2 数据分析

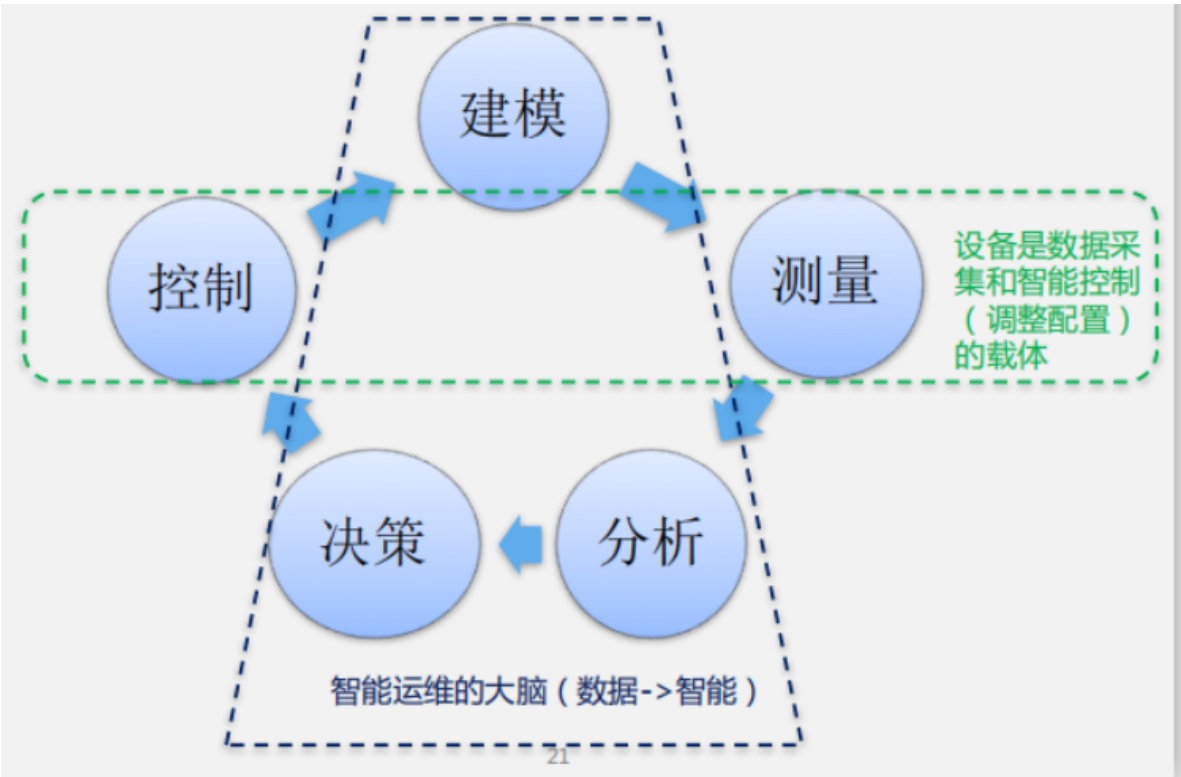


图2-1 智能日志分析流程图

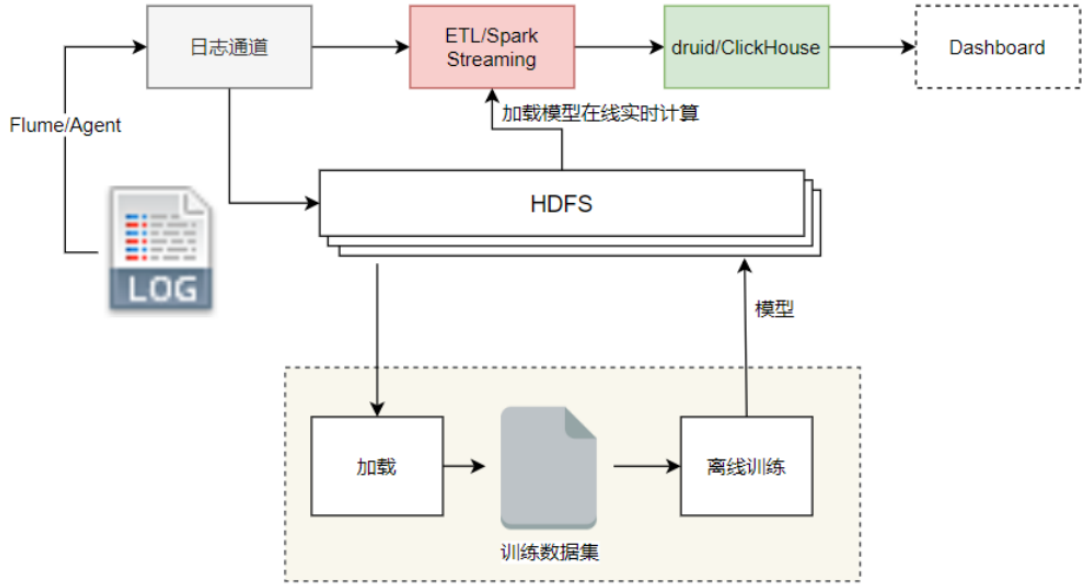


图2-2 机器学习日志分析架构图

2.1 异常检测

能异常检测的主体框架大多遵从少见即异常原则。小概率发生的事件就判断为异常，如何划定小概率事件在日常的运维工作中也很重要，宽松了容易漏告，严格了又会造成骚扰。

稳定序列（stationary series）：可直接使用机器学习算法(孤独森林，SVM等)和深度算法(DeepLog、Auto-Encoder等)。

非稳定序列（non-stationary series）:可直接使用时间序列模型(AR/MA/ARMA/ARIMA/Holt winters等）和深度算法；也可以进行下面处理后再使用无监督机器学习算法：

方法一：利用差分、log转换等方法，先将非稳定序列转成稳定序列后，然后使用非监督机器学习算法，可查看"stationarize the series"。

方法二：用STL等方法将序列分解成趋势（trend）、周期（season）和残差（residual），残差一般是稳定序列，然后仅对residual部分使用上述无监督机器学习算法,可参考页面。

2.1.1 指标异常检测（网关访问日志）

2.1.1.1

1. KPIs (Key Performance Indicators)：用来衡量服务性能的关键指标；
2. KPI异常行为：潜在的风险、故障、bugs、攻击……
3. KPI异常检测：在KPI时序曲线上识别异常行为

   a）诊断和修复；

   b）阻止进一步损失或潜在风险；

1、 CVAE 算法

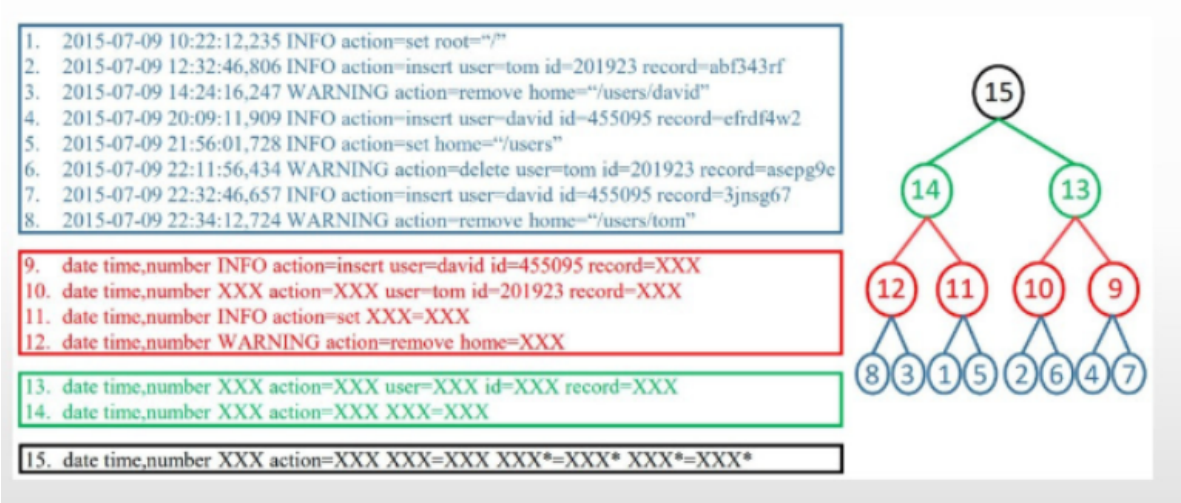
2、 iForest 算法

3、 KDE 算法

2.1.2 日志异常检测（业务日志）

除了指标的异常，还有就是日志的异常。针对日志异常的处理，最常见的日志告警就是关键字匹配。不过，大多数系统的研发，不会把日志写的那么规范，这时就需要 AI 算法来辅助。

1、日志模式 - 层次聚类



通过层次聚类能实现大量日志的模式发现，并进行聚类，将大量的日志原文转化为少量的**日志模式**，并反应相应模式在日志原文中的占比，大大减少了人工筛选时间，帮助运维人员更快的定位故障根因。

先对日志进行最基础的分词和类型判断，然后聚类合并。聚类可以用最长子串，也可以用文本频率等等。聚类里，不同的部分就用通配符替换掉。如上图所示：把 8 条日志，先合并成 4 个日志格式，合并成 2 个，再合并成 1 个。

（1）故障定位

一种是故障定位的时候。比如我们查错误日志，单纯用关键词，可能出来几百上千条。你要一个一个看过去，翻好几页，耗时就比较长了。如果内容字很多，还可能看漏了。

模式树的信息，可以直接查看匹配关键字的日志的模式情况，可能就只有那么三五条信息，一眼就可以看完，很快就可以知道问题在哪，就可以进行下一步了。

（2）异常检测

另一个用途，就是把得到的，加载到日志采集的实时处理流程里，进行异常检测，提前发现问题，这时候，我们除了模式，还可以检测参数，检测占比。

	词元1	词元2	词元3
日志1	we	are	80
日志2	we	are	100
日志3	you	are	100
模式	*	are	<NUM>
参数	enum{we, you}		X~N(93.3, 9.4²)

上图是一个最简单的示例，3 条日志，得到的模式是*are，然后我们同时可以检测符合这个模式的日志，前边的只能是 we 或 you，第三位只能落在平均值为 93.3、标准差为 9.4 的正态分布区间内。

然后日志采集进来，先检测一下日志模式是不是合法的。如果合法，再检测一下各个参数位置的取值是不是合法的。如果依然合法，再检测一下这段时间这个模式的日志数量，和之前相比是不是正常的。

这么三层检测下来，相当于把模式异常、数值异常、时序指标异常融合到了一起。

2.2 趋势分析

2.1.2.1 常用算法

> 统计学方法 - ARIMA 模型

2.3 根因分析

2.1.3.1 常用算法

G-RCA - 基于规则的根因分析；（运维人员人工给出）

> 机器学习 - *自动挖掘*模块报警事件之间的关联关系

> *数据*
>
> *标注*
>
> 工具（算法和系统）
>
> *应用*
>
> • “基于机器学习的智能运维”具有得天独厚的基础
>
> – 互联网应用天然有海量日志作为特征数据
>
> – 运维日常工作日志产生标注数据
>
> – 大量成熟的机器学习算法和开源系统
>
> – 直接用于改善互联网应用
>
> 运维日常工作产生标注数据 !!!!(给数据打标签) eg：problem type

- 基于学习的根因分析；