Real-Time Outlier Detection with Dynamic Process Limits

PC 2023

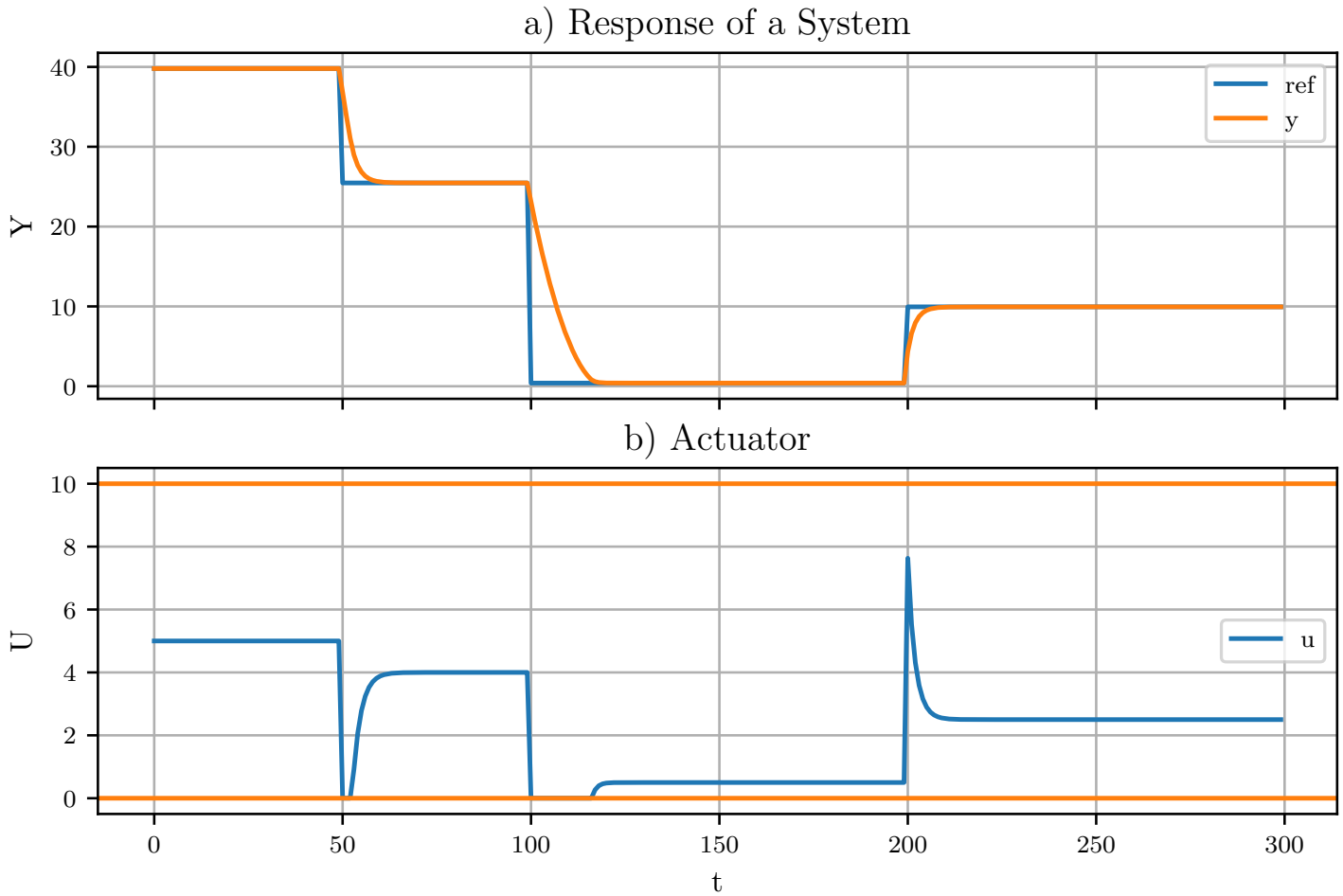
Marek Wadinger[1], Michal Kvasnica[1]

[1]Institute of Information Engineering, Automation, and Mathematics
marek.wadinger@stuba.sk

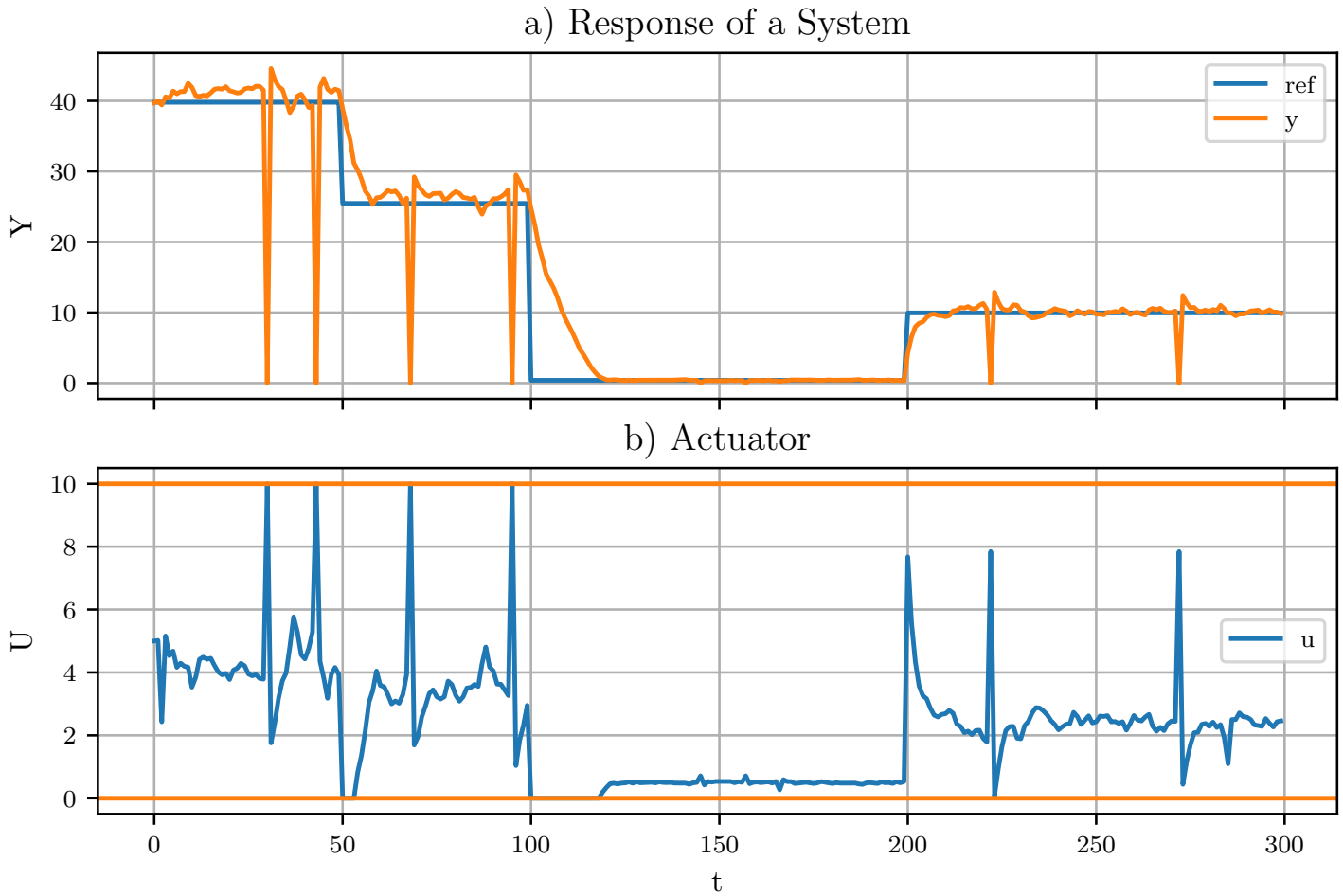
2023-5-18

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

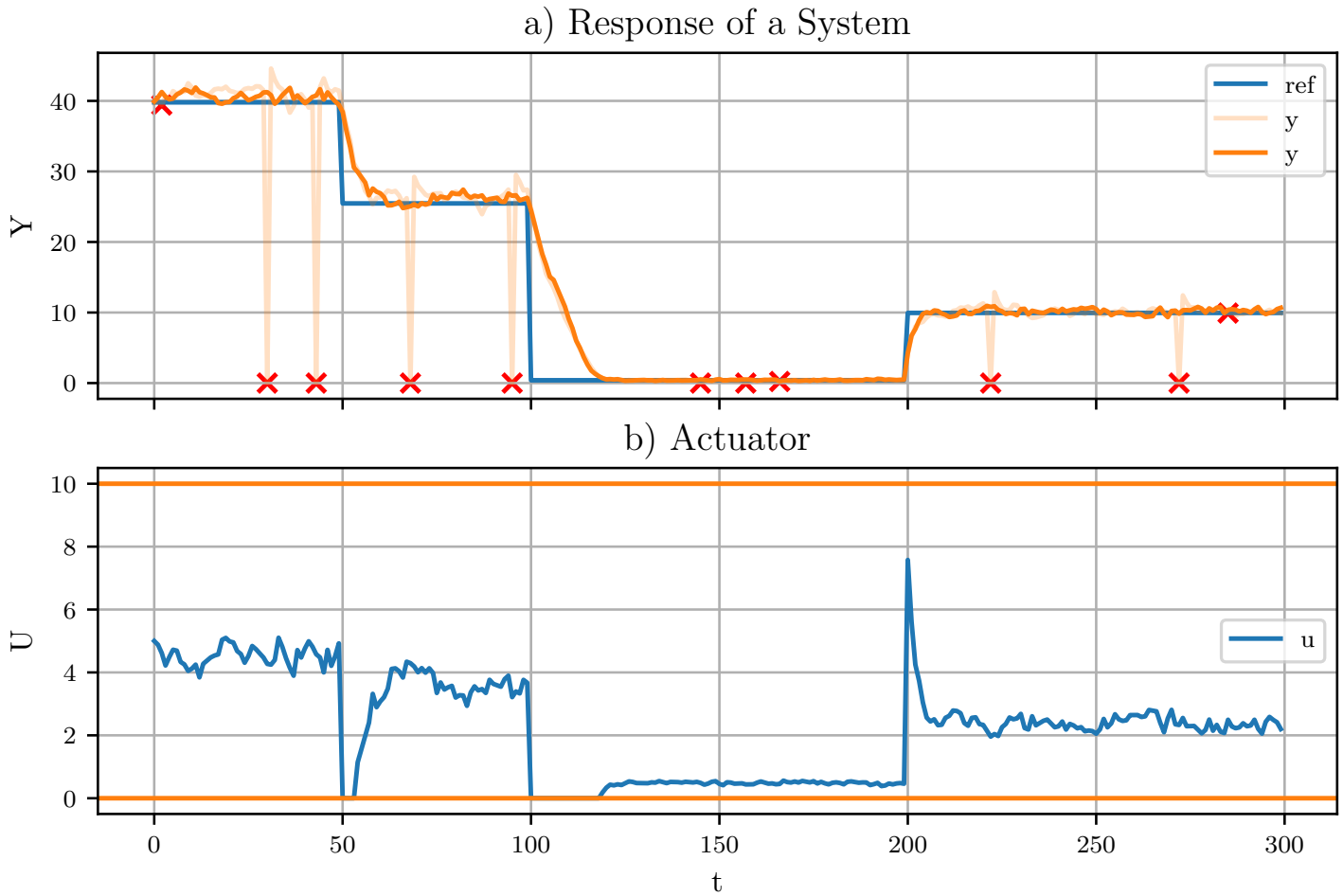
Control Engineering meets Artificial Intelligence



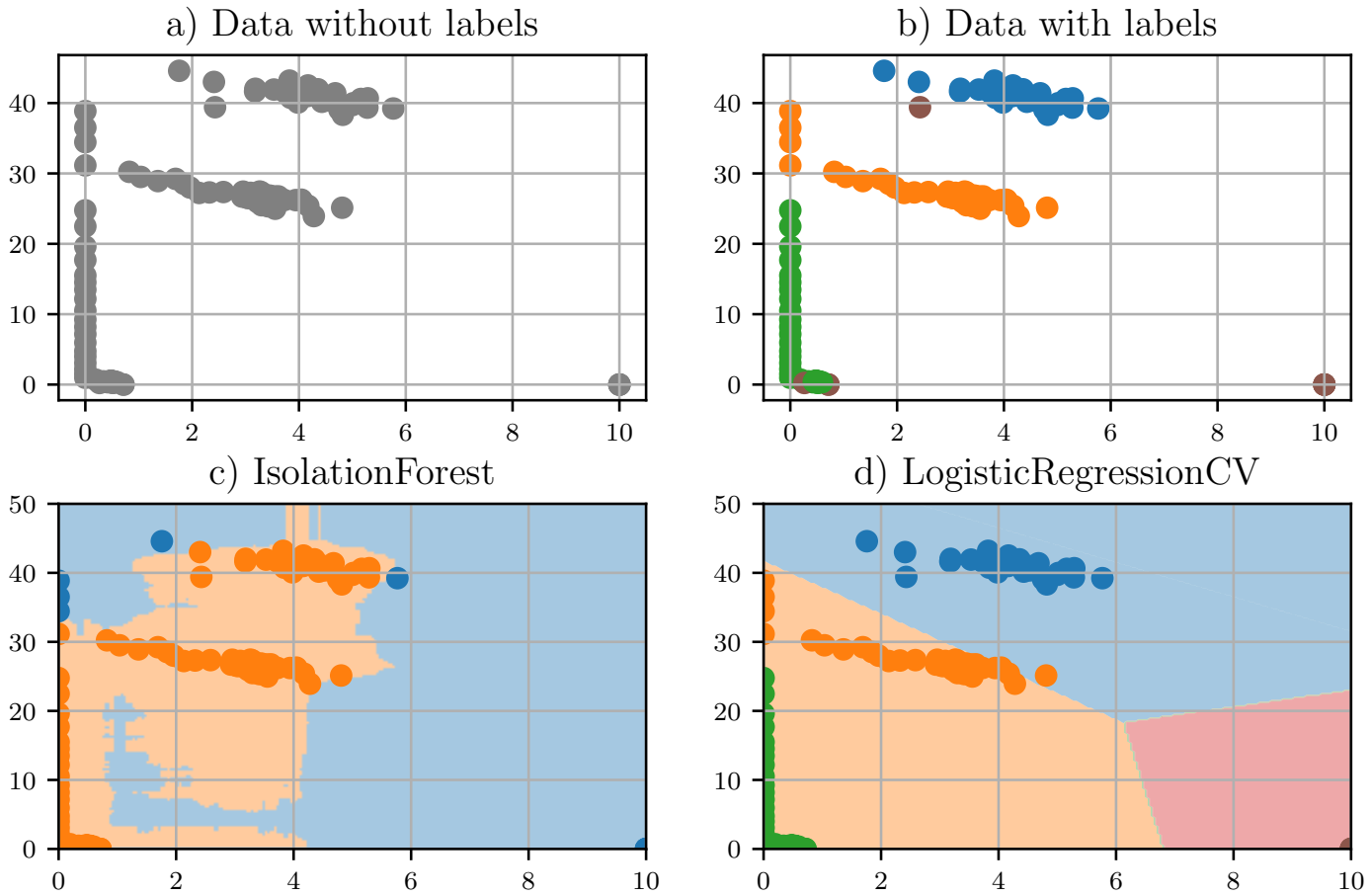
Coming Problem



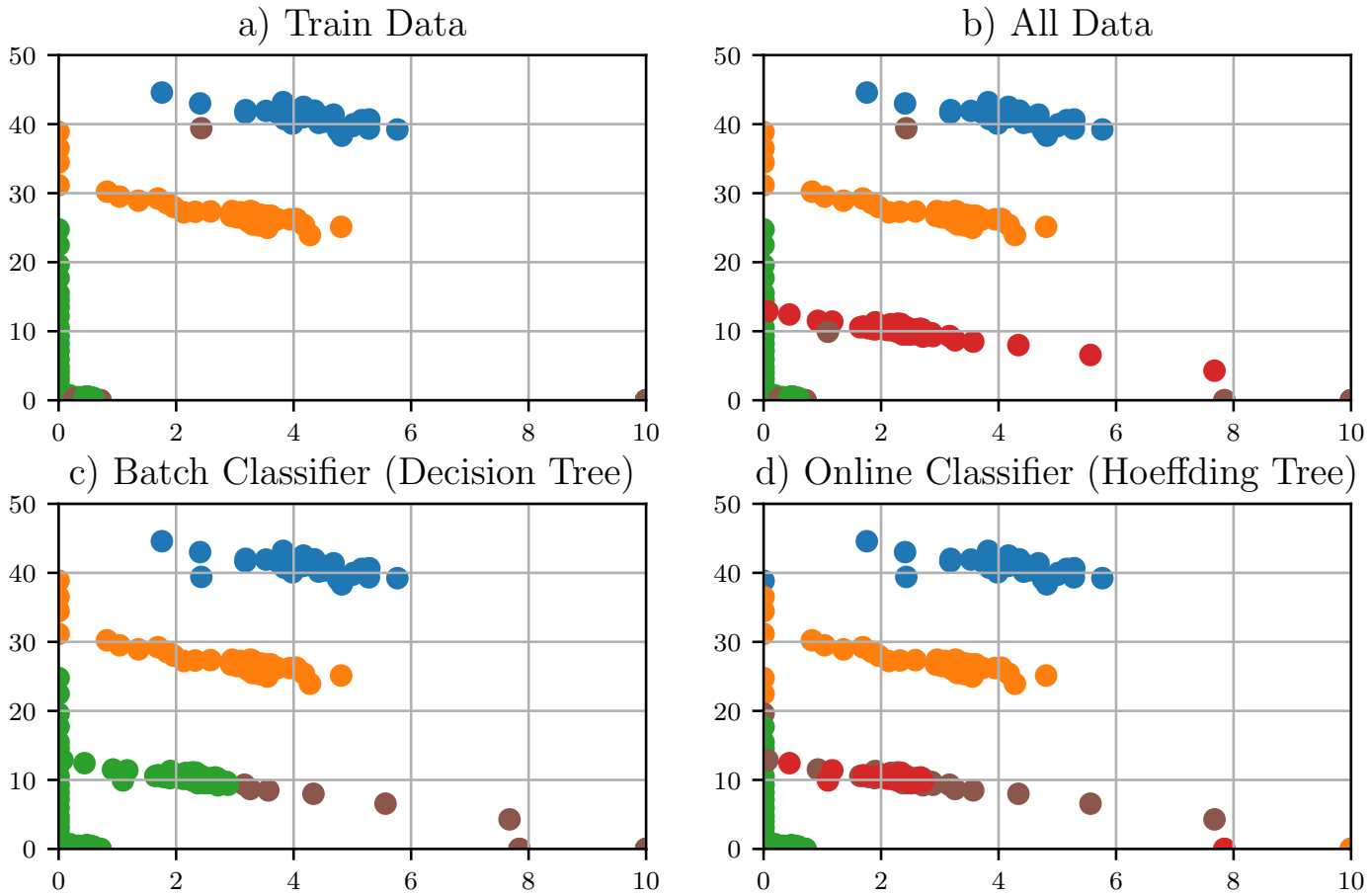
Emotional Win



Drawbacks of Existing Solutions



Drawbacks of Existing Solutions



Audacious Reality

We need to make detector that:

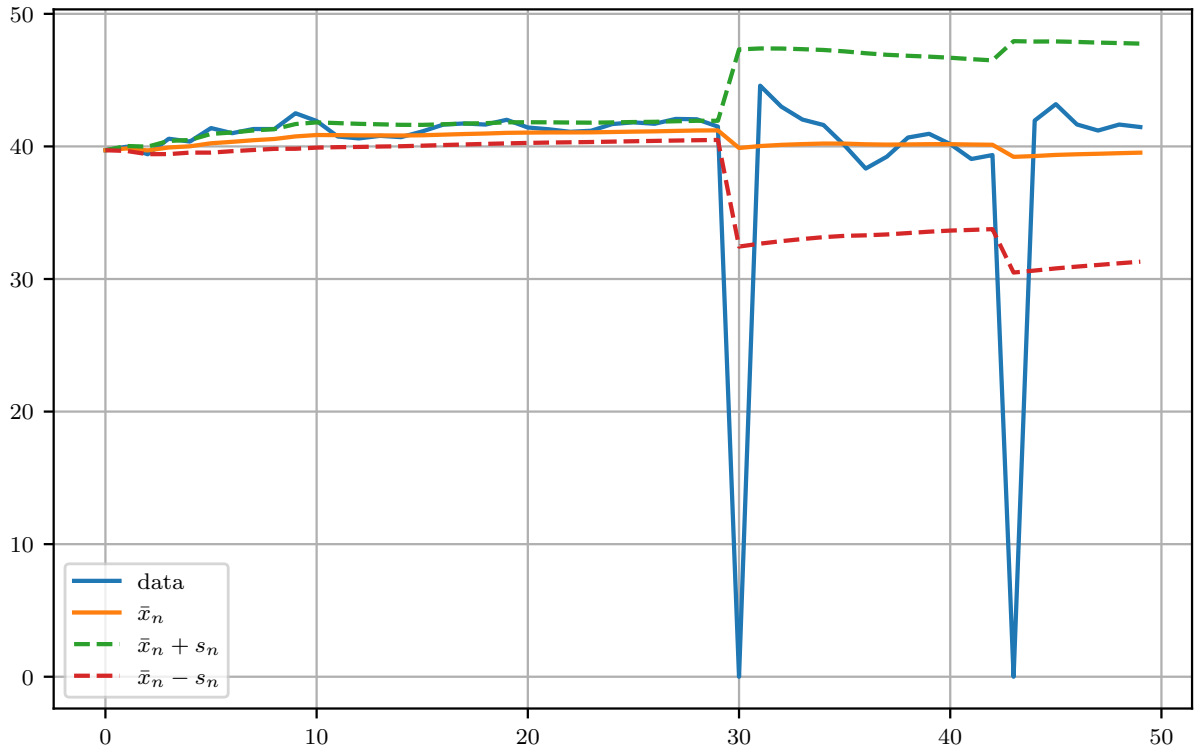
- Does not require huge amount of data
- Adapts to unseen operation
- Offers credible interpretation
- Does not alter operation of existing systems
- Improves maintenance scheduling

Proposed Solution

Combine:

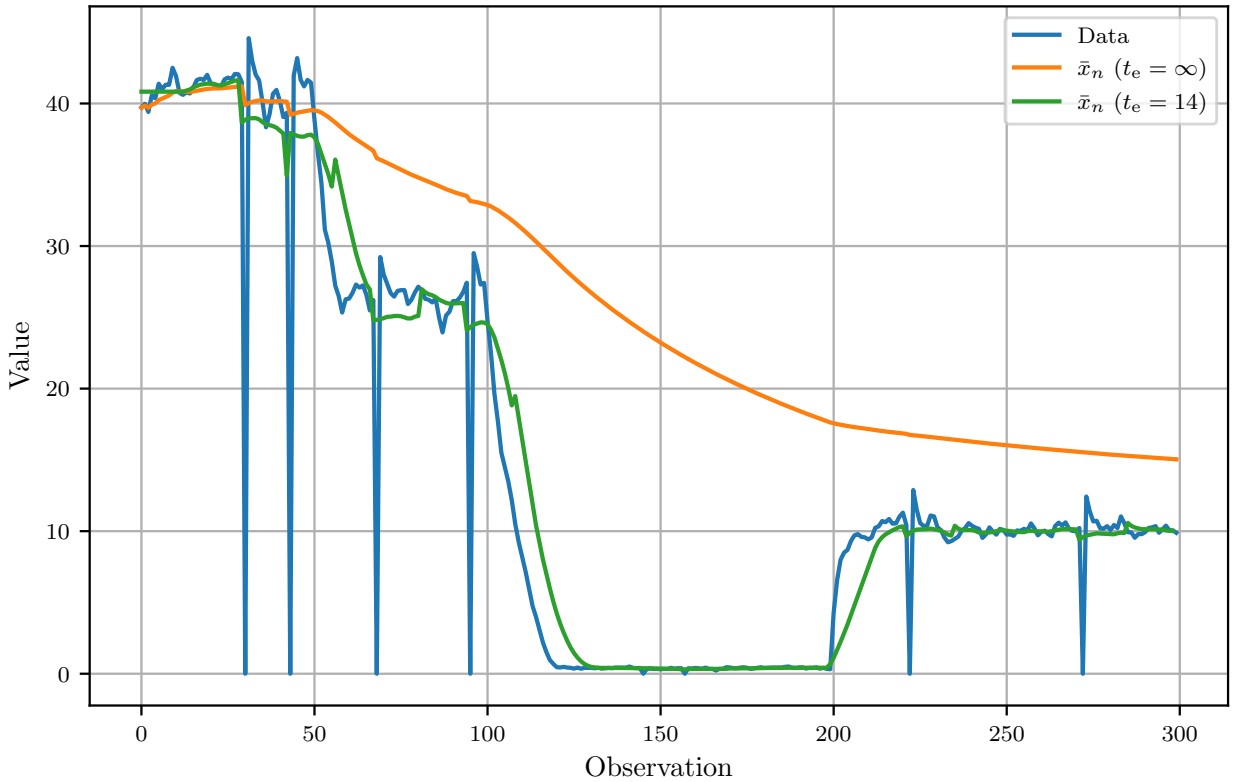
- Incremental Learning
- Anomaly Detection
- Interpretable Probabilistic Model

Welford Algorithm



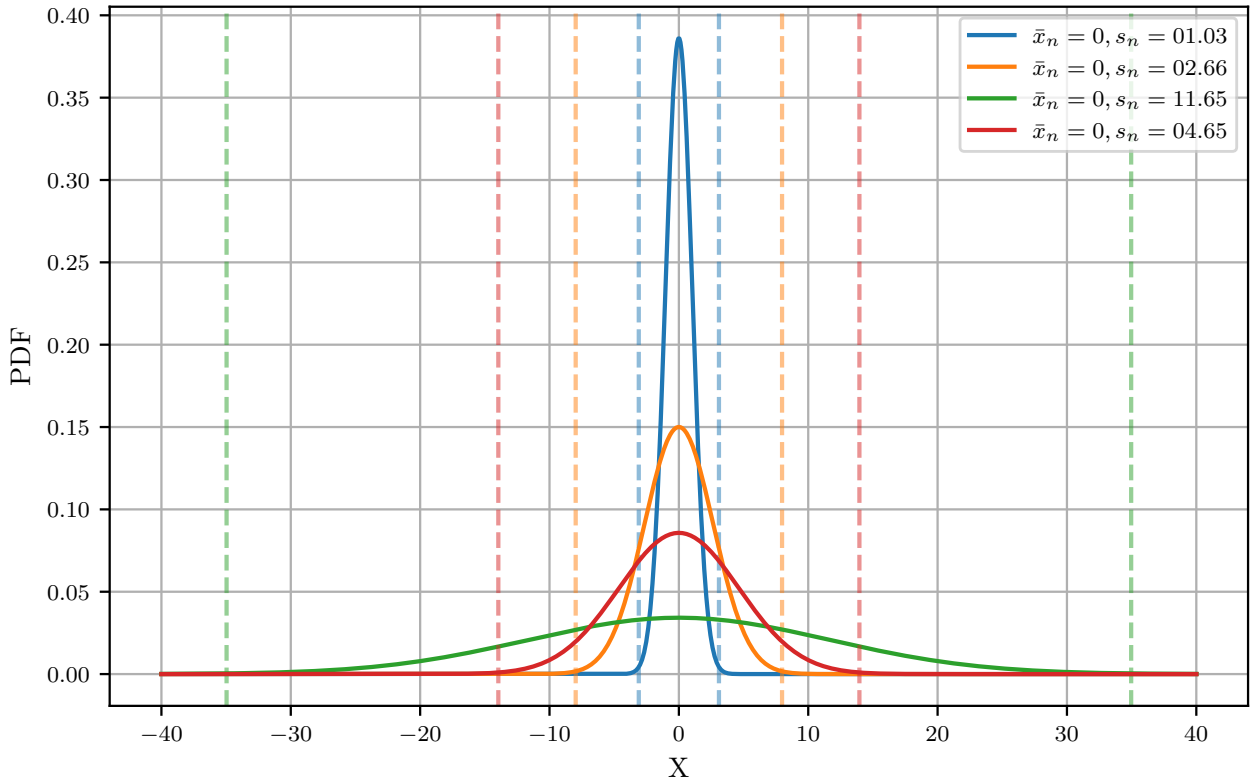
\+ One-Pass Algorithm | − Adaptation Slows Down

Inverse Welford Algorithm



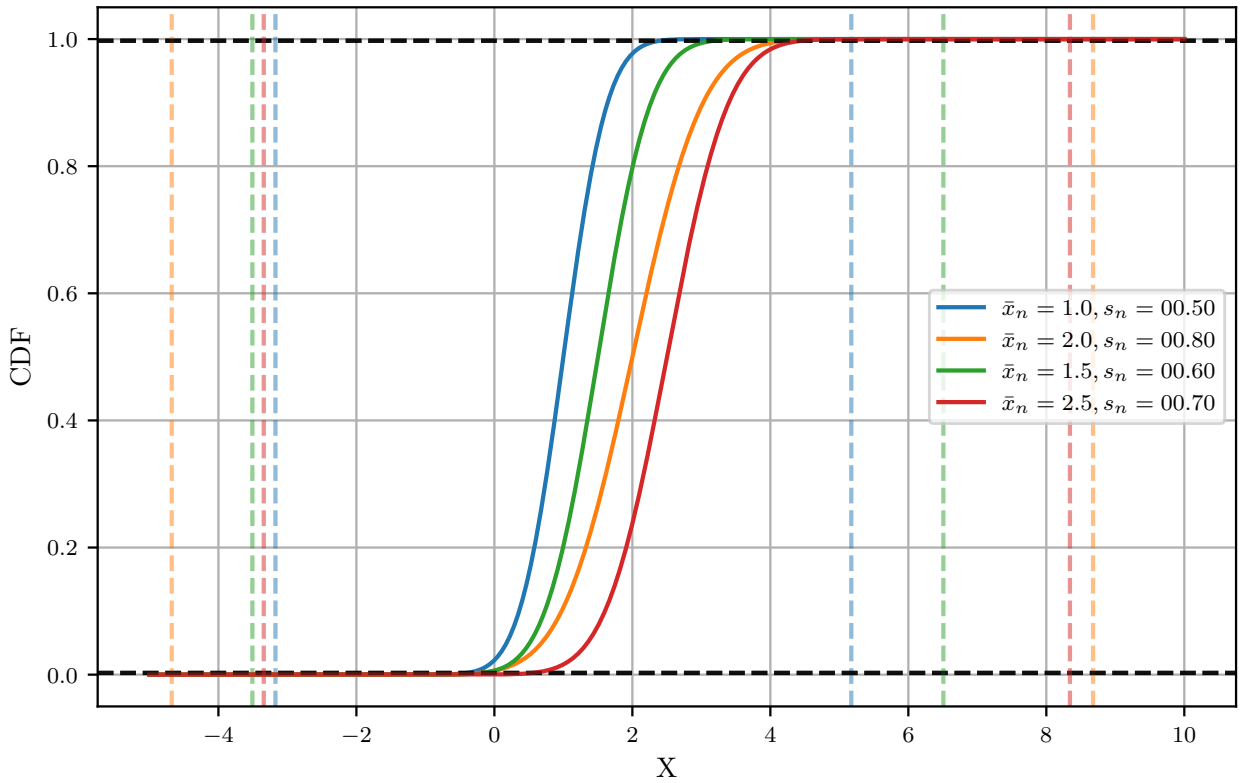
\+ Constant Adaptation | − Memorizes Data Window

Deviation-based Detection



$$y_i = \begin{cases} 0 & \text{if } q \leq F_X(x_i; \bar{x}_n, s_n), \quad (1a) \\ 1 & \text{if } q > F_X(x_i; \bar{x}_n, s_n). \quad (1b) \end{cases}$$

Inversion of CDF



$$x_{\mathrm{l}} = F_X(1-q;\bar{x}_n,s_n)^{-1},$$
$$x_{\mathrm{u}} = F_X(q;\bar{x}_n,s_n)^{-1}.$$

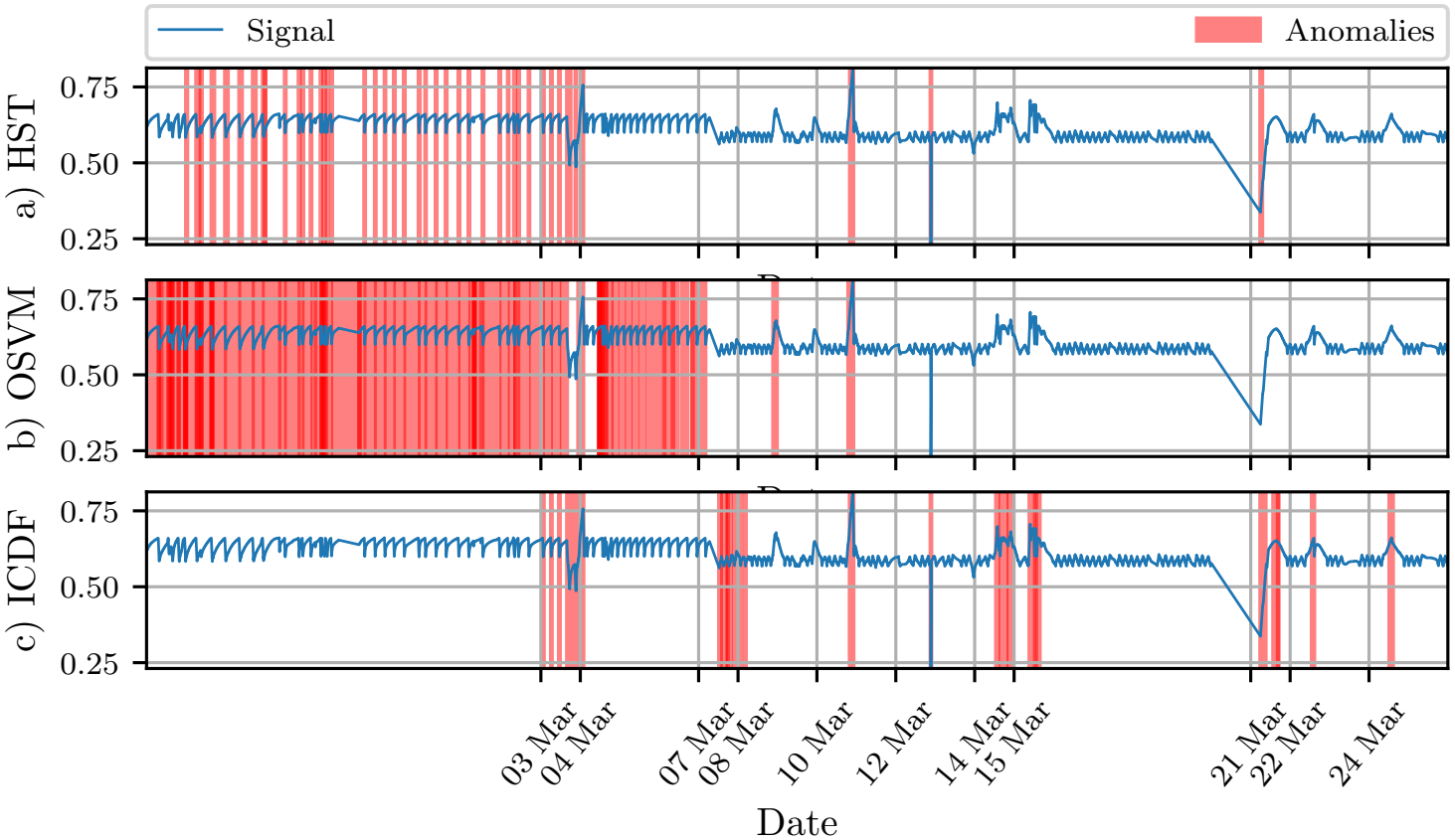
Online Anomaly Detection Workflow

Input: expiration period t_e, time constant t_c
Output: score y_i, threshold $x_{q,i}$

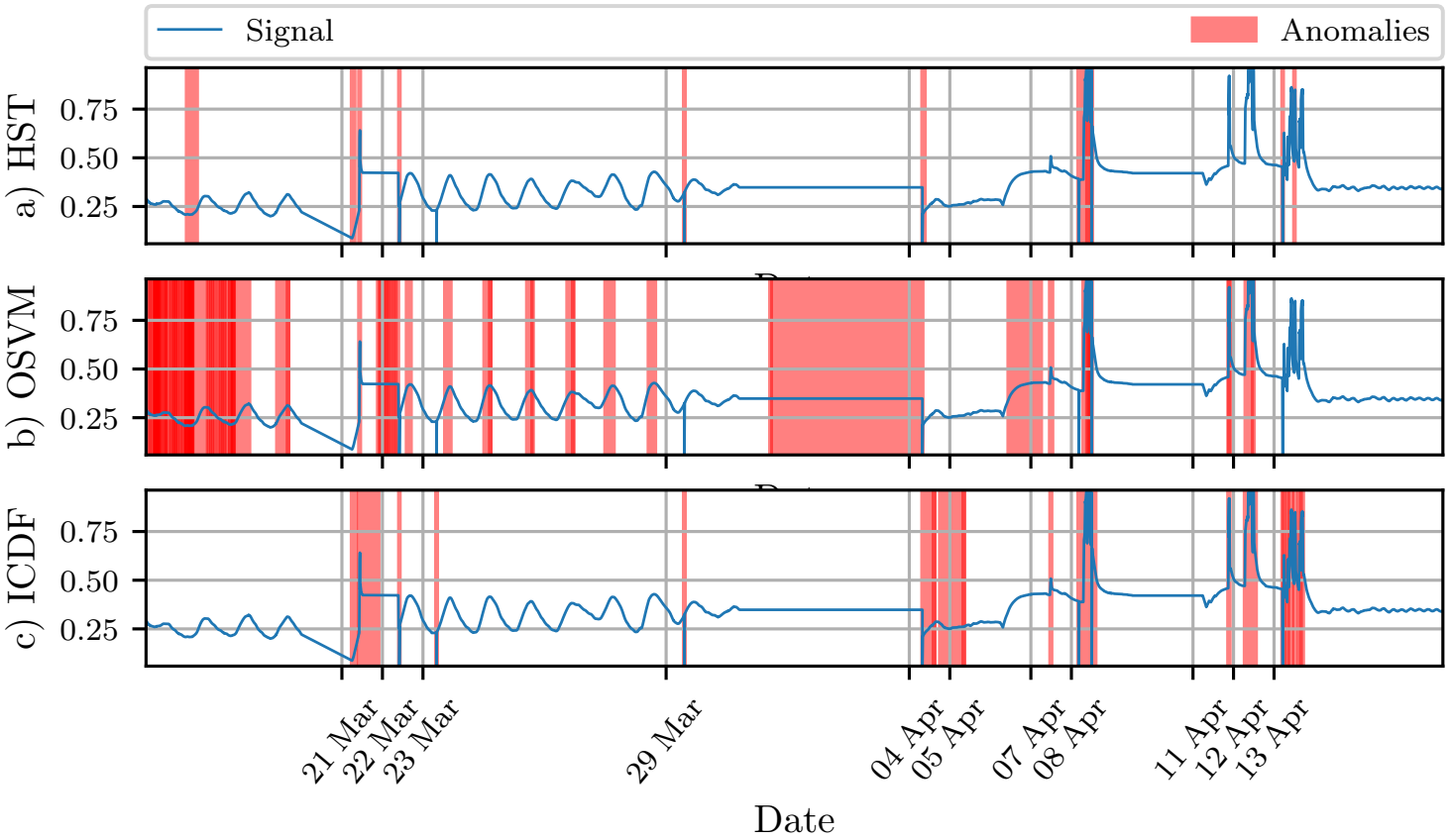
Initialisation :

$i \leftarrow 1;\ n \leftarrow 1;\ q \leftarrow 0.9973;\ \bar{x} \leftarrow x_0;\ s^2 \leftarrow 1;$
compute $F_X(x_0)$;
LOOP Process
loop
  $x_i \leftarrow$ RECEIVE();
  $y_i \leftarrow$ PREDICT(x_i) ;
  $x_{q,i} \leftarrow$ GET$(q, \bar{x}, s^2)$;
  if (1a) **or** $\sum_{y \in Y} y/n(Y) > q$ **then**
    $\bar{x},\ s^2 \leftarrow$ UPDATE$(x_i, \bar{x}, s^2, n)$;
    $n \leftarrow n + 1$;
    for x_{i-t_e} **do**
      $\bar{x},\ s^2 \leftarrow$ REVERT$(x_{i-t_e}, \bar{x}, s^2, n)$;
      $n \leftarrow n - 1$;
    end for
  end if
  $i \leftarrow i + 1$;
end loop

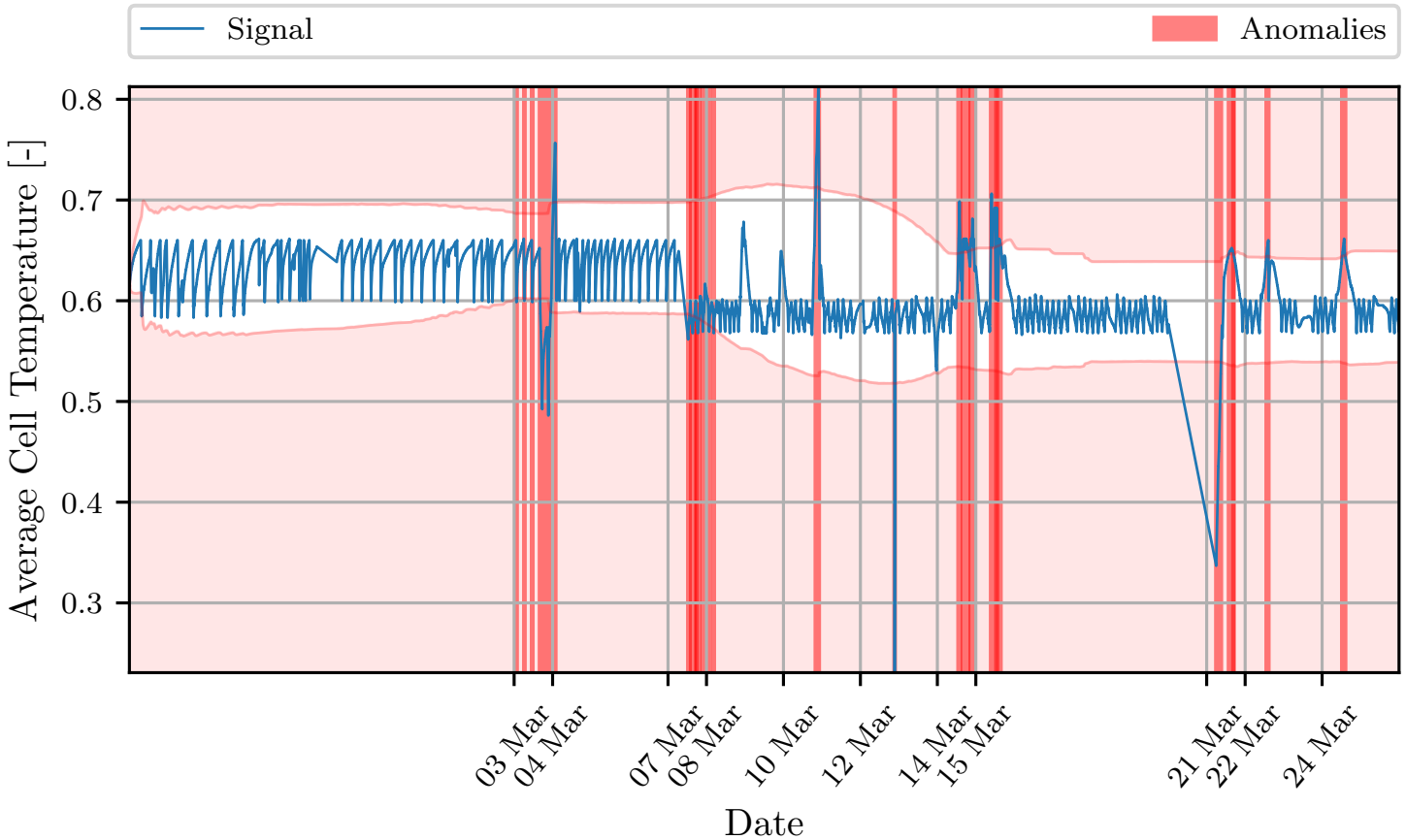
ICDF-based Outlier Detection - BESS



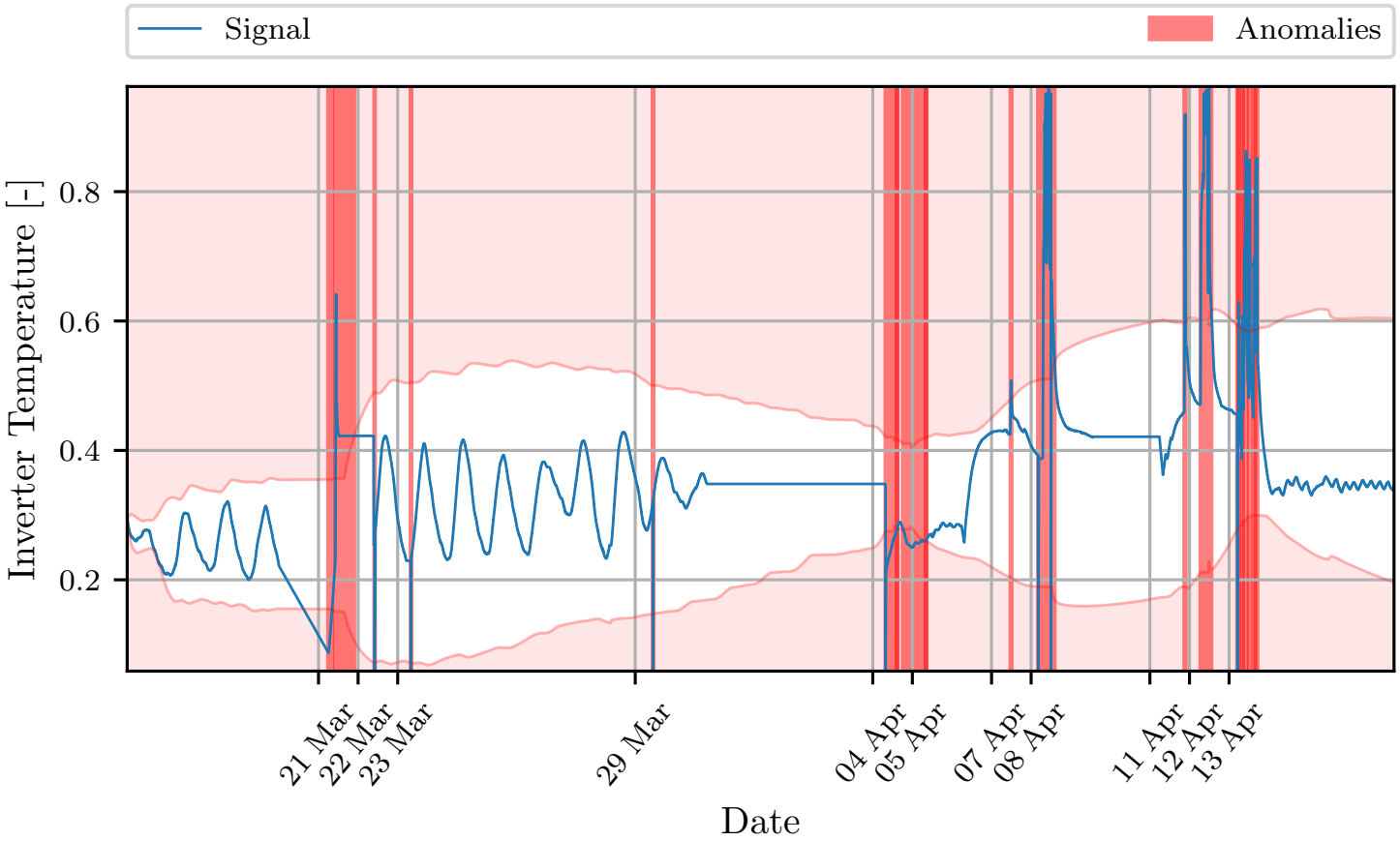
ICDF-based Outlier Detection - Inverter



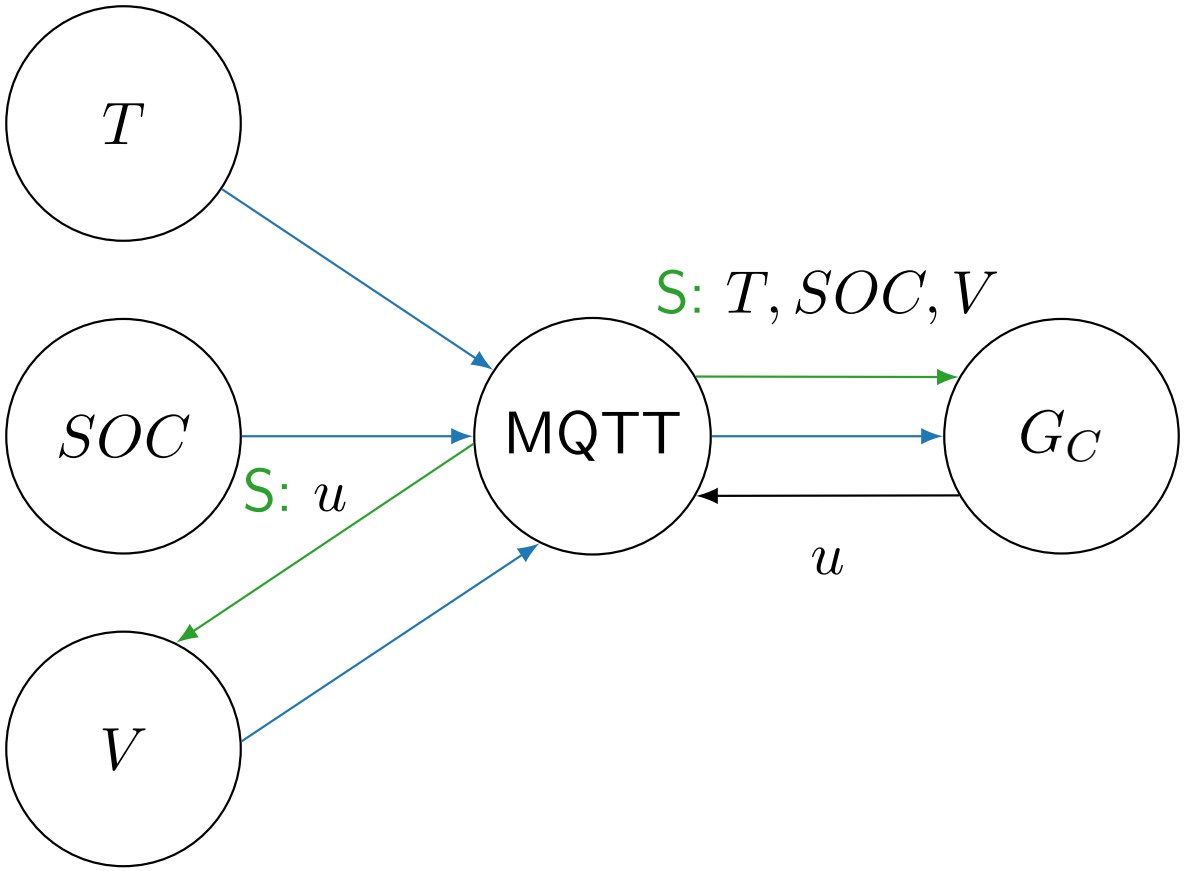
Dynamic Process Limits



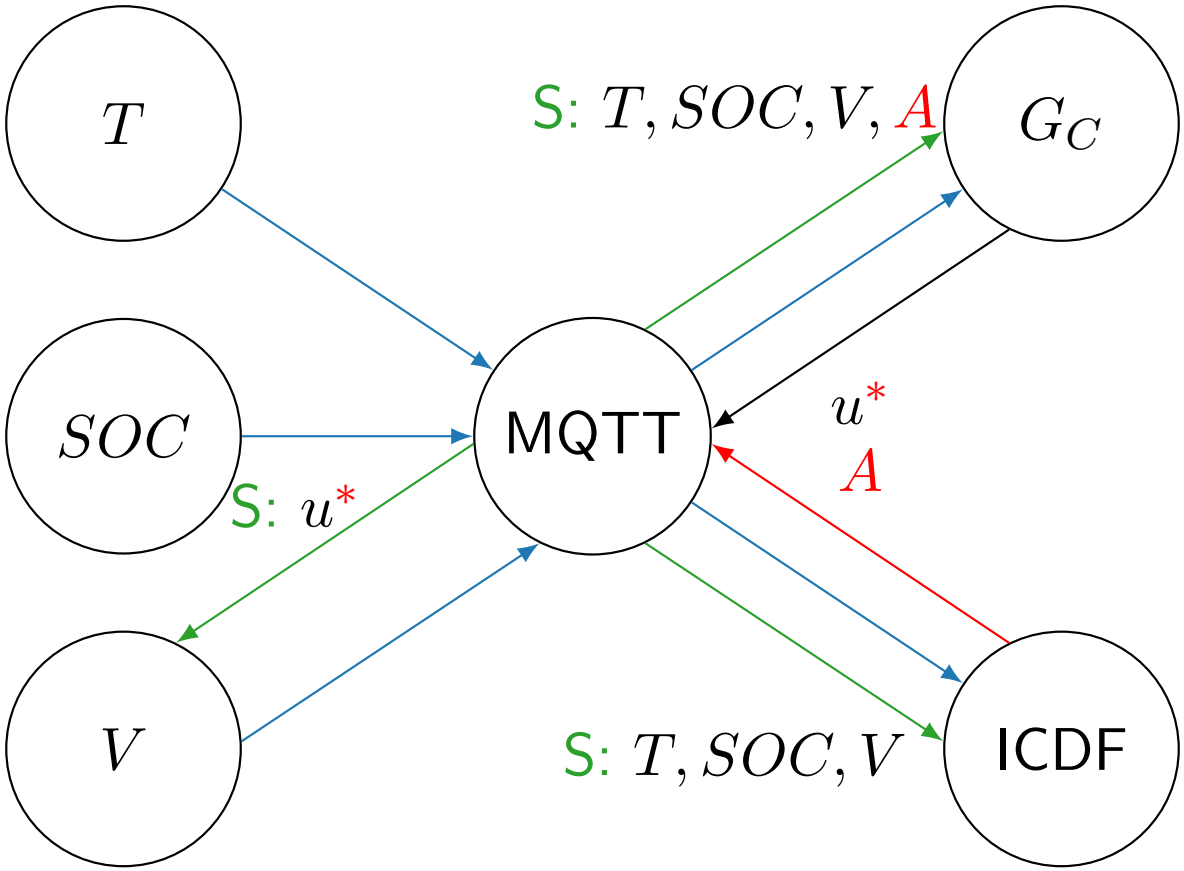
Dynamic Process Limits



Utilize Existing Infrastructure



Utilize Existing Infrastructure



The Long Win

1

[1] CDC