICE OF ANOMALY DETECTION PLATFORM

4. 未来展望

THE FUTURE PLANNING

3.1 异常检测背景

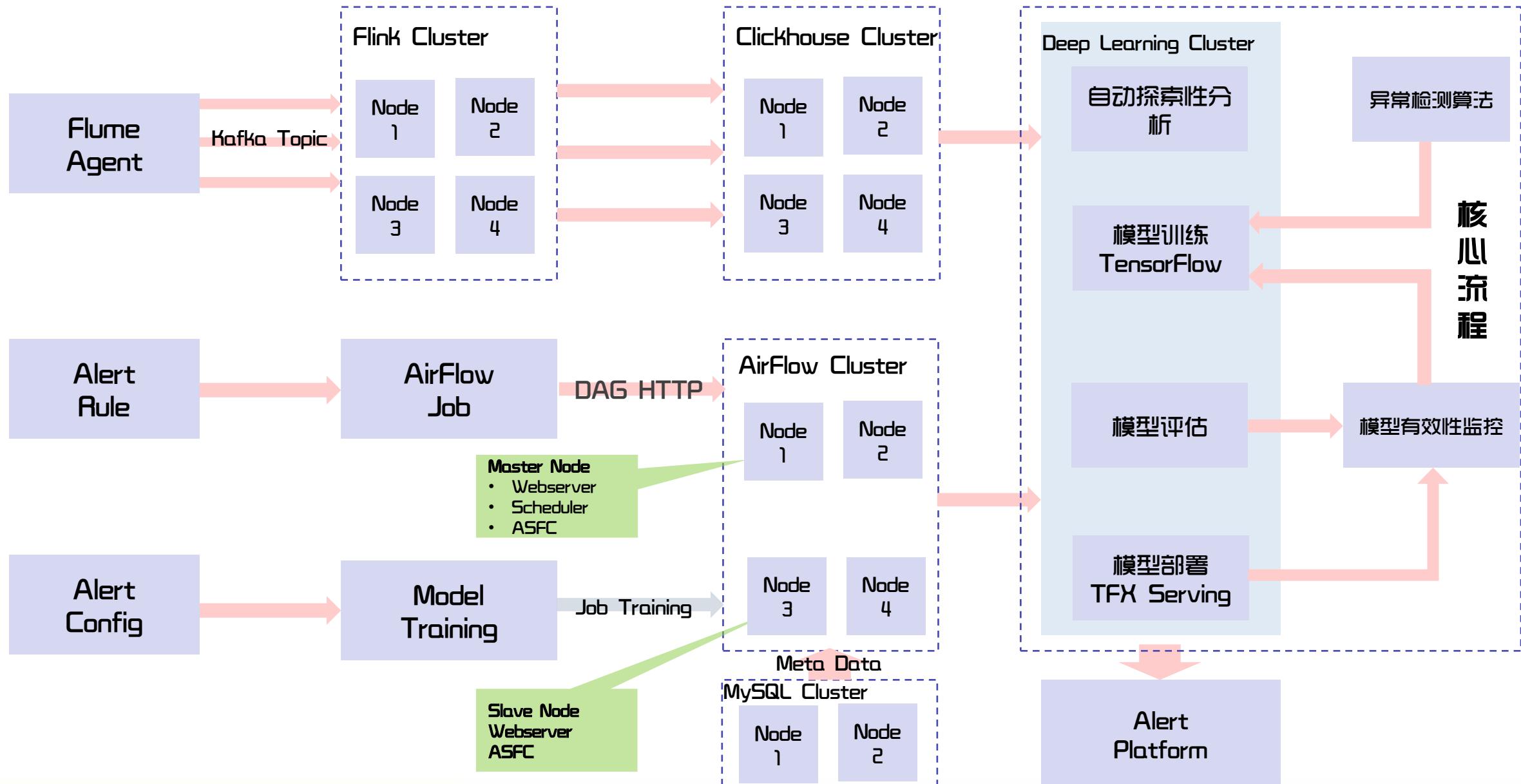
■ 背景

- **体量大**: 现在易购四级页每天产生的业务日志数据在100T以上, 业务监控都是多维度实时监控, 核心数据以1分钟为周期, 一般监控的数据以5分钟或1小时为周期, 监控目标非常多, 按人工维护这些监控的阈值、启停等几乎是很难达到。
- **变化多**: 易购四级页监控对象及指标变化也非常多, 业务指标也有周期性变化的特点, 在日常的促销活动或大促活动期间, 这些监控对象及指标也是经常调整相关策略, 很难保障人工设定的静态报警阈值准确性。
- **迭代快**: 在不同的时期, 易购经常也会上不同的促销活动, 监控的对象和指标维度变化比较频繁, 采用当前这种传统的静态报警阈值不能快速的反映线上实时的业务健康情况。

■ 待解决的问题

- 类似于四级页商品无货的场景, 业务上希望能够监控到什么品类、什么品牌、什么经销渠道、什么地区等比较细颗粒度的维度监控, 这样就能针对性的做好商品无货的运维了, 当前拿到的商品无货的监控维度比较粗
- 当前对于商品无货的监控采用的是传统的人工维护静态阈值, **运维成本比较高**
- 告警短信过多, 一天几百条短信, 无法判断哪些是有效告警, **告警准确率低, 运维难度大**

3.2 异常检测流程



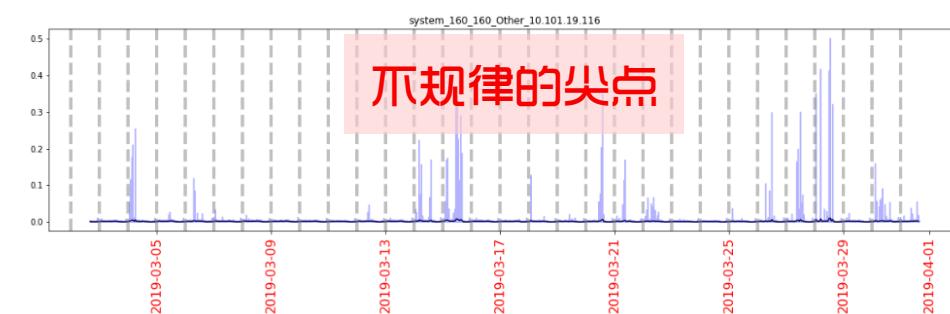
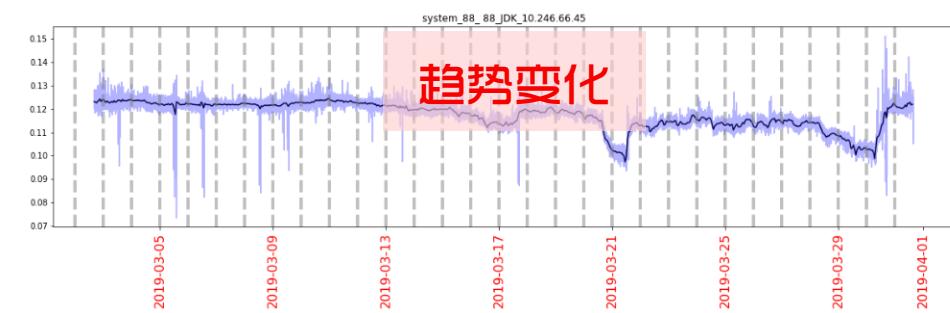
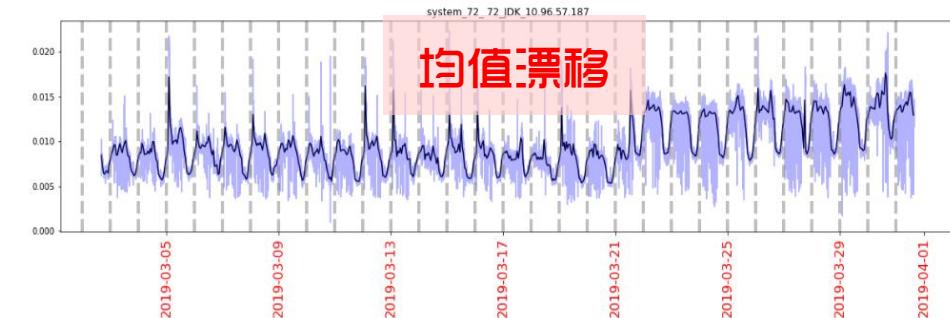
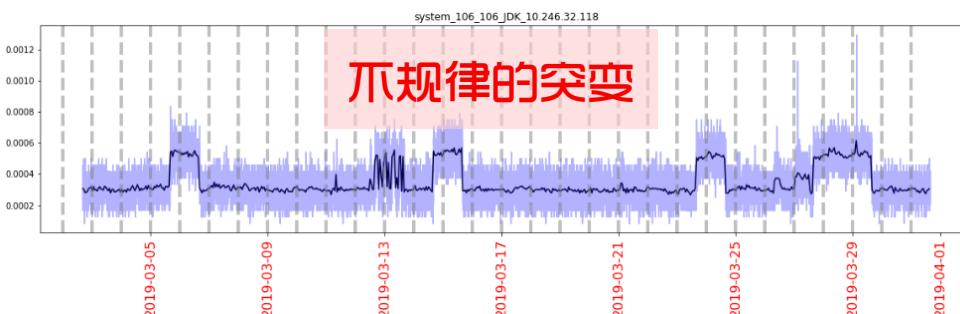
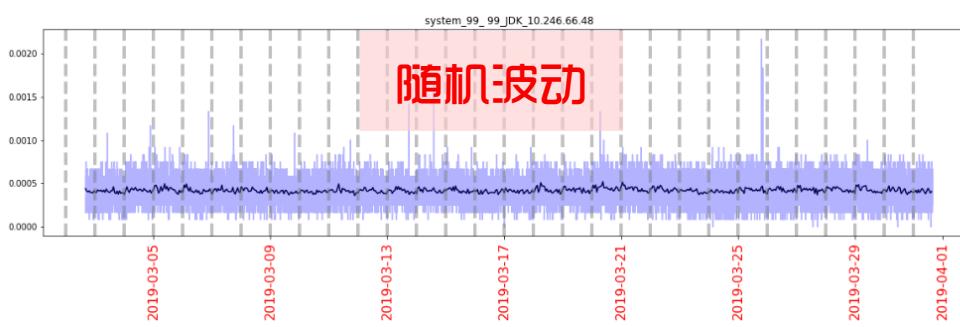
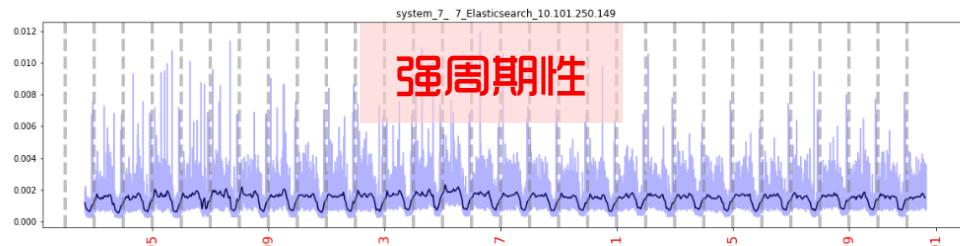
3.3 探索性数据分析

探索性数据分析（Exploratory Data Analysis, EDA）通过一系列手段（如可视化、假设检验等）形成数据的整体概貌，为后续选择算法进行建模预测提供参考。主要的EDA手段包括：

- 数据可视化（`run sequence plot`、`distplot`、`boxplot`…）
- 正态性检验（`jarque-bera`检验、`Shapiro-Wilk`检验…）
- 平稳性检验（`adfuller` 检验）
- 时间序列分解（`seasonal decompose`）（`trend + seasonality + residual`）
- 聚类分析（DBSCAN）（形态相近的曲线归为一类，构成时间序列的大致类别）
- 周期性分析
- 突变点检测

3.3 探索性数据分析

某系统虚机CPU利用率典型数据形态：紫色为1min粒度，黑色为聚合到1h粒度

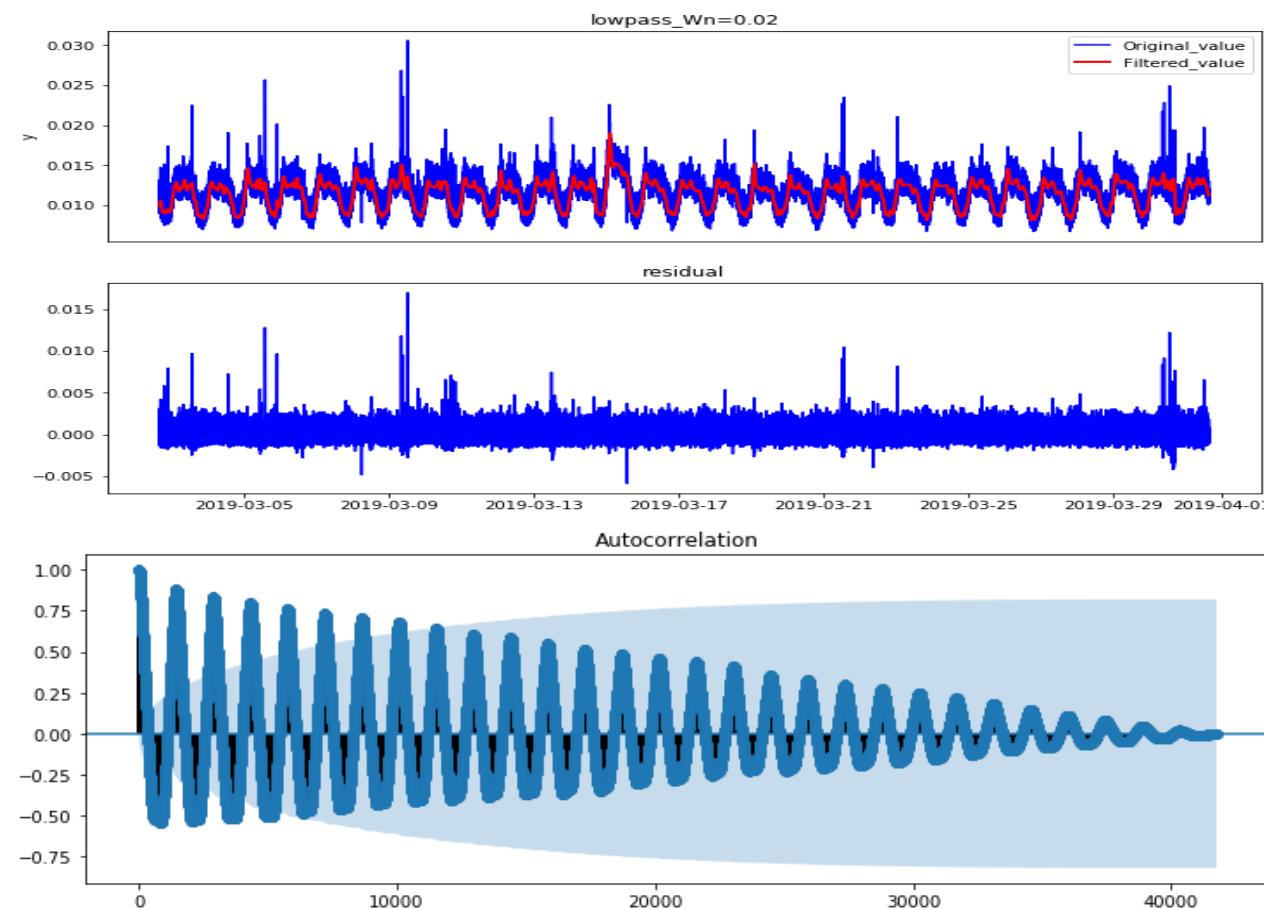


3.3.1 探索性数据分析之周期检测

时间序列周期检测流程：



- 首先去除趋势的影响，趋势会影响周期性的判断。
- 滤波是为了解决多重周期性的问题。我们采用分而治之的策略，分别对滤波后的时间序列（低通，中通，高通），进行周周期，天周期，小时周期的检测。
- 使用ACF方法进行周期检测。通过计算自相关系数的局部极大值的间隔来确定周期，对异常比较鲁棒。

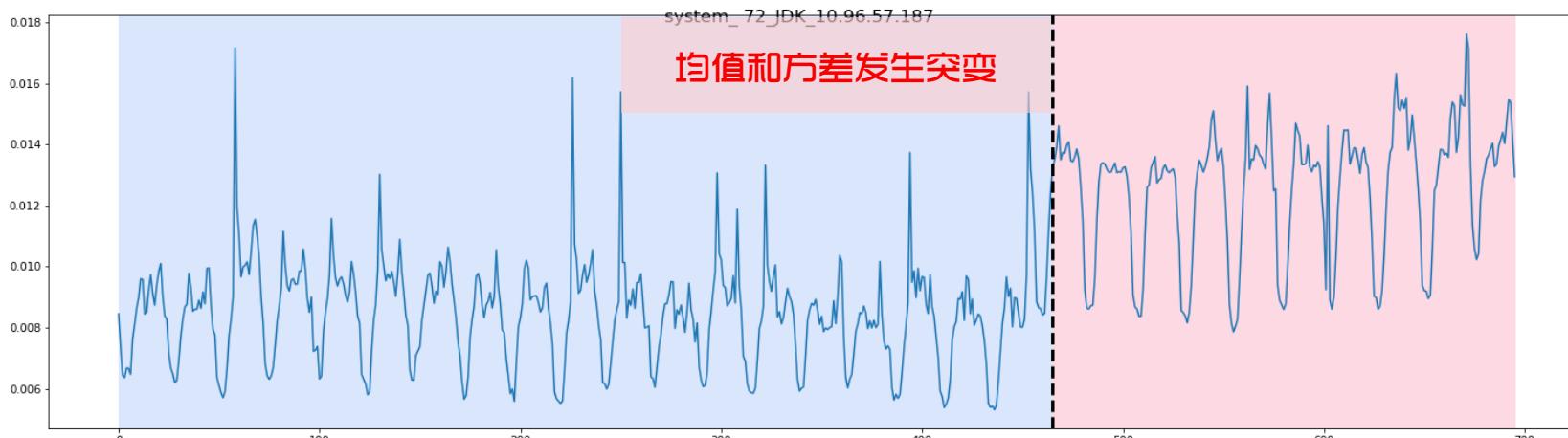


3.3.2 探索性数据分析之突变点检测

ruptures突变点检测算法



检测效果



突变点检测的意义：

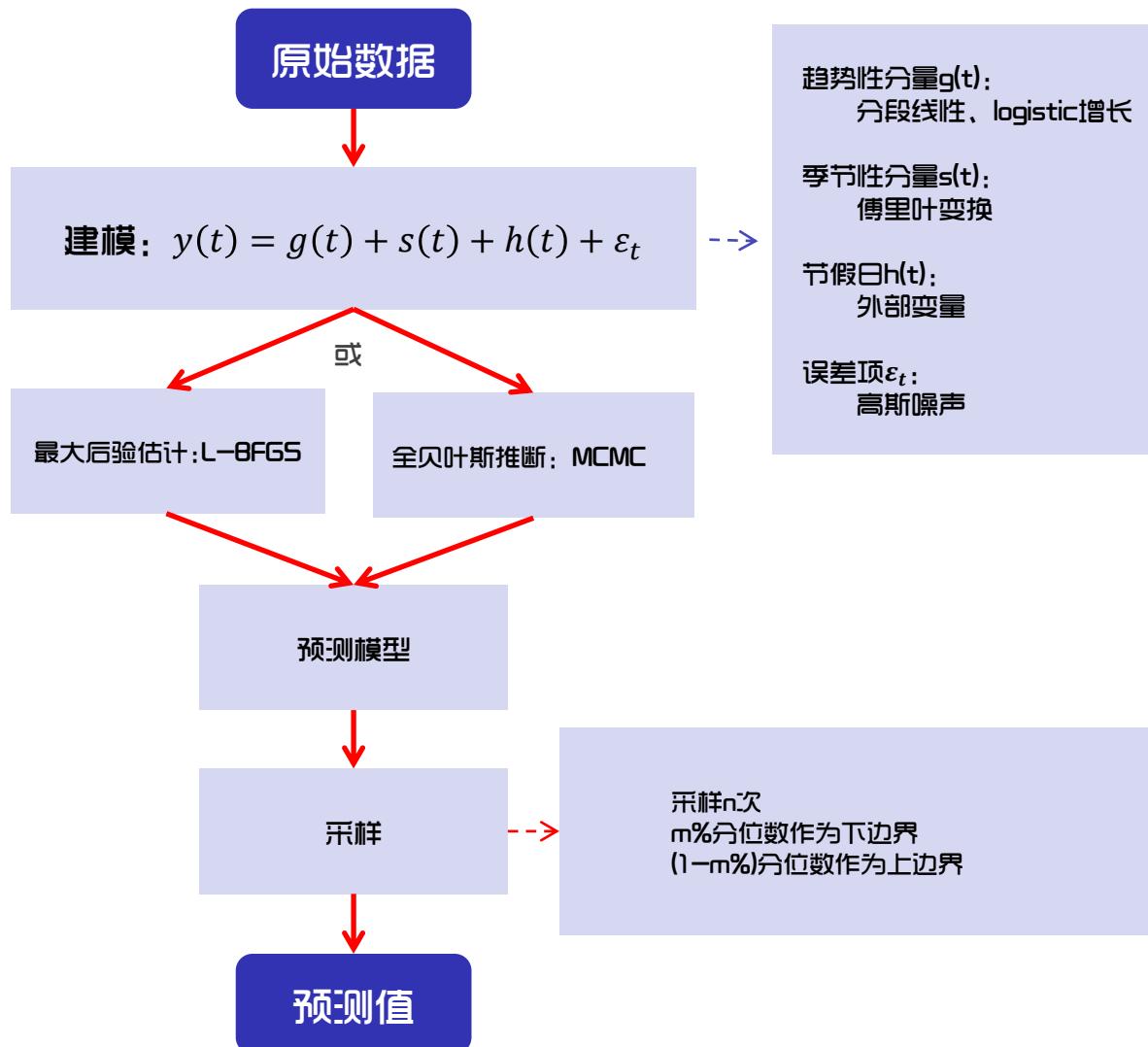
数据发生模式改变后，普通算法往往不能跟随模式变化，导致预测性能降低；
Kalman Filter算法可以很好地适应模式改变，对于发生模式改变的数据可优先考虑选择Kalman Filter算法。

3.4 异常检测算法

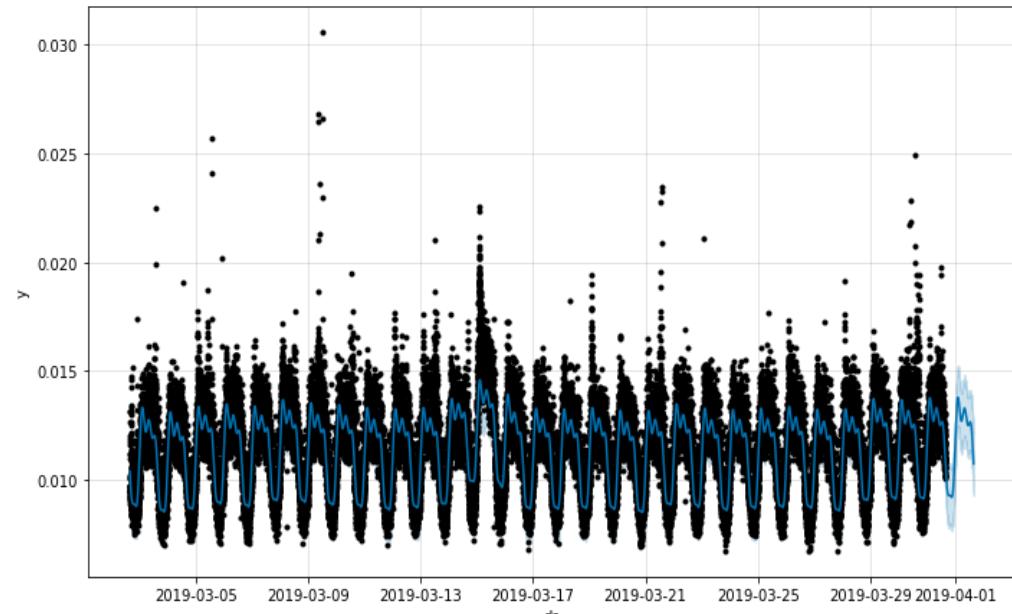
主要异常检测算法手段包括：

- Prophet
- Kalman Filter
- Fourier Extrapolation
- ARNN
- DeepAR
- 其他NN算法

3.4.1 异常检测算法之Prophet



预测效果:



pros

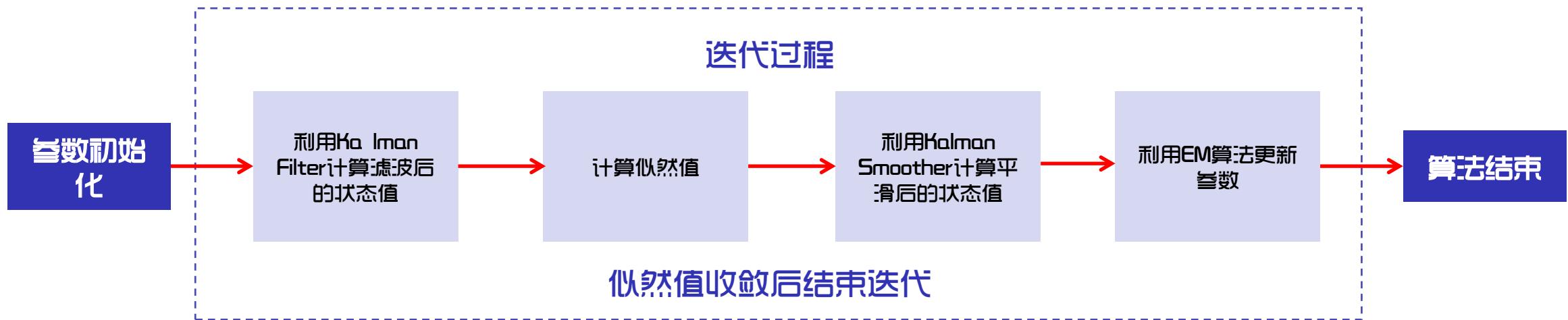
- ✓ 能够对趋势进行分段拟合，适用于趋势发生变化的数据
- ✓ 能够提取多重周期（天、周、年）

cons

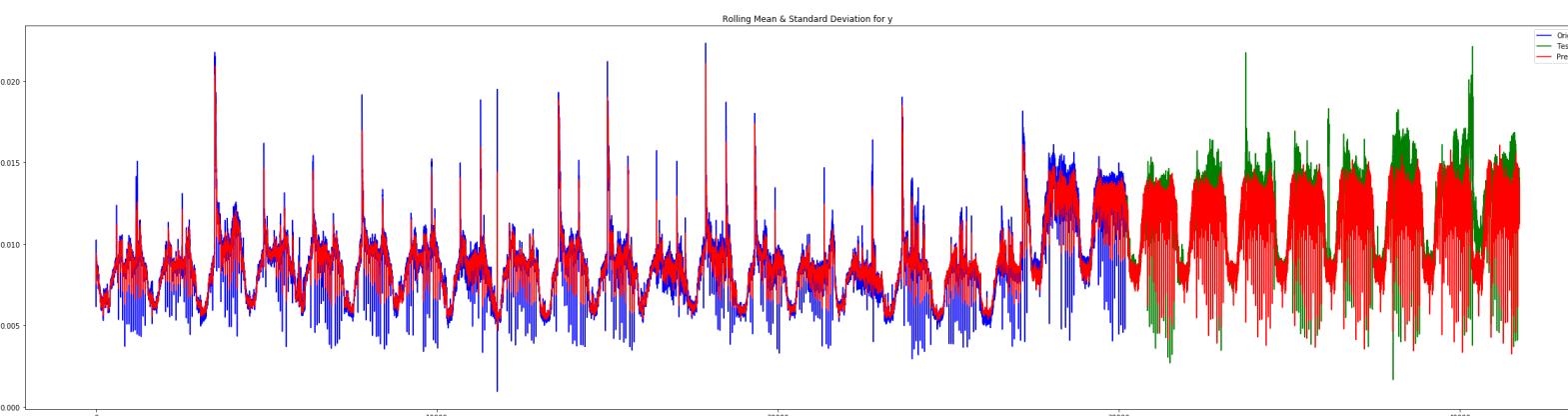
- ◆ 预测曲线平滑，对急剧变化的数据（如尖点）预测效果较差
- ◆ 仅支持提取天、周、年周期，不支持提取其他粒度的周期

3.4.2 异常检测算法之Kalman Filter

实现流程：



预测效果：



pros

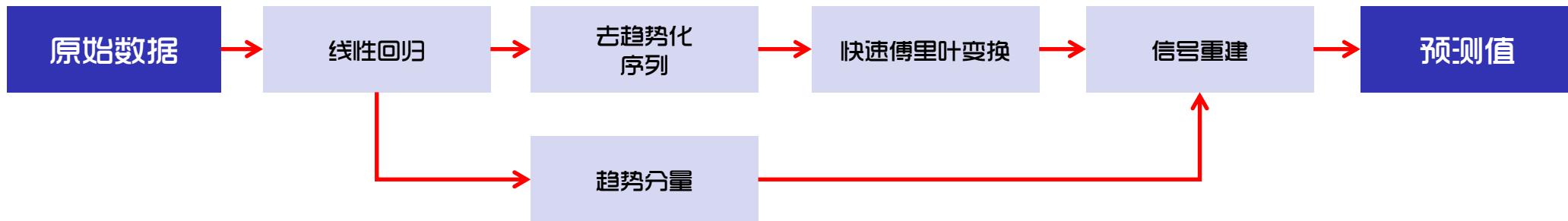
- ✓ 能够跟随模式变化
- ✓ 能够较准确地捕捉尖点发生的时刻

cons

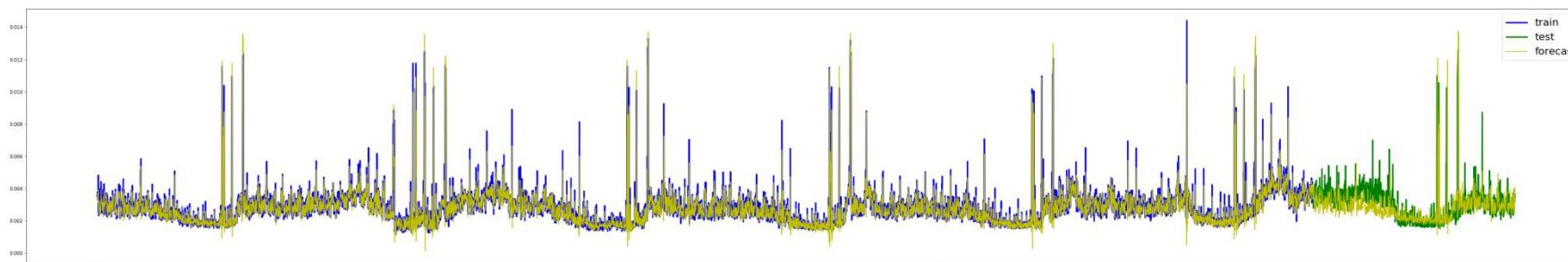
- ◆ 算法本身不能检测数据周期，需实施周期检测作为前置步骤
- ◆ 不能提取多重周期

3.4.3 异常检测算法之Fourier Extrapolation

算法流程：



预测效果：



pros

- 能够提取任意数量的频率成分
- 可准确捕捉数据中有规律的尖点位置，适用于变化剧烈的数据

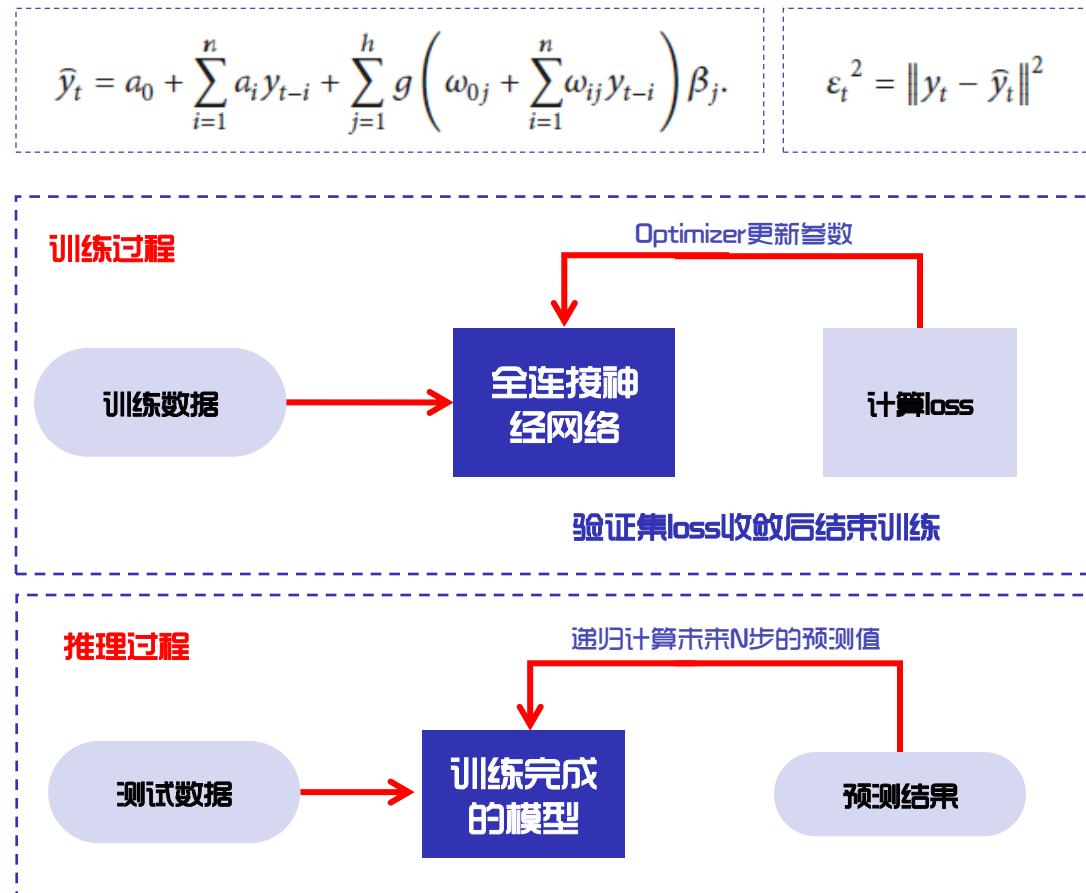
cons

- 以线性回归的方式提取趋势，不适用于趋势发生变化的数据
- 不能直接预测上下边界，须以其他方式指定

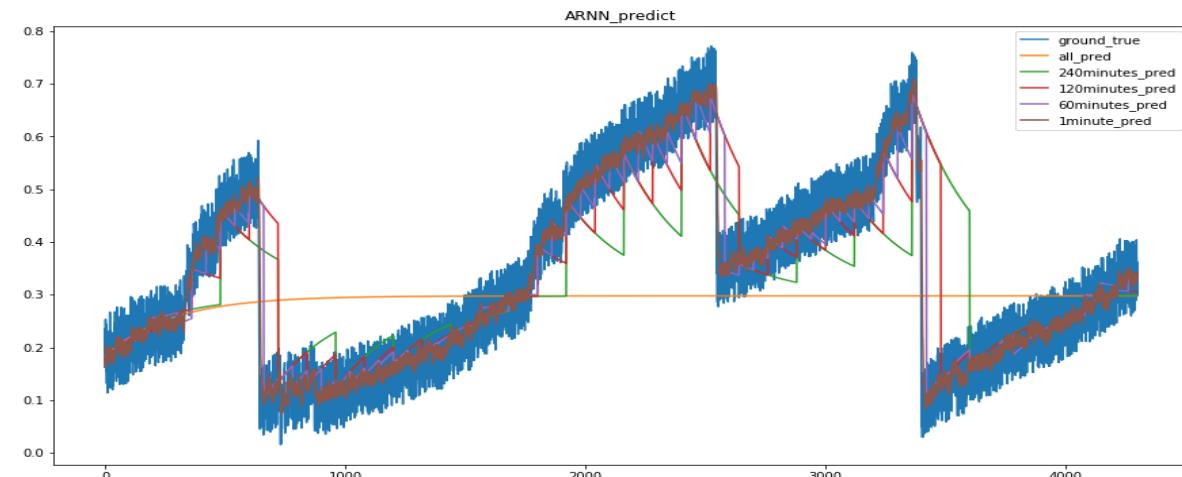
3.4.4 异常检测算法之ARNN

算法原理及流程：

- ◆ ARNN本质上是一个全连接神经网络，输入为t个历史数据，输出为第t+1个预测数据，loss为预测值和实际值的平方误差



预测效果：



预测范围	mape
3 day	50.216446
240 minutes	27.331686
120 minutes	20.674031
60 minutes	17.315611
1 minute	16.675982

Pros

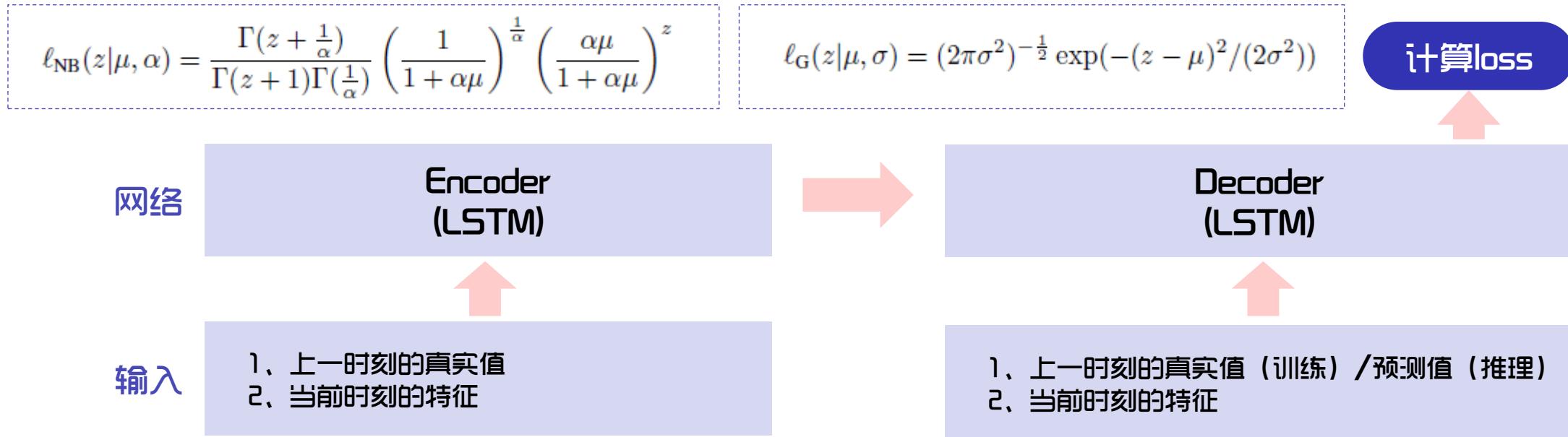
- 模型简单，训练速度快
- 对非周期数据的短期预测效果较好

Cons

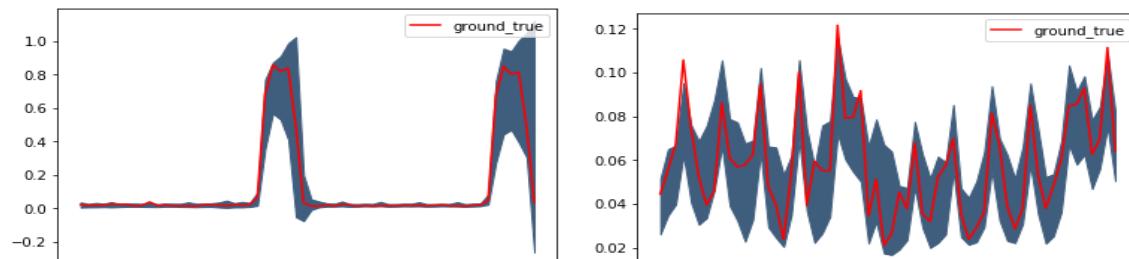
- 由于误差累积，长期预测效果较差
- 只利用了时间序列自身值的信息，无法利用外部特征
- 不同的时间序列需要单独建模，无法应用于海量数据场景

3.4.5 异常检测算法之DeepAR

算法原理及流程：



预测效果：



Pros

- 对相关的时间序列建立统一的预测模型，适用于海量数据场景
- 可以同时进行点预测和概率分布预测
- 冷启动预测，实现少量历史数据预测

Cons

- 没有使用attention机制，LSTM对较长的时间序列可能会出现记忆丢失的问题，无法捕获长周期、季节等信息

3.4.5 异常检测算法之DeepAR效果评估

与其他非神经网络方法的对比：

	SMAPE				
	DeepAR	Kalman	Mean	MAD	Prophet
count	240	240	240	240	240
mean	0.205501	0.26524	0.285353	0.290153	0.235895
std	0.092802	0.228215	0.284813	0.383116	0.213262
min	0.049374	0.002208	0.001974	0.001858	0.001845
25%	0.131634	0.144878	0.130986	0.123185	0.127684
50%	0.189284	0.228589	0.229291	0.199145	0.195503
75%	0.278271	0.328544	0.333387	0.241119	0.31819
max	0.563849	1.292141	1.739594	1.802471	1.872138

- 评估指标：由于240个时间序列scale差异较大，使用scale independent的SMAPE作为评估指标
- 评估数据：240个时间序列上的整体预测效果，预测值为时间序列的最后460个点
- 评估结论：DeepAR算法相对其他非神经网络算法整体效果明显，适用于大规模时间序列的统一建模

3.4.6 异常检测算法之其他NN预测模型对比

模型	优点	缺点
DeepAR	<ol style="list-style-type: none">1. 可以同时进行点预测和概率分布预测2. 对实数和计数分别设计了不同的loss3. 冷启动预测，实现少量历史数据预测4. 数据预处理方面使用Scale变换和weighted sampling	<ol style="list-style-type: none">1. 没有使用attention机制，LSTM对较长的时间序列可能会出现记忆丢失的问题，无法捕获长周期、季节等信息
seq2seq	<ol style="list-style-type: none">1. 引入简单的lag-attention机制，解决了长时间序列的记忆丢失问题2. 训练时decoder使用预测值作为下一个时间点的输入，增加模型的稳定性3. 使用 cuDNN GRU、COCOB optimizer等新技术，加快模型训练和收敛速度4. 使用ensemble learning、ASGD等技术降低模型variance	<ol style="list-style-type: none">1. 只能进行点预测
ES-ANN	<ol style="list-style-type: none">1. 使用ES+RNN的层次模型，其中ES模型可以捕获每个时间序列的个性，RNN可以捕获全局特征2. 可以同时进行点预测和区间预测	<ol style="list-style-type: none">1. 模型复杂，由于使用ES捕获每个时间序列的个性，每个时间序列的计算图都不相同；月度、季度、年度等不同时间粒度的RNN模型采用了不同的结构。2. 没有使用外部特征，只使用了时间序列的数值信息
DSSM	<ol style="list-style-type: none">1. 和DeepAR等直接将目标值作为网络输入的方法比，该模型对噪声的鲁棒性较好，因为目标值通过似然项引入，噪声相对较少2. 容易处理缺失值，可以直接丢弃缺失值对应的似然项3. 对趋势性、周期性、季节性较强的结构性时间序列，需要的训练数据较少	<ol style="list-style-type: none">1. 由于SSM模型适用于趋势性、周期性、季节性较强的结构性时间序列，该模型对无此特征的时间序列预测效果未知
DFGP	<ol style="list-style-type: none">1. 由于增加了GP，相比其他DNN模型，局部不确定性估计得更好2. 由于使用了attention，长期预测效果明显比其他模型要好，和短期预测结果差距也不大	<ol style="list-style-type: none">1. 对周期数据预测结果较好，非周期数据结果未知

3.5 异常检测模型构建

• 1、原始数据存储

- 消费者业务数据包括登录、支付、注册、四级页等，数据以分钟为粒度聚合
- 原始数据存储在Clickhouse集群中

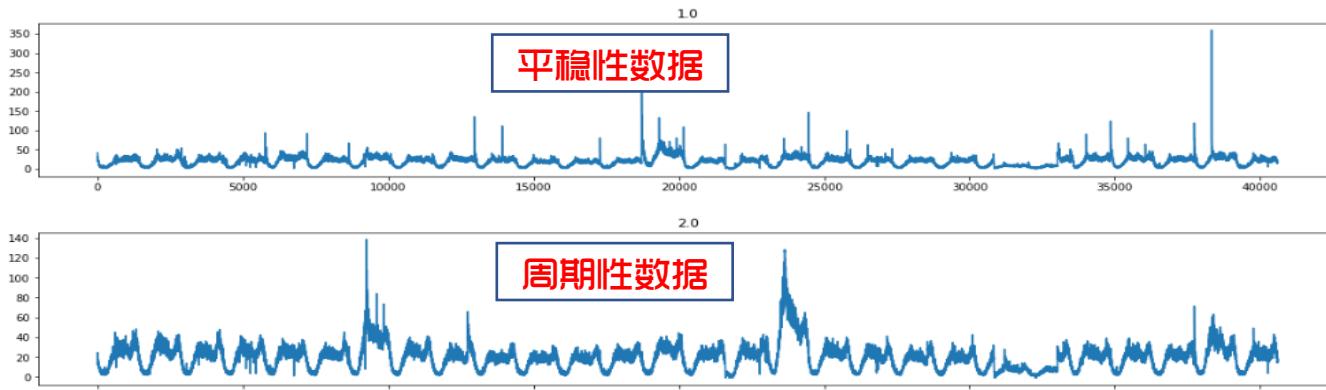
• 2、数据预处理

- 通过Pandas、Numpy等数据处理工具将原始多维数据转换为单维分钟粒度时间序列
- 构建数据清洗模块，清洗内容包括异常点剔除，缺失值填充等
- 利用StatsModels工具对时间序列进行特征检测，包括周期性检测、平稳性检测、趋势检测和突变检测

• 3、模型设计及训练

- 由于时间序列规模量大，传统的对每条时间序列单独建立预测模型的方法已不适用
- 考虑使用深度神经网络构建统一预测模型，学习相关时间序列的全局和局部模式

	dt	city	loginType	loginChannel	memberRole	cnt
0	2019-04-22 00:00:00	秦皇岛市	AuthCodeCredentialsAuthenticationHandler	2.080002e+11	142000000154	2
1	2019-04-22 00:00:00	上海市	SuningUsernamePasswordAuthenticationHandler	2.080001e+11	142000000154	5
2	2019-04-22 00:00:00	太原市	RememberMeCredentialsAuthenticationHandler	2.080002e+11	142000000154	17
3	2019-04-22 00:00:00	汕头市	TrustLoginCredentialsAuthenticationHandler	2.080002e+11	142000000154	1
4	2019-04-22 00:00:00	佛山市	AuthCodeCredentialsAuthenticationHandler	2.080002e+11	142000000154	1



3.5 异常检测模型构建

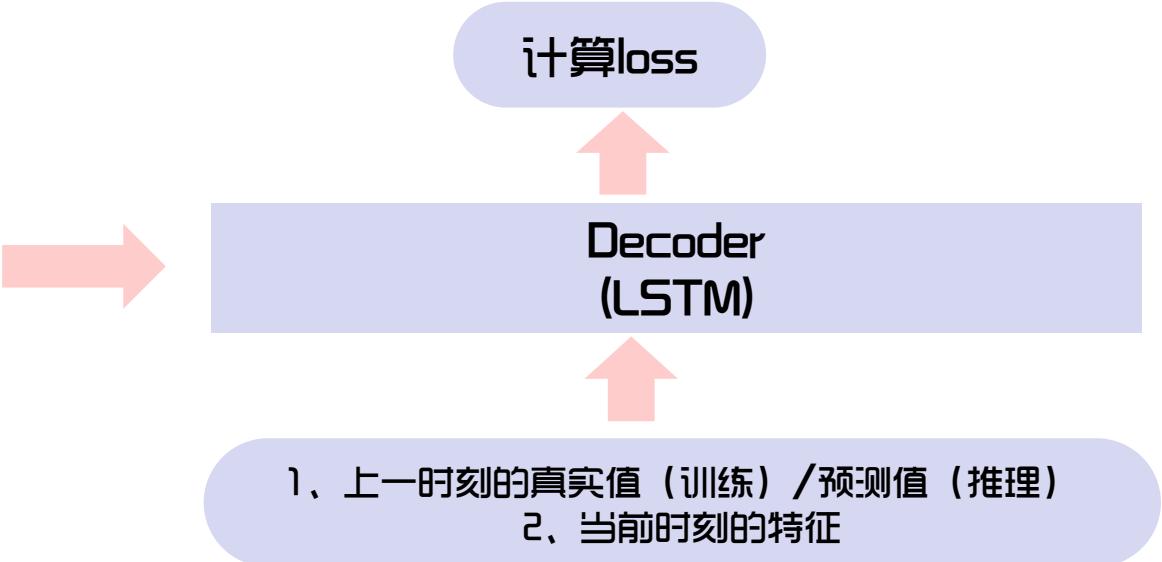
• 3、模型设计及训练

网络

Encoder
(LSTM)

输入

- 1、上一时刻的真实值
- 2、当前时刻的特征

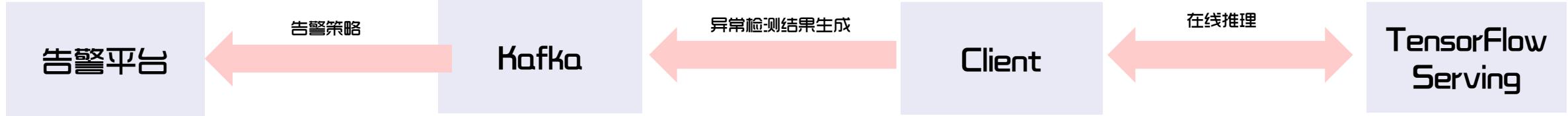


- 使用TensorFlow深度学习框架实现神经网络时间序列预测模型，并进行迭代训练
- 经过多轮超参优化后确定最优超参及预测范围，利用历史400分钟数据预测未来30分钟数据的概率分布
- 将原始模型转换为TensorFlow Serving可以调用的轻量化模型，并保存至模型版本库中

超参数	说明	值
encoder_len	编码器长度	400
decoder_len	解码器长度	30
embedding_size	类别嵌入尺寸	20
hidden_size	LSTM隐层尺寸	128
layer_num	LSTM层数	3
keep_prob	Dropout保持率	0.8
batch_size	批尺寸	128
lr	学习率	0.001

3.6 异常检测模型部署与监控

模型部署上线检测



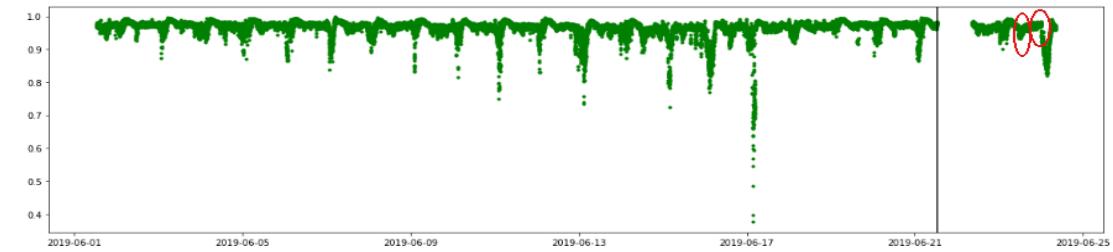
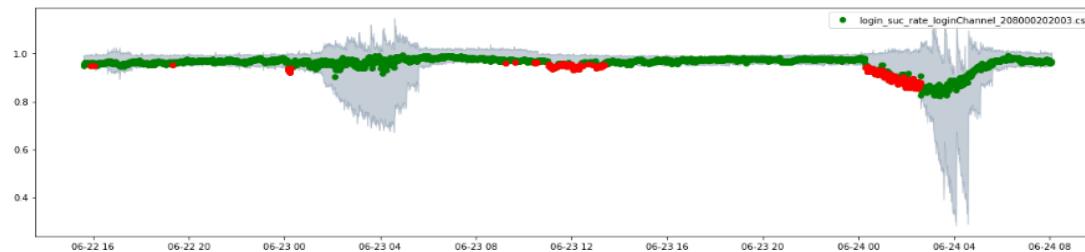
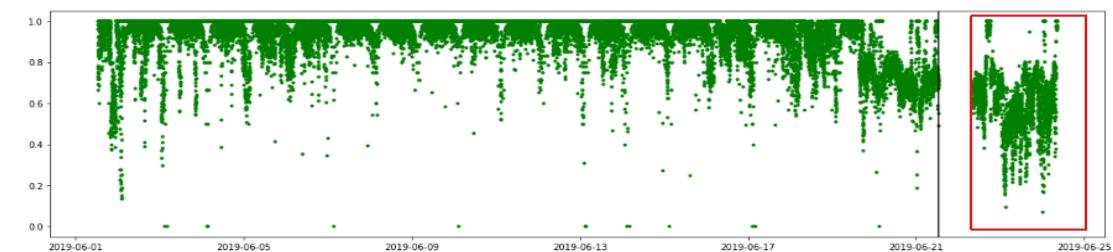
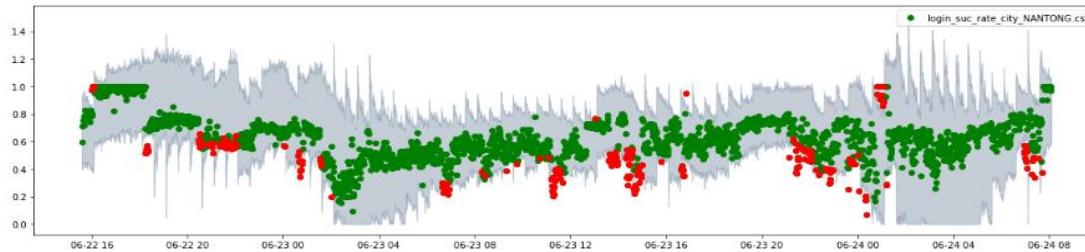
- 后台每20分钟调度一次Client，Client从ClickHouse中读取最近的历史400分钟数据，调用Serving进行在线推理
- Client得到预测结果后计算异常warning及critical边界值， warning边界取90%分位数， critical边界取98%分位数
- Client将warning及critical边界值发送至Kafka
- 持续5分钟异常产生告警，告警平台根据前台配置策略向告警信息接收人推送豆芽、短信、邮件告警信息

模型监控

- 数据监控：通过均值、标准差、中位数、IQR等统计指标以天为粒度主动监控数据漂移，数据漂移超过阈值后模型自动提取最新数据重训练
- 模型监控：每天自动统计模型输出的全部时间序列异常点占比的概率分布，概率分布变化超出阈值后模型自动提取最新数据重训练

3.7 异常检测结果展示

异常时间序列： 绿点为正常真实点， 阴影为预测的上下边界， 红点为异常真实点



告警信息：

智能告警监控

告警对象	全部	接收人	查询	2019-07-10 至 2019-07-10
告警名称:	登录产品线deepAR异常告警			
告警对象:	登录			
告警内容:	【登录】的【失败数】通过DeepAR动态阈值算法在2019-07-10 09:44:00已经发出【严重】级别告警，对应的维度为：【渠道:2342355; 统计值为：231，总数为556 严重下边界为：3 告警下边界为：5 告警上边界为：32 严重上边界为：65			
通知人员:	张大帅			
是否通知:	已通知			
告警名称:	登录产品线deepAR异常告警			
告警对象:	登录			
告警内容:	【登录】的【失败数】通过DeepAR动态阈值算法在2019-07-10 09:44:00已经发出【严重】级别告警，对应的维度为：【渠道:2342355; 统计值为：231，总数为556 严重下边界为：3 告警下边界为：5 告警上边界为：32 严重上边界为：65			
通知人员:	张大帅			
是否通知:	已通知			

3.8 异常检测算法库—随机砍伐森林(RRCF)

尚待解决的问题

- 1、实际业务数据的模式可能会经常发生变化，模型如何适应这种变化？
- 2、不断有新的时间序列接入，在没有充足历史数据的情况下，如何对新的序列实施异常检测？

RRCF

Pros:

- 1、模型实时更新，可以较快跟随数据模式的改变；
- 2、针对新接入的数据，可实现冷启动，随着数据的流入，检测准确率不断提升；
- 3、适用于流式检测场景；

Cons:

- 1、需对每一个序列单独建模，应用于大规模序列时，资源开销较大；

参考网

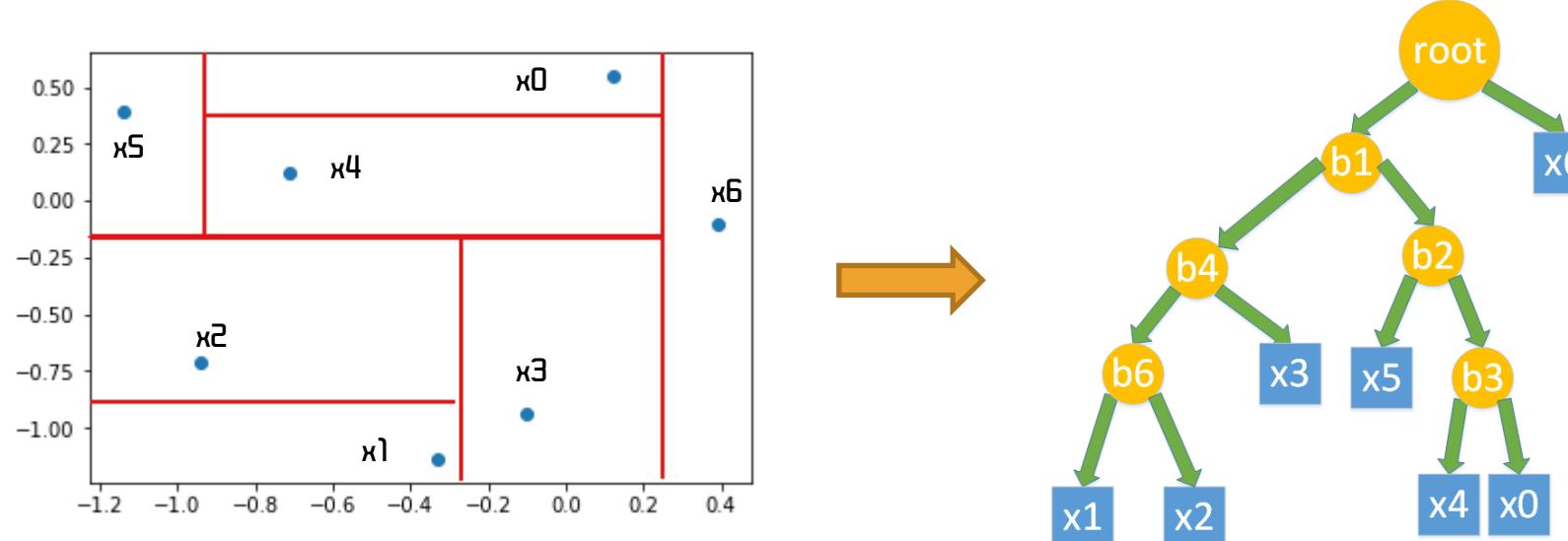
址：https://docs.aws.amazon.com/zh_cn/sagemaker/latest/dg/randomcutforest.html

3.8 异常检测算法库—随机砍伐森林(RRCF)

树的构建：

- 1、时间序列转换为m维样本点；
- 2、依一定概率随机选取切分维度q，概率正比于各维度值的跨度；
- 3、对选定维度依均匀分布随机选择切分点p；
- 4、维度q和切分点p构成一次切分，将点集划分为两个子集；
- 5、迭代步骤2、3、4，直至每一个子集只包含一个样本点。

异常得分： $score(x) = E_{S \subseteq \mathbb{Z}, T} \left[\max_{x \in C \subseteq S} \frac{1}{|C|} \sum_{y \in S - C} (f(y, S, T) - f(y, S - C, T'')) \right]$



3.8 异常检测算法库—随机砍伐森林(RRCF)

以哈尔滨市4—5月份的登录成功数为例：

训练数据准备

原始时间序列：

time, value

2019-04-22 00:00:00,67.0
2019-04-22 00:01:00,23.0
2019-04-22 00:02:00,26.0
2019-04-22 00:03:00,31.0
2019-04-22 00:04:00,24.0
2019-04-22 00:05:00,24.0
2019-04-22 00:06:00,22.0
2019-04-22 00:07:00,29.0
2019-04-22 00:08:00,25.0
2019-04-22 00:09:00,28.0
2019-04-22 00:10:00,32.0
2019-04-22 00:11:00,26.0
2019-04-22 00:12:00,16.0
.....

滑动窗口
window size=5,
step=1

高维数据点：

X=[(67.0,23.0,26.0,31.0,24.0),
(23.0,26.0,31.0,24.0,24.0),
(26.0,31.0,24.0,24.0,22.0),
(31.0,24.0,24.0,22.0,29.0),
(24.0,24.0,22.0,29.0,25.0),
(24.0,22.0,29.0,25.0,28.0),
(22.0,29.0,25.0,28.0,32.0),
(29.0,25.0,28.0,32.0,26.0),
(25.0,28.0,32.0,26.0,16.0),
.....]

数据分区：

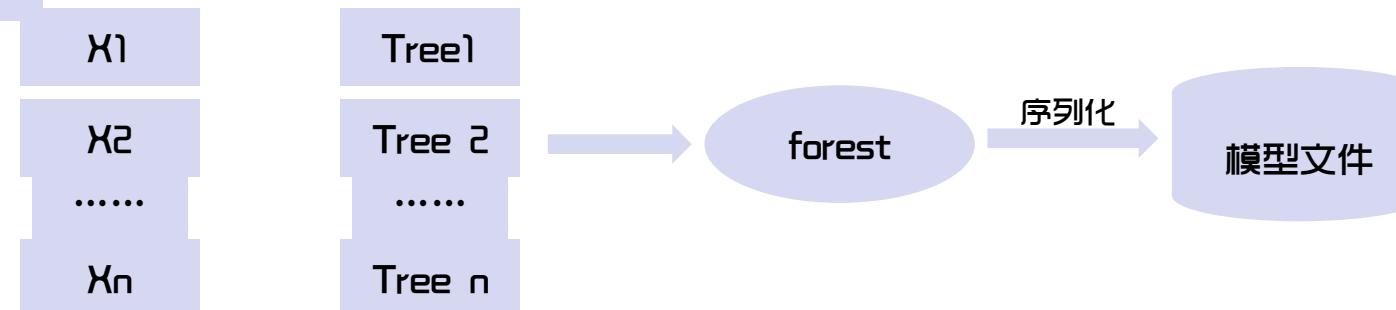
X1=[(67.0,23.0,26.0,31.0,24.0),
(23.0,26.0,31.0,24.0,24.0),
(26.0,31.0,24.0,24.0,22.0),
.....]

X2=[.....]

.....

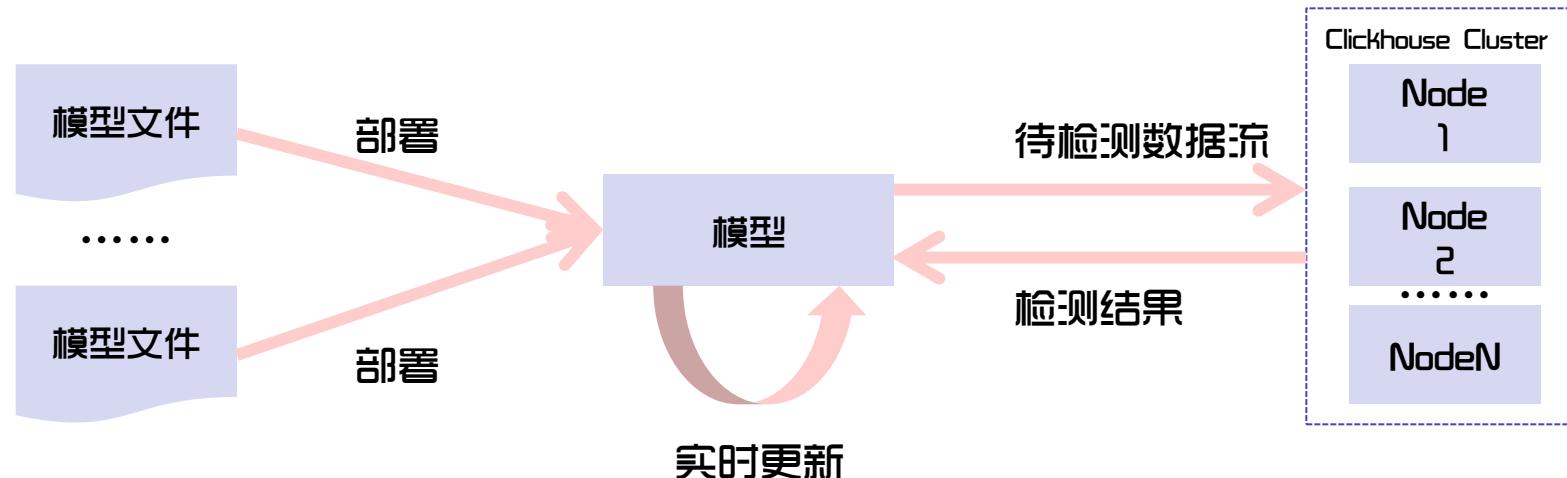
Xn=[.....]

模型构建

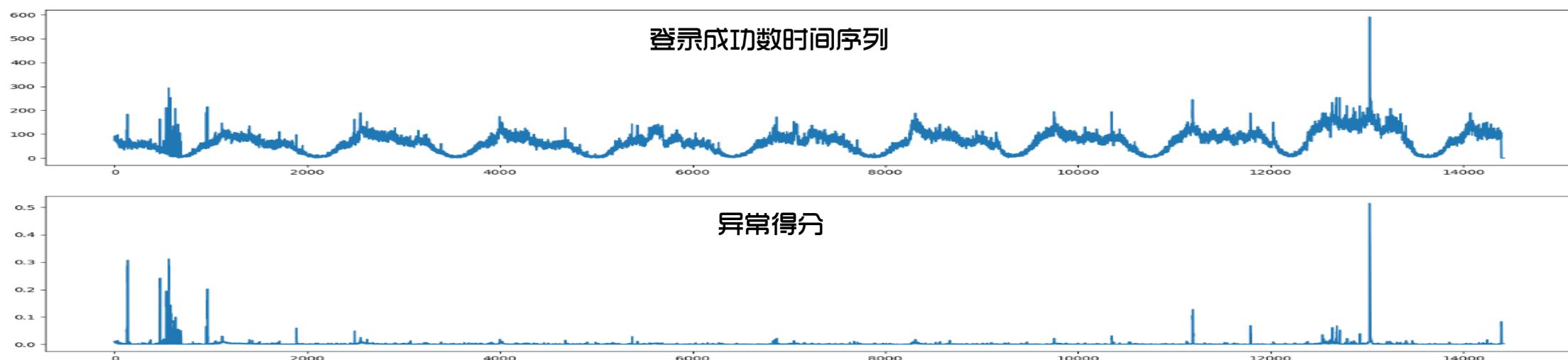


3.8 异常检测算法库—随机砍伐森林(RRCF)

异常检测



算法效果



1. 背景介绍

BACKGROUND INFORMATION

2. 监控开放平台设计

THE DESIGNATION OF MONITOR OPEN PLATFORM

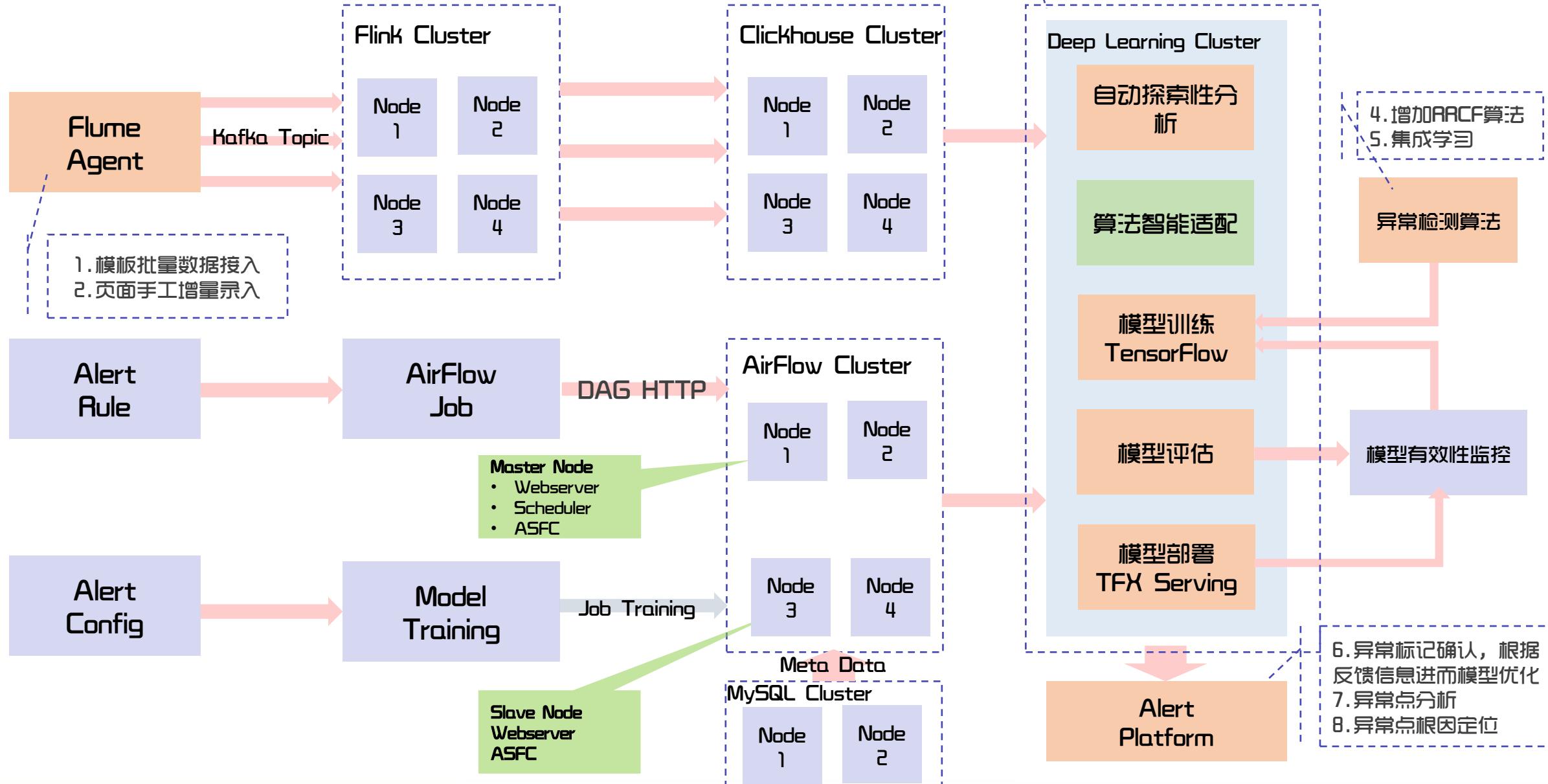
3. 异常检测平台实践

THE PRACTICE OF ANOMALY DETECTION PLATFORM

4. 未来规划

THE FUTURE PLANNING

4 未来规划





THANKS!
Q&A