数据挖掘漫游

Mathew Shen(datahonor@gmail.com)

2023 年 8 月 3 日

1. Security
2. AIOps
   - TSAD: Algorithm
   - TSAD: System
3. QA

Security

帐号安全

Situation

- 自动机模拟登录验密
- 暴力破解，拖/洗/撞库

帐号安全

Situation

- 自动机模拟登录验密
- 暴力破解，拖/洗/撞库

Task

- 保证正常验密请求的前提下拦截黑产验密请求
- 降低盗号率及其他衍生指标

帐号安全

Situation

- 自动机模拟登录验密
- 暴力破解，拖/洗/撞库

Task

- 保证正常验密请求的前提下拦截黑产验密请求
- 降低盗号率及其他衍生指标

Action

- FPGrowth(PFP[11]) + Spark
- 基于请求特征挖掘关联规则，用于实时封禁

帐号安全

Situation

- 自动机模拟登录验密
- 暴力破解，拖/洗/撞库

Task

- 保证正常验密请求的前提下拦截黑产验密请求
- 降低盗号率及其他衍生指标

Action

- FPGrowth(PFP[11]) + Spark
- 基于请求特征挖掘关联规则，用于实时封禁

Result

- 挖掘出来的规则符合专家经验
- 拦截相当部分黑产验密请求

风控漫谈

查杀分离: 信息差

对能够很好定位黑产的特征应给予保护，不直接用于账号封禁等

风控漫谈

查杀分离: 信息差

对能够很好定位黑产的特征应给予保护，不直接用于账号封禁等

对抗的关键: ROI

提高攻击成本, 降低攻击 ROI 是关键

AIOps

AIOps 概览

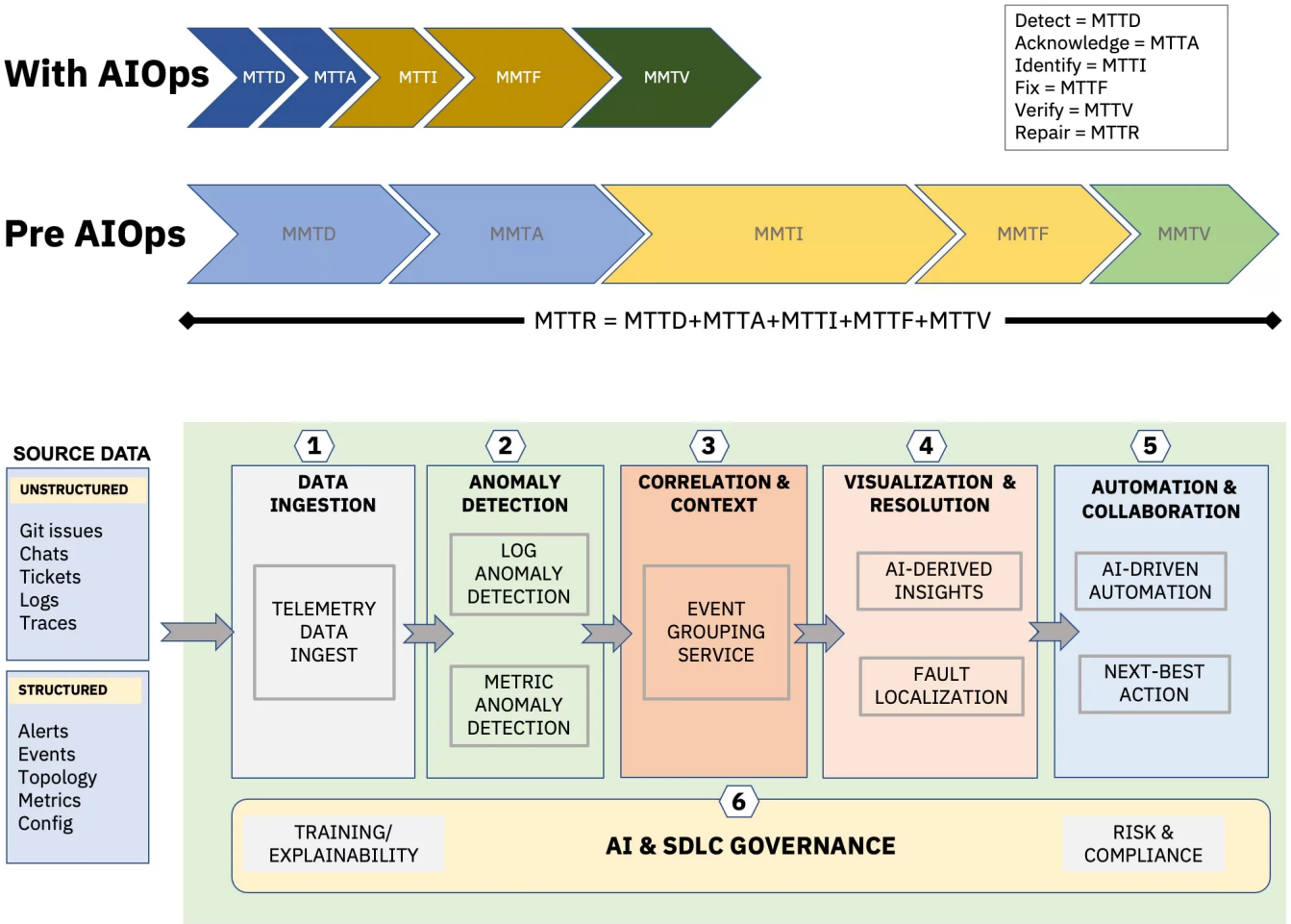


图 1: AIOps: A Path to Reliability at Cloud Scale[1]

[1] AIOps: A Path to Reliability at Cloud Scale

TSAD

Situation

- 大量指标需要监控以及时发现系统问题
- 人工阈值的方法难以自适应，告警噪声大

TSAD

Situation

- 大量指标需要监控以及时发现系统问题
- 人工阈值的方法难以自适应，告警噪声大

Task

- 检测各类时间序列存在的异常，及时发出预警信息
- Precision, Recall, MTTD

TSAD

Situation

- 大量指标需要监控以及时发现系统问题
- 人工阈值的方法难以自适应，告警噪声大

Action

- 构建算法库系统地检测各类时序异常
- 基于算法库搭建 TSAD 系统来解决业务问题

Task

- 检测各类时间序列存在的异常，及时发出预警信息
- Precision, Recall, MTTD

TSAD

Situation

- 大量指标需要监控以及时发现系统问题
- 人工阈值的方法难以自适应，告警噪声大

Action

- 构建算法库系统地检测各类时序异常
- 基于算法库搭建 TSAD 系统来解决业务问题

Task

- 检测各类时间序列存在的异常，及时发出预警信息
- Precision, Recall, MTTD

Result

- Precision>80%
- Recall $\approx$ 100%
- MTTD(<3 Min)

TSAD: 算法/模型

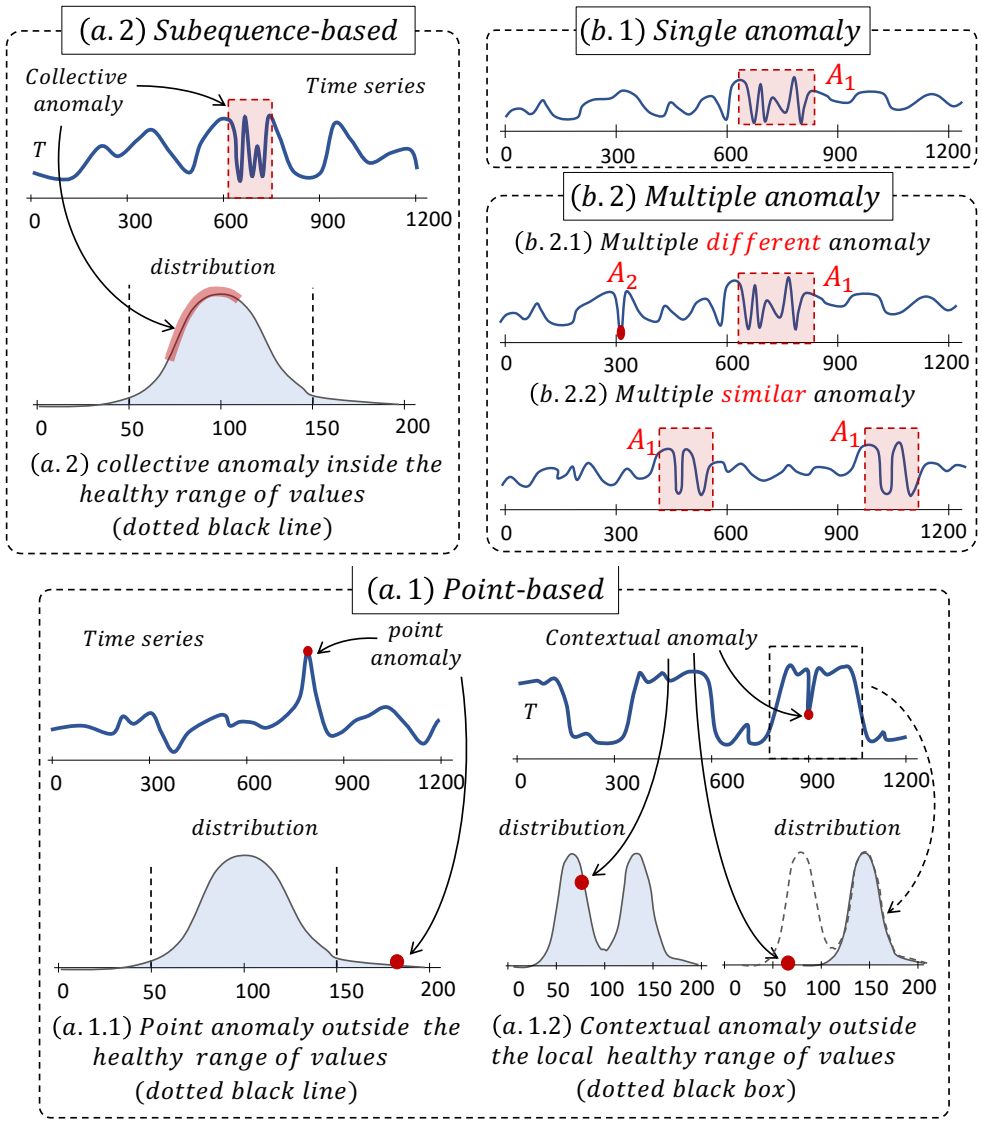


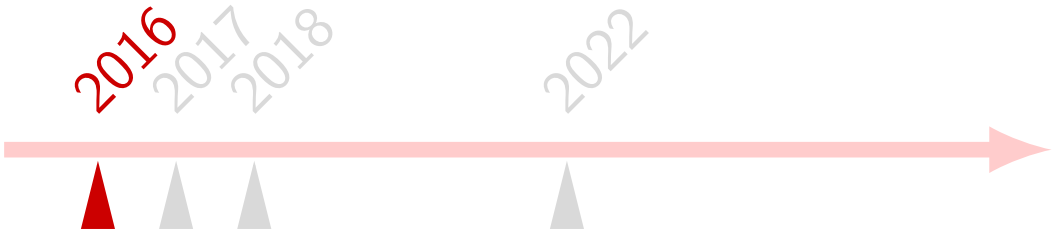
图 2: New Trends in Time-Series Anomaly Detection[4]

TSAD: 算法/模型

A Comprehensive Evaluation on TSAD[15]

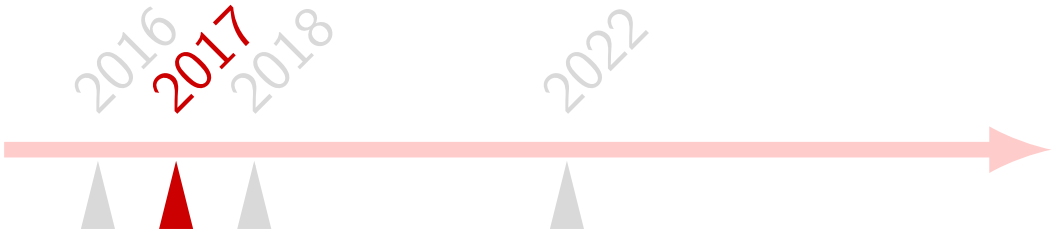
1. Deep learning approaches are not (yet) competitive despite their higher processing effort on trainning data
2. There is no one-size-fits-all solution in the set of currently available algorithms...there is no clear winner
3. Simple methods yield performance almost as good as more sophisticated methods
4. Every practical algorithm deployment needs careful testing
5. Anomalies on periodic time series are easier to detect than on non-periodic time series

TSAD: 算法/模型



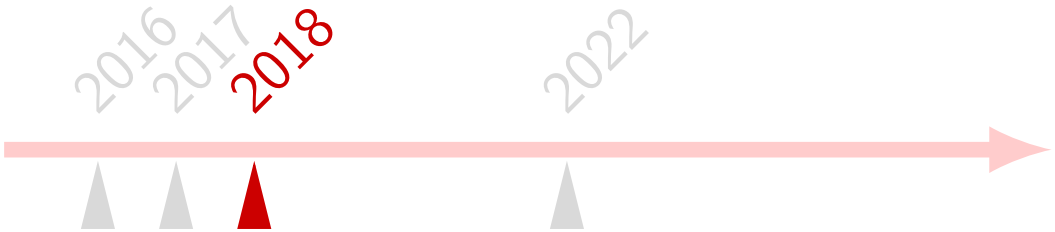
- 2016: Amazon: RRCF[2]

TSAD: 算法/模型



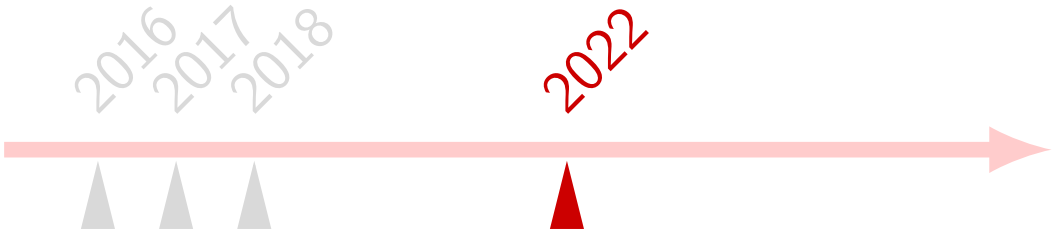
- 2016: Amazon: RRCF[2]
- 2017: Twitter: H-S-ESD[9]
- 2017: IRISA: SPOT[16]

TSAD: 算法/模型



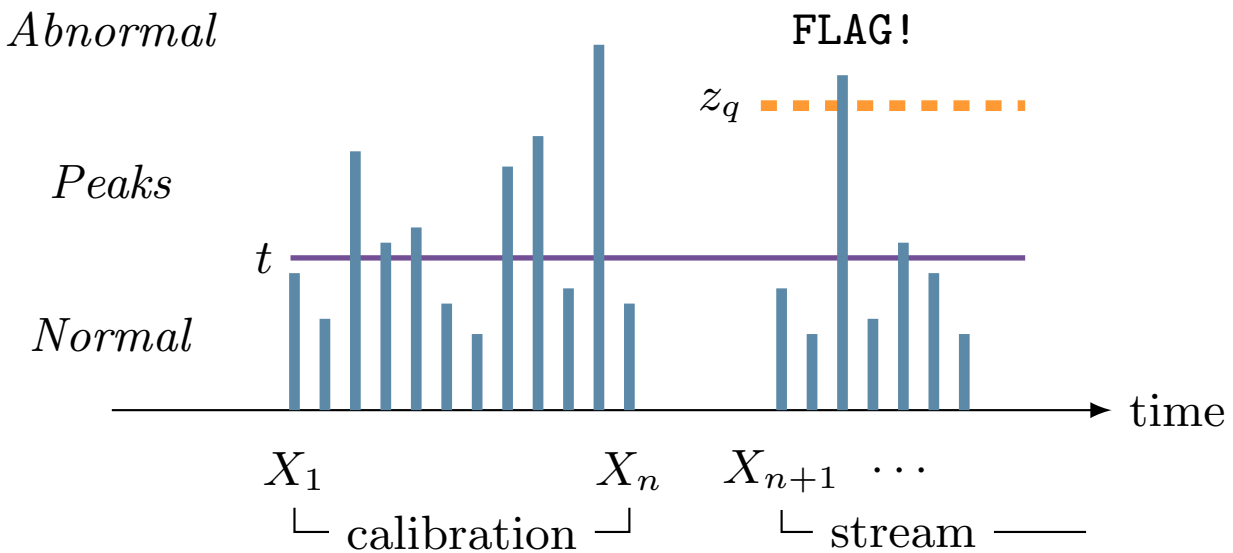
- 2016: Amazon: RRCF[2]
- 2017: Twitter: H-S-ESD[9]
- 2017: IRISA: SPOT[16]
- 2018: Facebook(Meta): Prophet[17]

TSAD: 算法/模型



- 2016: Amazon: RRCF[2]
- 2017: Twitter: H-S-ESD[9]
- 2017: IRISA: SPOT[16]
- 2018: Facebook(Meta): Prophet[17]
- 2021: Huawei: FluxEV[12]

TSAD: 算法/模型



Algorithm 1 POT (Peaks-over-Threshold)

procedure POT($X_1, \ldots X_n, q$)
    $t \leftarrow$ SetInitialThreshold($X_1, \ldots X_n$)
    $\mathbf{Y}_t \leftarrow \{X_i - t \,|\, X_i > t\}$
    $\hat{\gamma}, \hat{\sigma} \leftarrow$ Grimshaw($\mathbf{Y}_t$)
    $z_q \leftarrow$ CalcThreshold($q, \hat{\gamma}, \hat{\sigma}, n, N_t, t$)
    return z_q, t
end procedure

图 3: Anomaly detection in streams with extreme value theory[16]

TSAD: 算法/模型

Algorithm 2 SPOT (Streaming POT)

procedure SPOT($(X_i)_{i>0}, n, q$)
  $\mathbf{A} \leftarrow \emptyset$ ▷ set of the anomalies
  $z_q, t \leftarrow \mathrm{POT}(X_1, \ldots X_n, q)$
  $k \leftarrow n$
  for $i > n$ **do**
    if $X_i > z_q$ **then** ▷ anomaly case
      Add (i, X_i) in $\mathbf{A}$
    else if $X_i > t$ **then** ▷ real peak case
      $Y_i \leftarrow X_i - t$
      Add Y_i in $\mathbf{Y}_t$
      $N_t \leftarrow N_t + 1$
      $k \leftarrow k + 1$
      $\hat{\gamma}, \hat{\sigma} \leftarrow \textsc{Grimshaw}(\mathbf{Y}_t)$
      $z_q \leftarrow \textsc{CalcThreshold}(q, \hat{\gamma}, \hat{\sigma}, k, N_t, t)$
    else ▷ normal case
      $k \leftarrow k + 1$
    end if
  end for
end procedure

图 4: IRISA: SPOT[16]

TSAD: 算法/模型

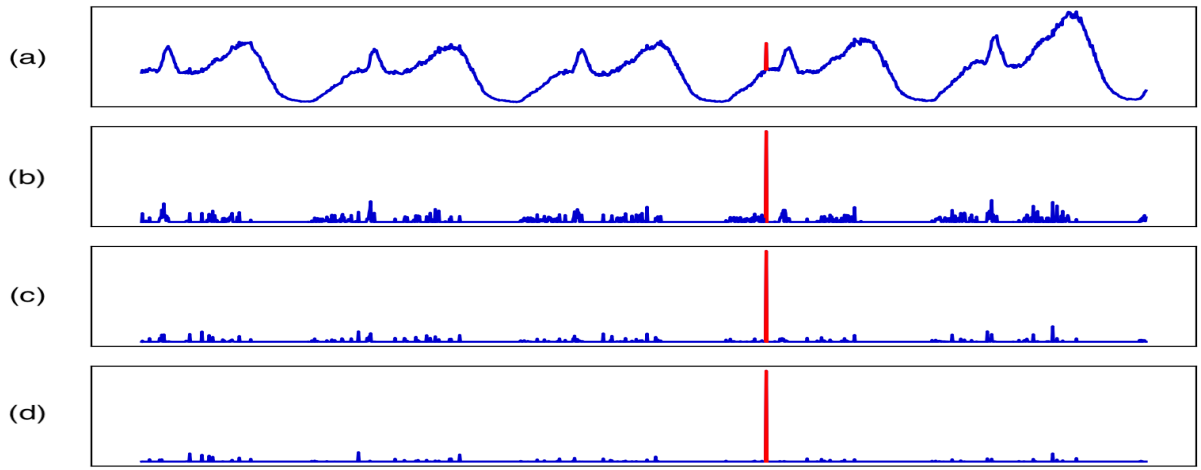


Figure 6: Smoothing. The first row (a) shows the raw time-series curves and anomalies are highlighted in red; (b) stands for originally extracted fluctuations; (c)-(d) represent the fluctuation features after the first and second smoothing respectively.

图 5: Huawei: FluxEV[12]

TSAD: 算法/模型

Algorithm 1: ExtAndSmooth

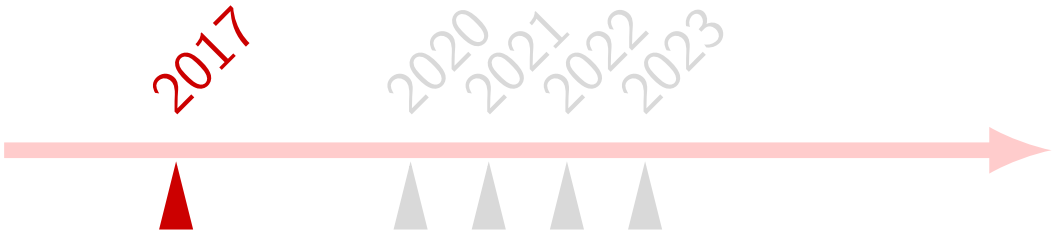
Input: input data $\boldsymbol{X} = [X_1, .., X_n]$, window sizes s, p, d, period l

Output: $\boldsymbol{E} = [E_1, .., E_n]$, $\boldsymbol{F} = [F_1, .., F_n]$, $\boldsymbol{S} = [S_1, .., S_n]$, $\boldsymbol{M} = [M_1, .., M_{n-d}]$

for $i = 1$ *to* n **do**
- $E_i = F_i = M_i = S_i = None$;
- **if** $i > s$ **then**
  - $E_i = X_i - EWMA(\boldsymbol{X}_{i-s,i-1})$;
- **if** $i > 2s$ **then**
  - $\Delta\sigma = \sigma(\boldsymbol{E}_{i-s,i}) - \sigma(\boldsymbol{E}_{i-s,i-1})$;
  - $F_i = \max(\Delta\sigma, 0)$;
- **if** $i > 2s + 2d$ **then**
  - $M_{i-d} = \max(\boldsymbol{F}_{i-2d,i})$;
- **if** $i > 2s + d + l(p-1)$ **then**
  - $\Delta F_i = F_i - \max\left(M_{i-l(p-1)}, .., M_{i-2l}, M_{i-l}\right)$;
  - $S_i = \max(\Delta F_i, 0)$;

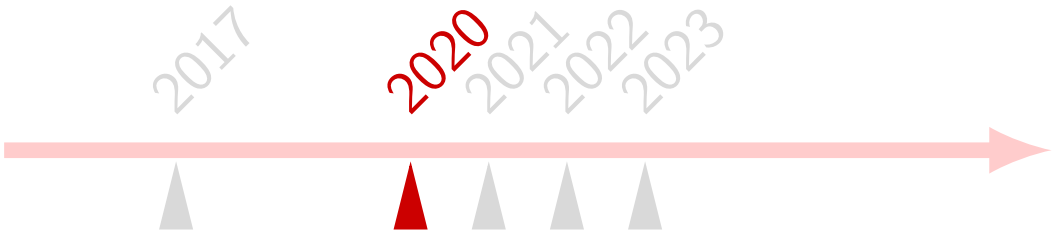
图 6: FlHuawei: FluxEVuxEV[12]

TSAD-System: 工业界开源系统



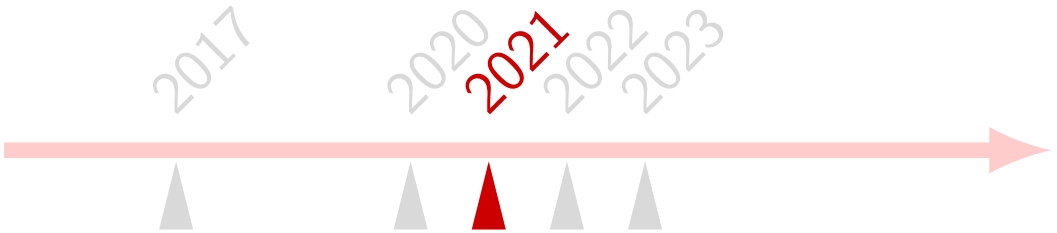
- 2017: Twitter: H-S-ESD[9]

TSAD-System: 工业界开源系统



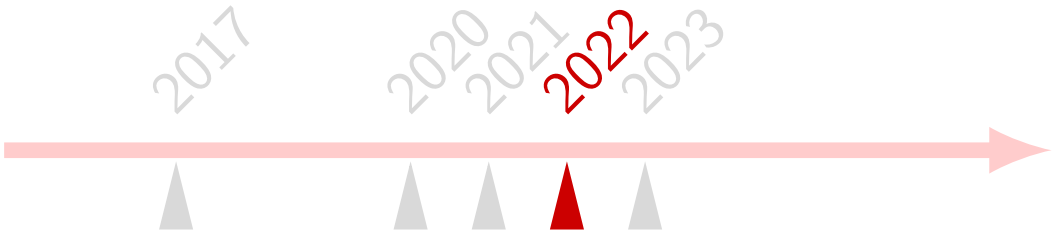
- 2017: Twitter: H-S-ESD[9]
- 2020: Alibaba(DAMO): RobustX[19][20][7][21]
- 2020: Amazon: GluonTS[1]
- 2020: Zillow: Luminaire[5]
- 2020: MicroSoft: Auto-Selector[22]

TSAD-System: 工业界开源系统



- 2017: Twitter: H-S-ESD[9]
- 2020: Alibaba(DAMO): RobustX[19][20][7][21]
- 2020: Amazon: GluonTS[1]
- 2020: Zillow: Luminaire[5]
- 2020: MicroSoft: Auto-Selector[22]
- 2021: Salesforce: Merlion[3]
- 2021: Linkedin: Silverkite[10]

TSAD-System: 工业界开源系统

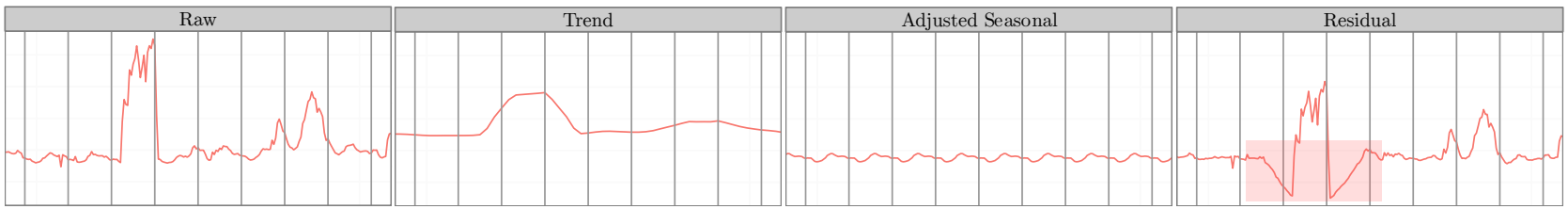


- 2017: Twitter: H-S-ESD[9]
- 2020: Alibaba(DAMO): RobustX[19][20][7][21]
- 2020: Amazon: GluonTS[1]
- 2020: Zillow: Luminaire[5]
- 2020: MicroSoft: Auto-Selector[22]
- 2021: Salesforce: Merlion[3]
- 2021: Linkedin: Silverkite[10]
- 2022: IBM: AnomalyKiTS[14]
- 2022: MicroSoft: HEAT-RL[18]

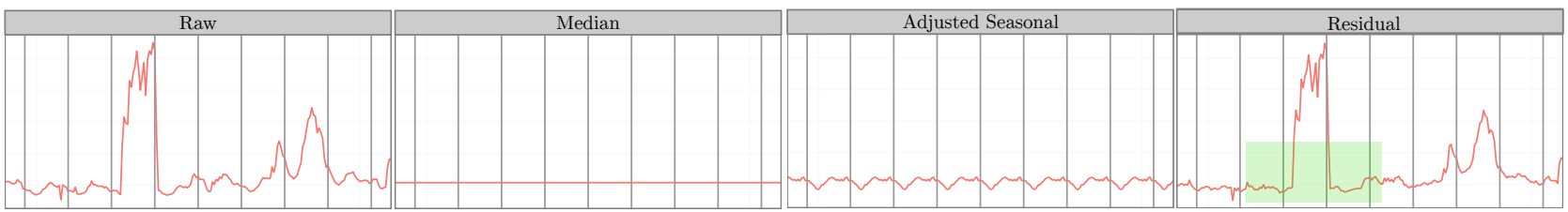
TSAD-System: 工业界开源系统

- 2017: Twitter: H-S-ESD[9]
- 2020: Alibaba(DAMO): RobustX[19][20][7][21]
- 2020: Amazon: GluonTS[1]
- 2020: Zillow: Luminaire[5]
- 2020: MicroSoft: Auto-Selector[22]
- 2021: Salesforce: Merlion[3]
- 2021: Linkedin: Silverkite[10]
- 2022: IBM: AnomalyKiTS[14]
- 2022: MicroSoft: HEAT-RL[18]
- 2023: Amazon: Model Selection[8]

TSAD-System: H-S-ESD(Twitter, 2017)



(a) STL with Trend Removal



(b) STL with Median Removal

图 7: Twitter: ESD[9]

TSAD-System: H-S-ESD(Twitter, 2017)

Algorithm 1 S-ESD Algorithm

Input:

X = A time series

n = number of observations in X

k = max anomalies (iterations in ESD)

Output:

X_A = An anomaly vector wherein each element is a tuple $(timestamp, observed\ value)$

Require:

$k \leq (n \times .49)$

1. Extract seasonal component S_X using STL Variant
2. Compute median $\tilde{X}$

/* Compute residual */

3. $R_X = X - S_X - \tilde{X}$

/* Detect anomalies vector X_A using ESD */

4. X_A = ESD(R, k)

return X_A

图 8: Twitter: ESD[9]

TSAD-System: RobustX(Alibaba, 2020)

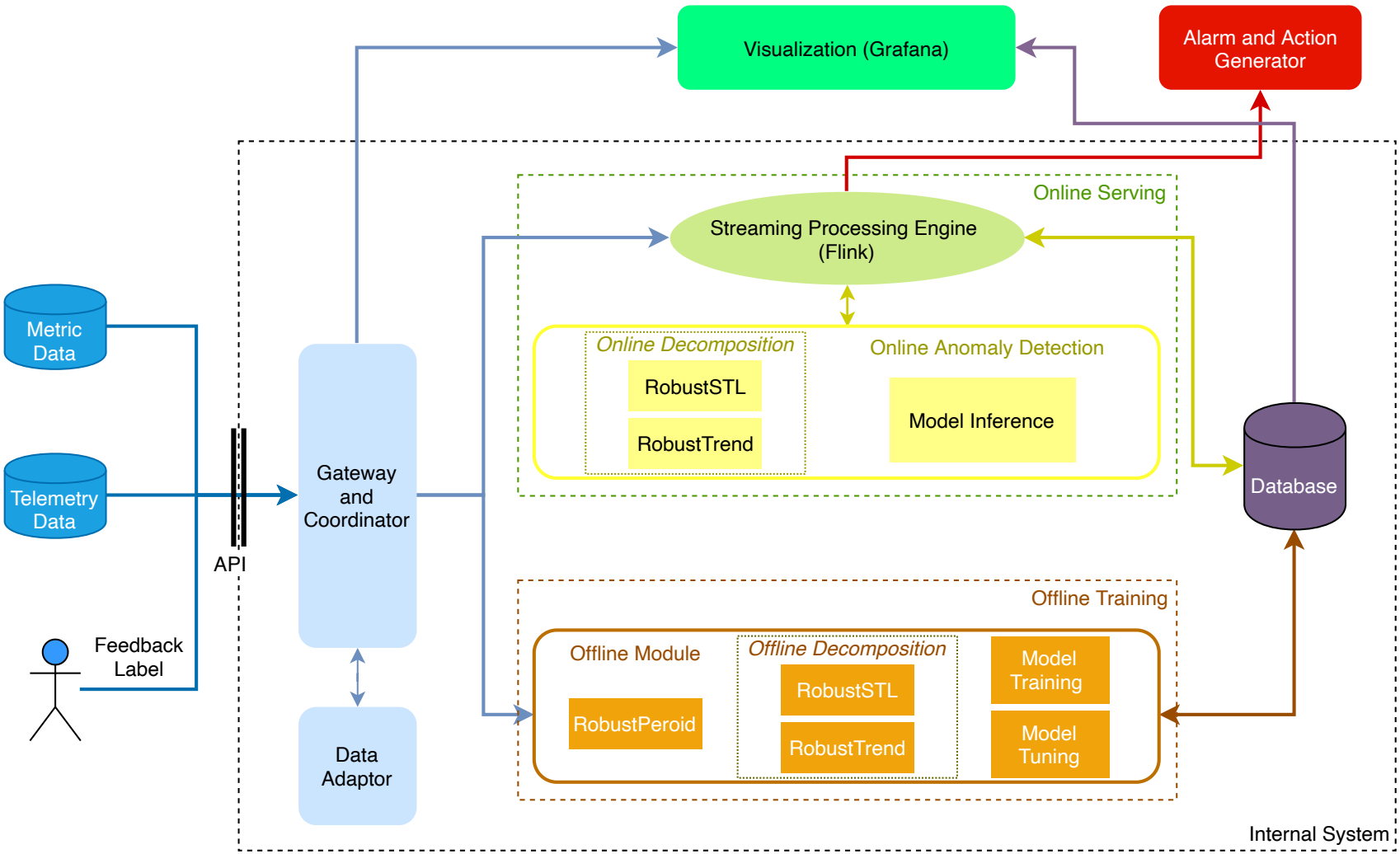


图 9: Alibaba: RobustTAD[7]

TSAD-System: Merlion(Salesforce, 2021)

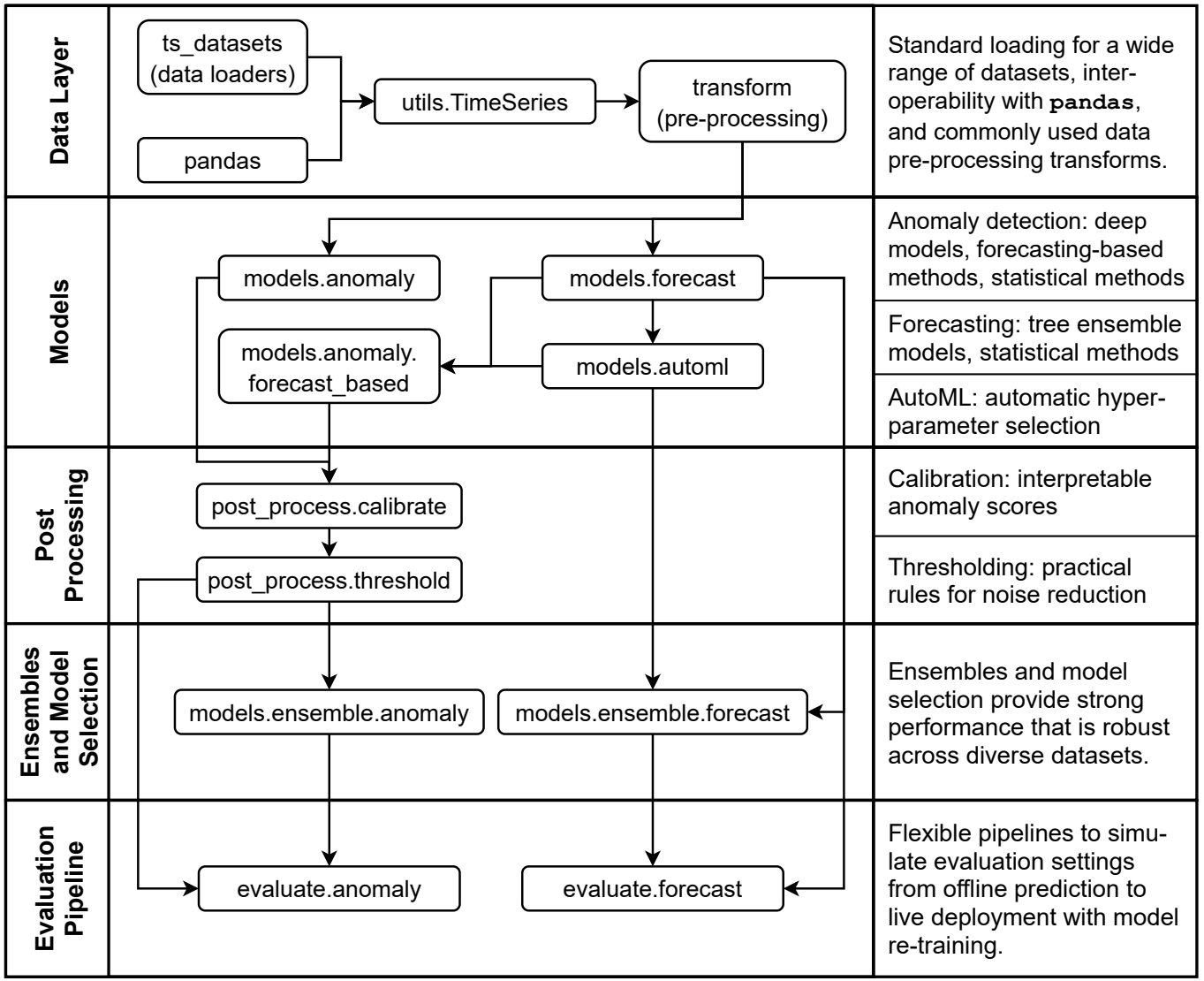
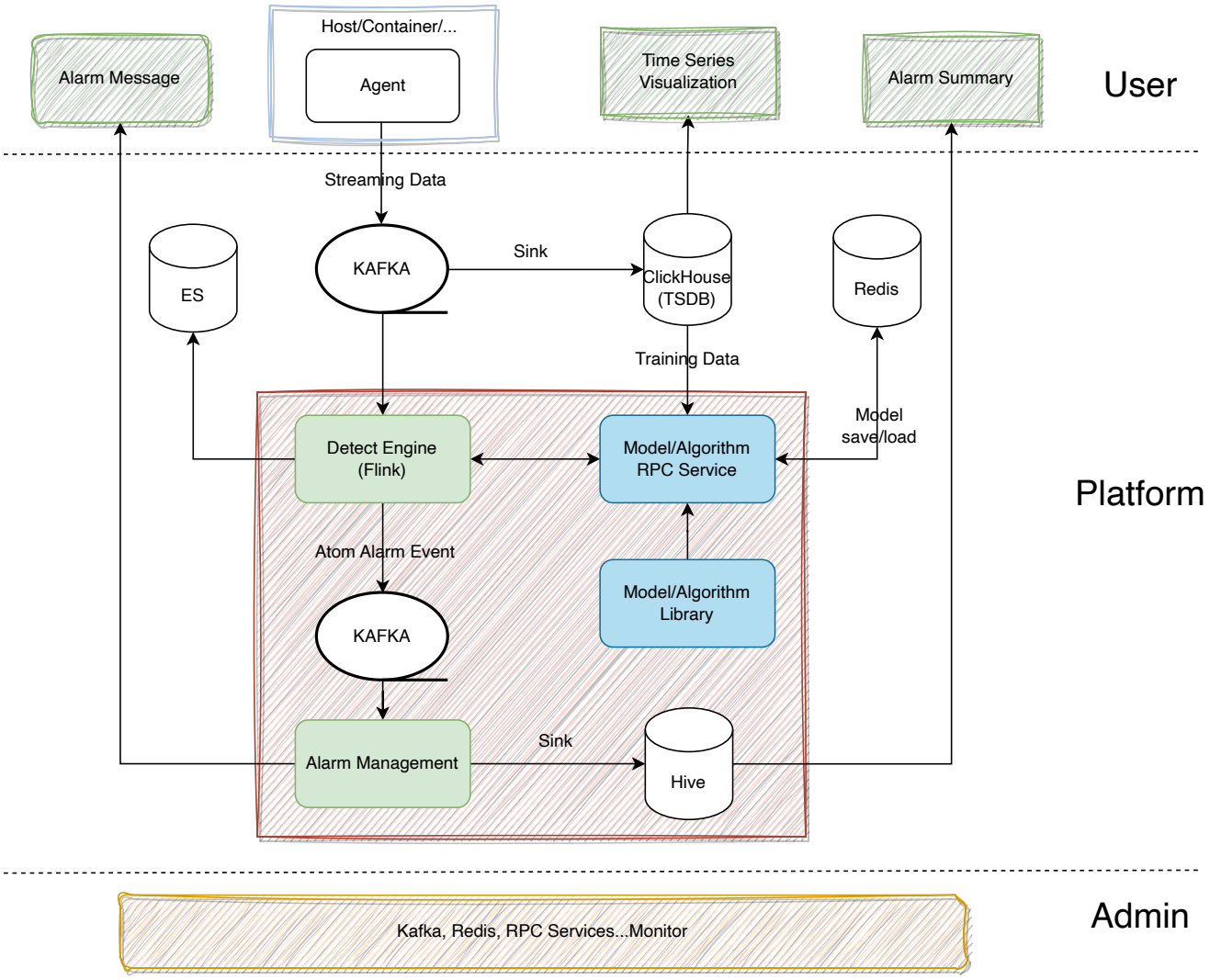


图 10: Salesforce: Merlion[3]

TSAD: 系统架构


Alarm Message
Host/Container/...
Agent
Time Series Visualization
Alarm Summary
User
Streaming Data
ES
KAFKA
Sink
ClickHouse (TSDB)
Redis
Training Data
Model save/load
Detect Engine (Flink)
Model/Algorithm RPC Service
Atom Alarm Event
Model/Algorithm Library
KAFKA
Alarm Management
Sink
Hive
Platform
Kafka, Redis, RPC Services...Monitor
Admin

TSAD: 系统性能问题与解决

Redis: 模型大 Key, 无法保证 SLA

优化算法实现, 降低模型内用到 10KB 以下

TSAD: 系统性能问题与解决

Redis: 模型大 Key, 无法保证 SLA

优化算法实现, 降低模型内用到 10KB 以下

Flink: 数据乱序

`SlidingEventTimeWindows`, 修改算法使能接受一段时间序列的输入，并记录时间状态

TSAD: 系统性能问题与解决

Redis: 模型大 Key, 无法保证 SLA

优化算法实现, 降低模型内用到 10KB 以下

Flink: 数据乱序

`SlidingEventTimeWindows`, 修改算法使能接受一段时间序列的输入，并记录时间状态

RPC: 序列化与反序列化开销过高

通过火焰图发现在进行 RPC 通讯时, Python 对象在序列化和反序列化时耗时较多

RCA: 根因分析

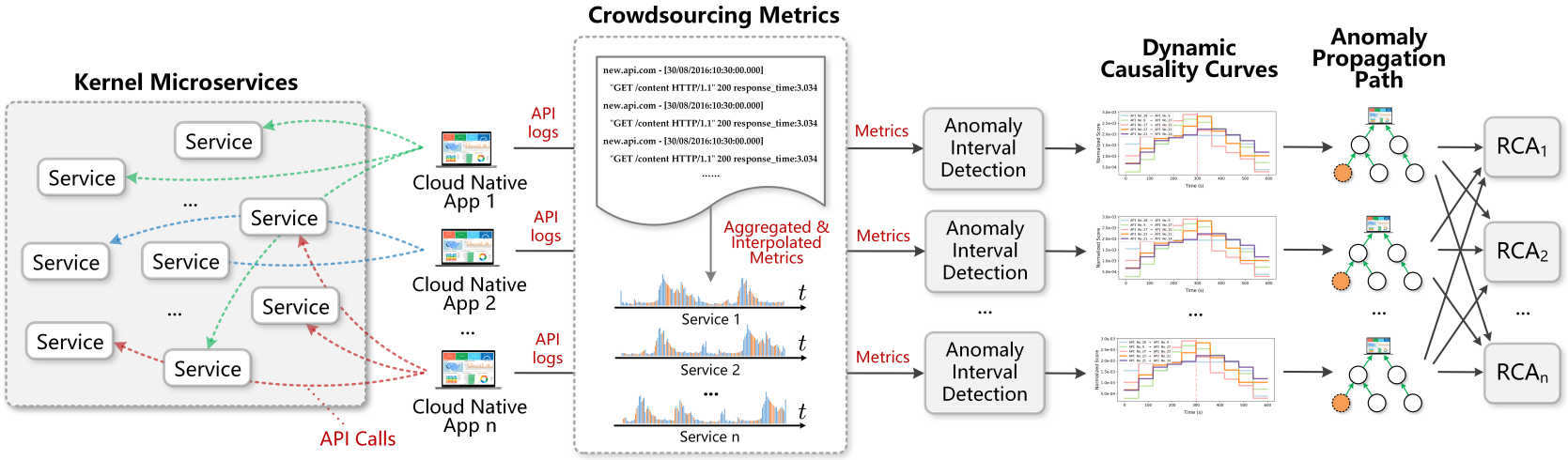


图 12: PKU: DyCause[13][24]

AI4X: 概览

内存故障预测

通过内存条中 CE 的模式**提前预测** UE 的发生，进而降低宕机率

硬盘故障预测

通过硬盘 SMART Log 数据**提前预测**硬盘故障，避免数据损坏

数据中心智能控制

结合更多业务侧的数据，升级数据中心控制系统

AI4X: 内存故障预测

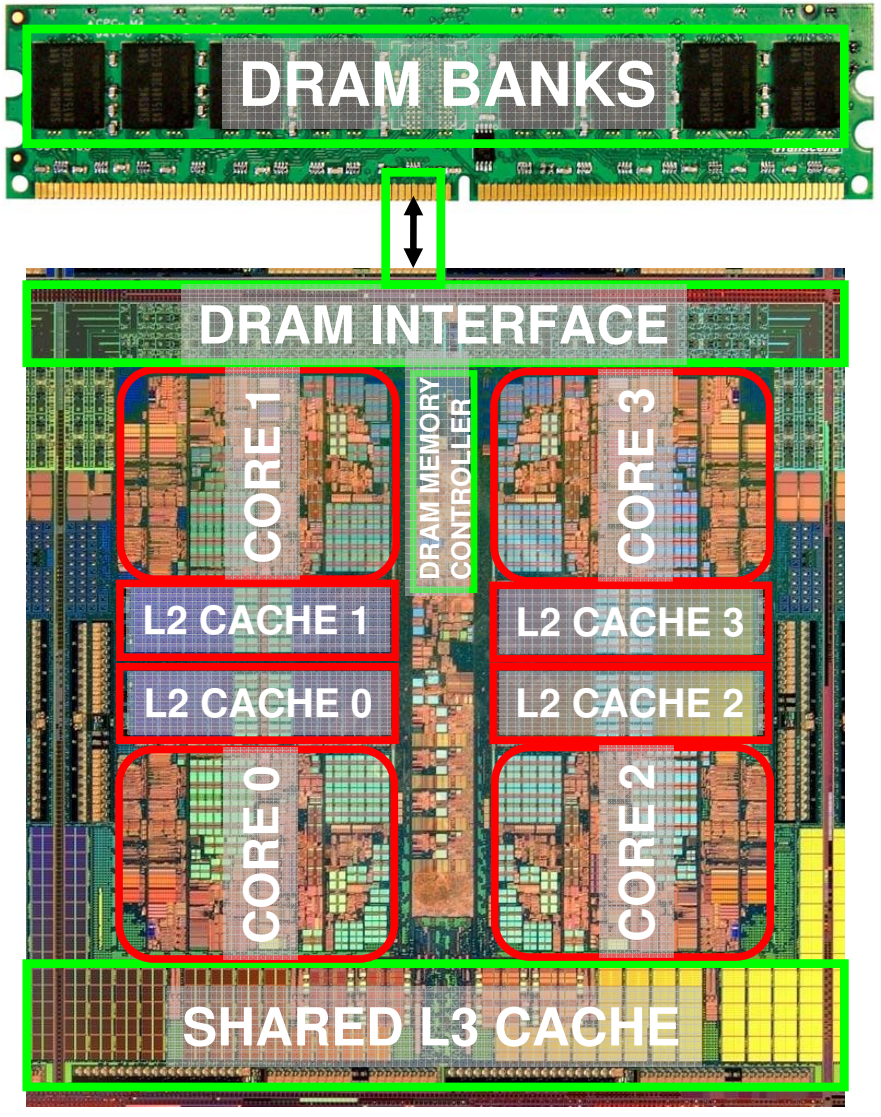


图 13: From CMU ECE740

AI4X: 内存故障预测

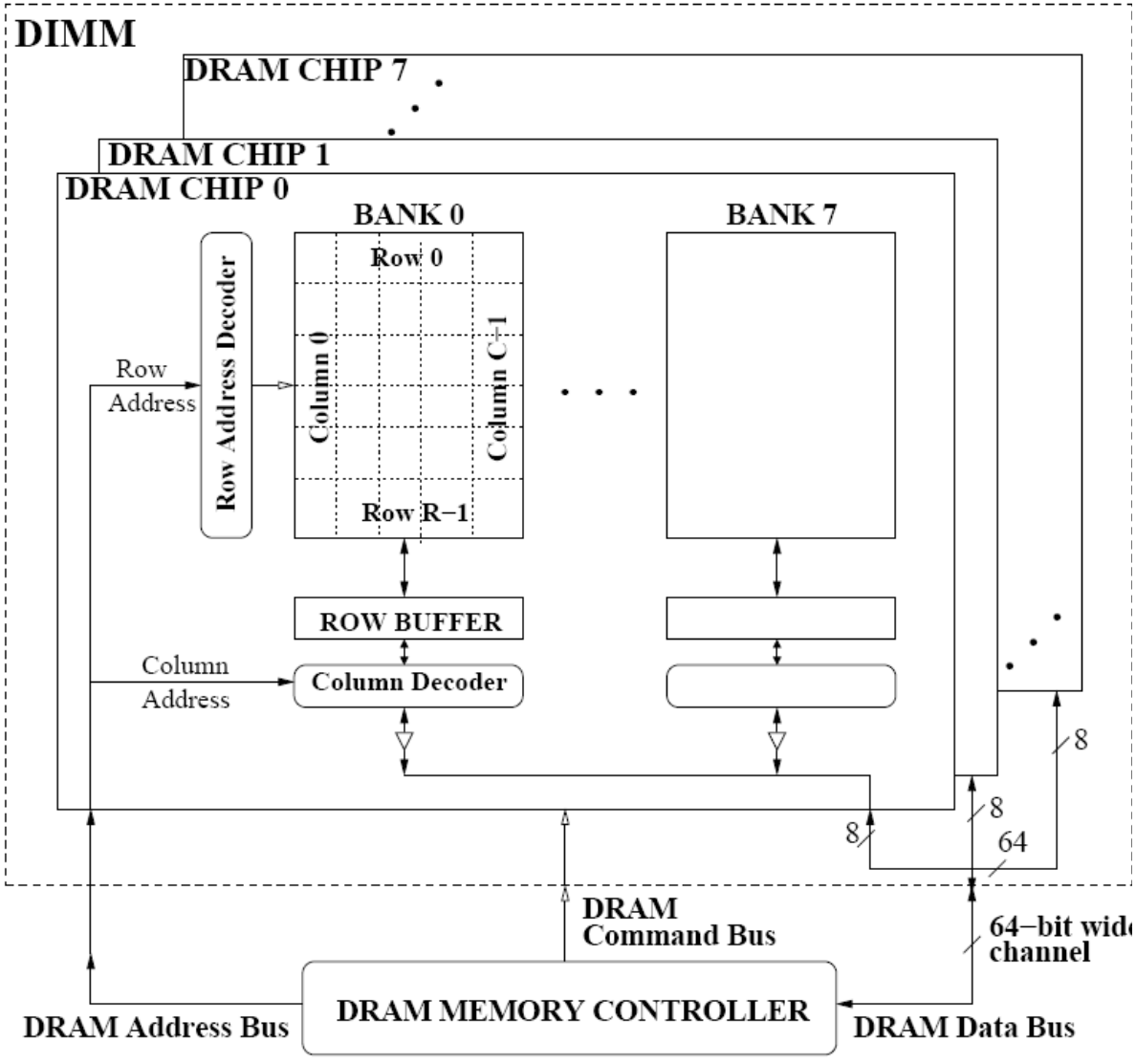


图 14: From CMU ECE740

AI4X: 内存故障预测

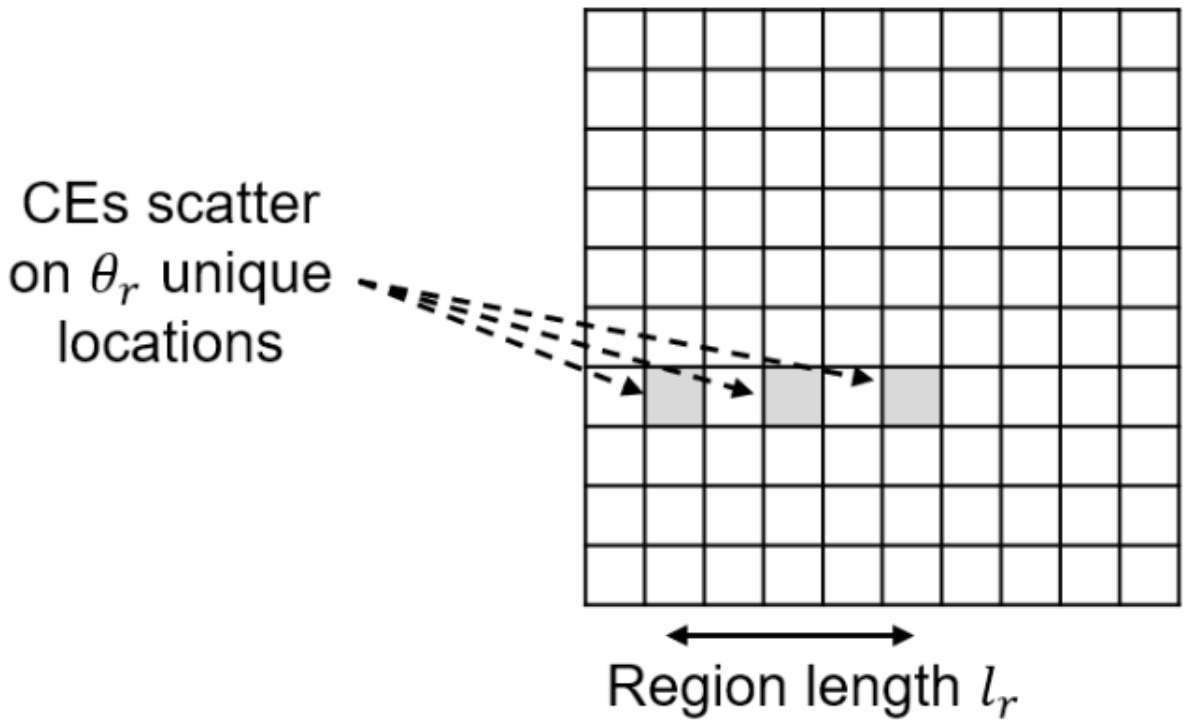


图 15: Intel, Bytedance: MFP[6]

AI4X: 硬盘故障预测

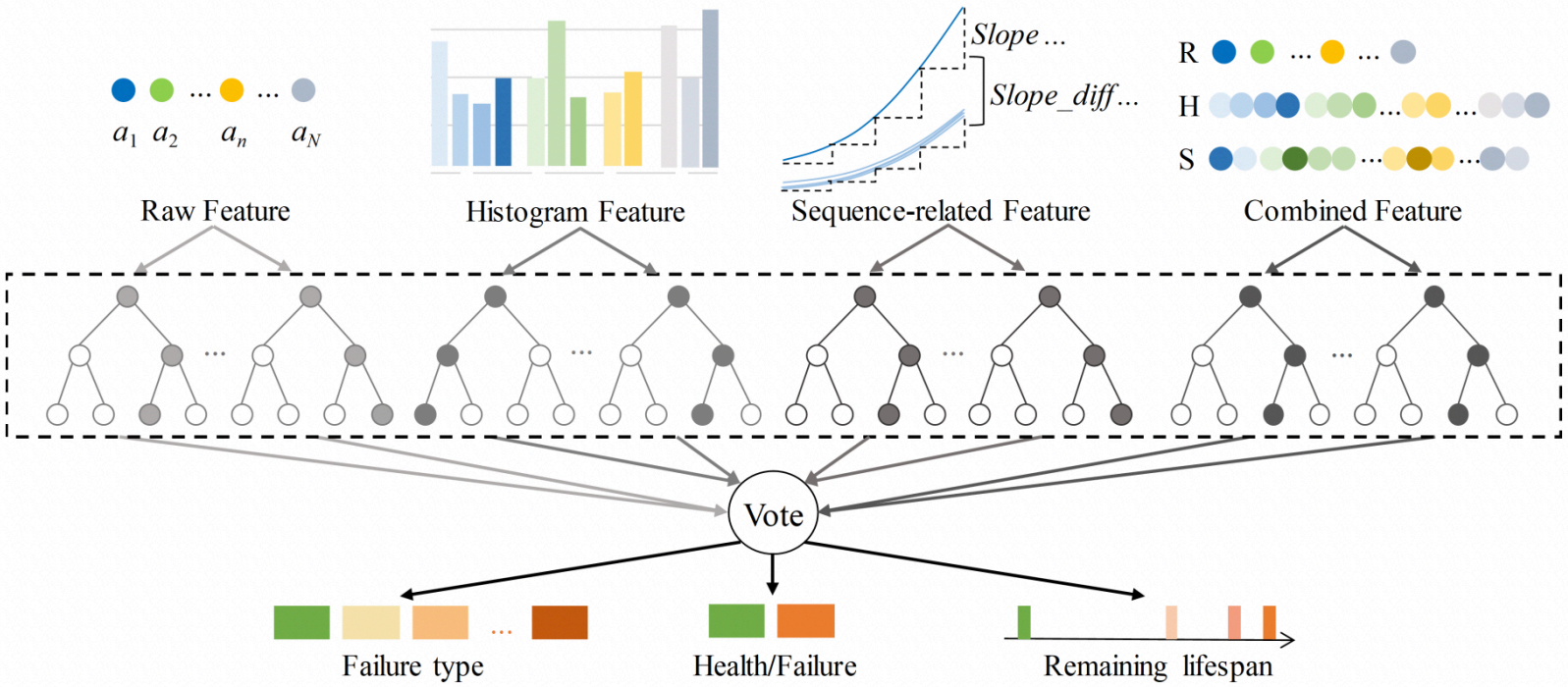


图 16: Samsung, Tencent: DFP(FAST'23)[23]

AI4X: 数据中心智能控制

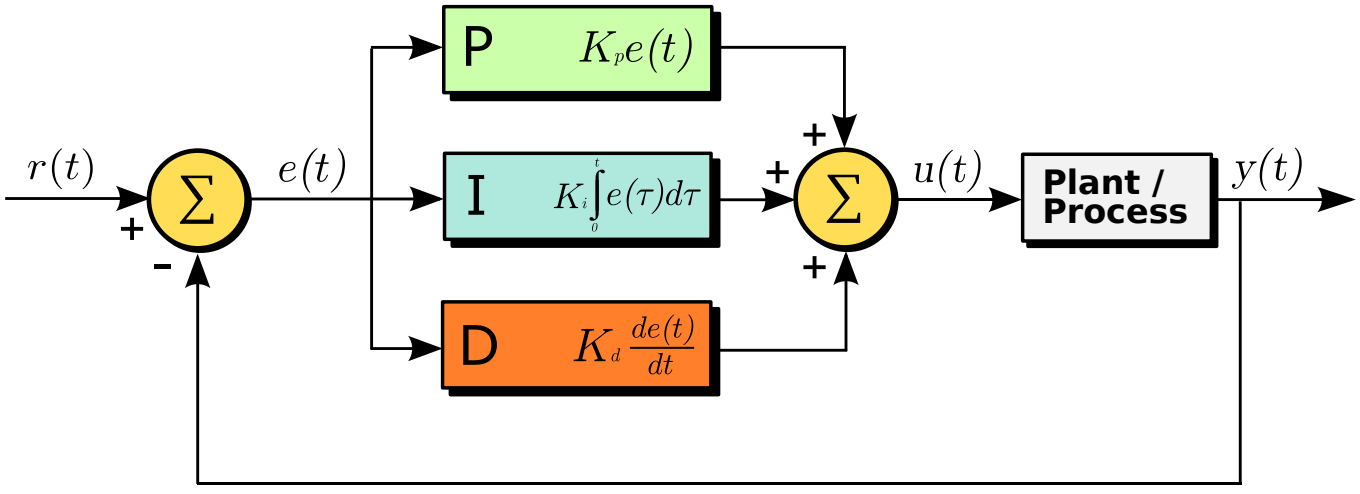


图 17: Wikipedia: PID controller[2]

[2] PID controller

AI4X: 数据中心智能控制

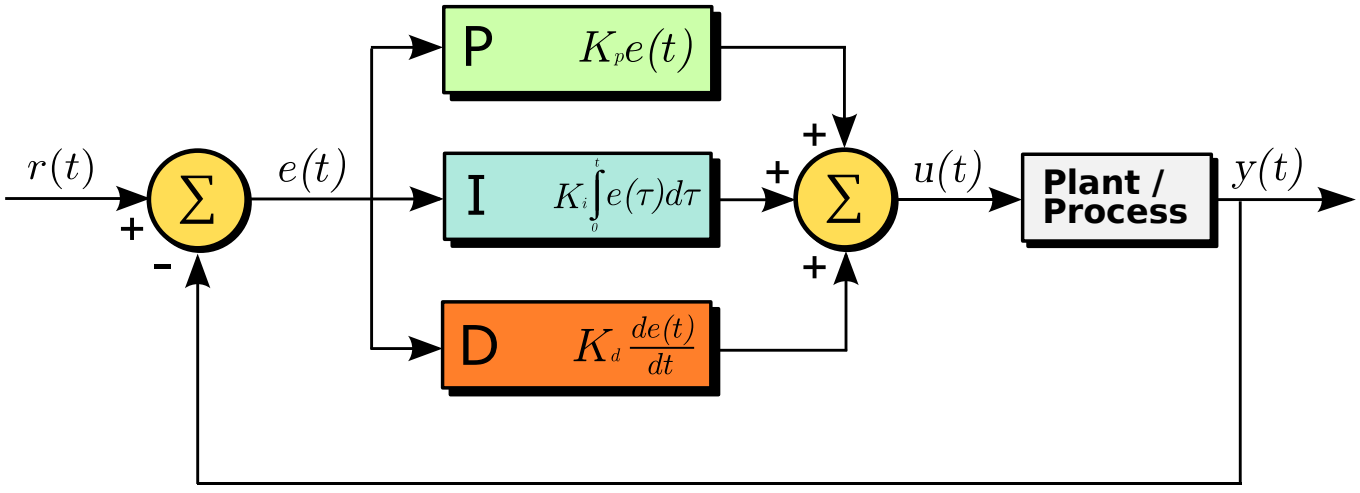


图 17: Wikipedia: PID controller[2]

PID -> MPC

机房内的机器的负荷预测可以提供很有利的信息, 借助 MPC 可以充分利用这一信息进行更加精准的温度调控 (字节液冷温控专利 [25])

[2] PID controller

参考文献 I

[1] Alexander Alexandrov et al. “GluonTS: Probabilistic and Neural Time Series Modeling in Python.”. In: *J. Mach. Learn. Res.* 21.116 (2020), pp. 1–6.

[2] Matthew D Bartos, Abhiram Mullapudi, and Sara C Troutman. “rrcf: Implementation of the robust random cut forest algorithm for anomaly detection on streams”. In: *Journal of Open Source Software* 4.35 (2019), p. 1336.

[3] Aadyot Bhatnagar et al. “Merlion: A machine learning library for time series”. In: *arXiv preprint arXiv:2109.09265* (2021).

[4] Paul Boniol, John Paparrizos, and Themis Palpanas. “New Trends in Time-Series Anomaly Detection”. In: (2023).

[5] Sayan Chakraborty et al. “Building an automated and self-aware anomaly detection system”. In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 1465–1475.

参考文献 II

[6] Xiaoming Du et al. “Fault-aware prediction-guided page offlining for uncorrectable memory error prevention”. In: *2021 IEEE 39th International Conference on Computer Design (ICCD)*. IEEE. 2021, pp. 456–463.

[7] Jingkun Gao et al. “Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks”. In: *arXiv preprint arXiv:2002.09545* (2020).

[8] Mononito Goswami et al. “Unsupervised Model Selection for Time-series Anomaly Detection”. In: *arXiv preprint arXiv:2210.01078* (2022).

[9] Jordan Hochenbaum, Owen S Vallis, and Arun Kejariwal. “Automatic anomaly detection in the cloud via statistical learning”. In: *arXiv preprint arXiv:1704.07706* (2017).

[10] Reza Hosseini et al. “A flexible forecasting model for production systems”. In: *arXiv preprint arXiv:2105.01098* (2021).

[11] Haoyuan Li et al. “Pfp: parallel fp-growth for query recommendation”. In: *Proceedings of the 2008 ACM conference on Recommender systems*. 2008, pp. 107–114.

参考文献 III

[12] Jia Li et al. “FluxEV: a fast and effective unsupervised framework for time-series anomaly detection”. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 824–832.

[13] Yicheng Pan et al. “Faster, deeper, easier: crowdsourcing diagnosis of microservice kernel failure from user space”. In: *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 2021, pp. 646–657.

[14] Dhaval Patel et al. “Anomalykits: Anomaly detection toolkit for time series”. In: *AAAI*. AAAI Press. 2022.

[15] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. “Anomaly detection in time series: a comprehensive evaluation”. In: *Proceedings of the VLDB Endowment* 15.9 (2022), pp. 1779–1797.

[16] Alban Siffer et al. “Anomaly detection in streams with extreme value theory”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 1067–1075.

参考文献 IV

[17] Sean J Taylor and Benjamin Letham. "Forecasting at scale". In: *The American Statistician* 72.1 (2018), pp. 37–45.

[18] Yujing Wang et al. "Heat-RL: Online Model Selection for Streaming Time-Series Anomaly Detection". In: *Conference on Lifelong Learning Agents*. PMLR. 2022, pp. 767–777.

[19] Qingsong Wen et al. "RobustPeriod: Time-frequency mining for robust multiple periodicities detection". In: *arXiv preprint arXiv:2002.09535* (2020).

[20] Qingsong Wen et al. "RobustSTL: A robust seasonal-trend decomposition algorithm for long time series". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 5409–5416.

[21] Qingsong Wen et al. "RobustTrend: A Huber loss with a combined first and second order difference regularization for time series trend filtering". In: *arXiv preprint arXiv:1906.03751* (2019).

[22] Yuanxiang Ying et al. "Automated Model Selection for Time-Series Anomaly Detection". In: *arXiv preprint arXiv:2009.04395* (2020).

参考文献 V

[23] Yuqi Zhang et al. “Multi-view Feature-based {SSD} Failure Prediction: What, When, and Why”. In: *21st USENIX Conference on File and Storage Technologies (FAST 23)*. 2023, pp. 409–424.

[24] 张宇 et al. “一种主机故障的检测方法、装置、电子设备及存储介质”. Chinese. Pat. CN116414653A. July 11, 2023.

[25] 陈昱 et al. “用于温度控制的方法、装置、设备和存储介质”. May 2023.

QA