Sliding window based fault detection from high-dimensional data streams

Liangwei Zhang

Division of Operation and Maintenance Engineering

Authors: Liangwei Zhang, Janet Lin, Ramin Karim

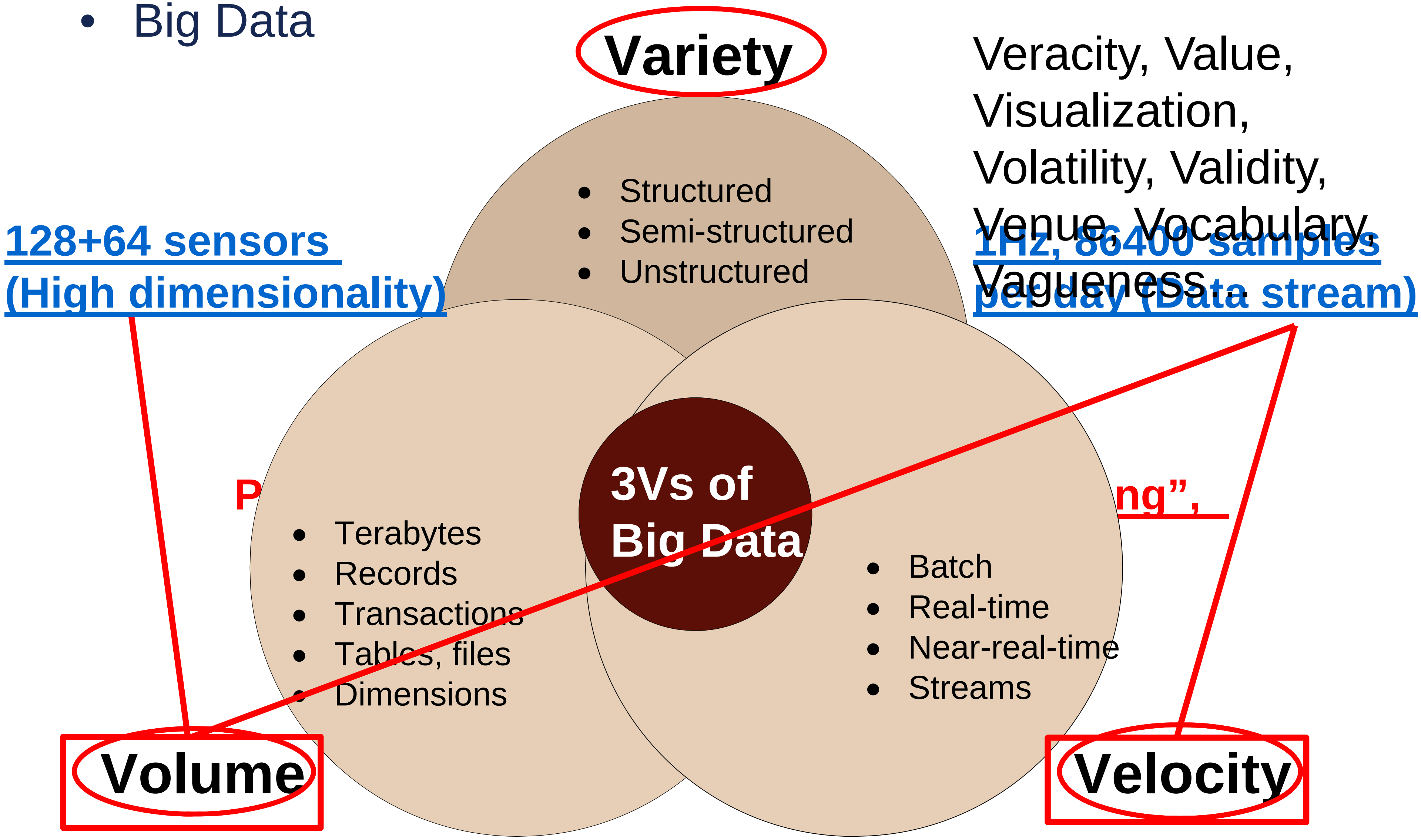
OUTLINE

1. Background
2. Problem statement
3. Existing work
4. Sliding Window-based ABSAD
5. Summary

1. Background

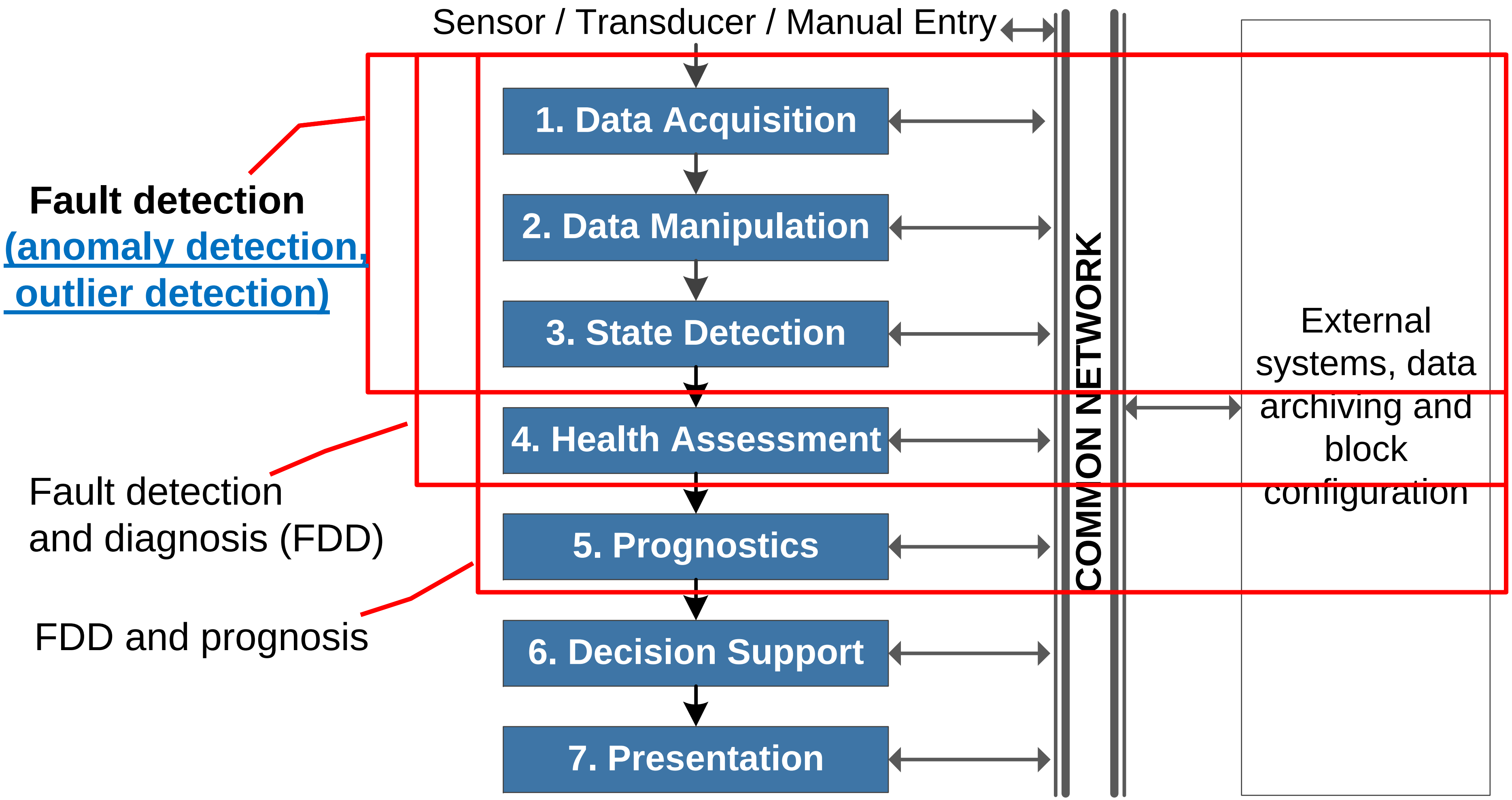
- Big Data

Veracity, Value, Visualization, Volatility, Validity, Venue, Vocabulary, Vagueness…



1. Background

- ISO 13374-2, OSA-CBM (Open System Architecture for Condition-Based Maintenance)

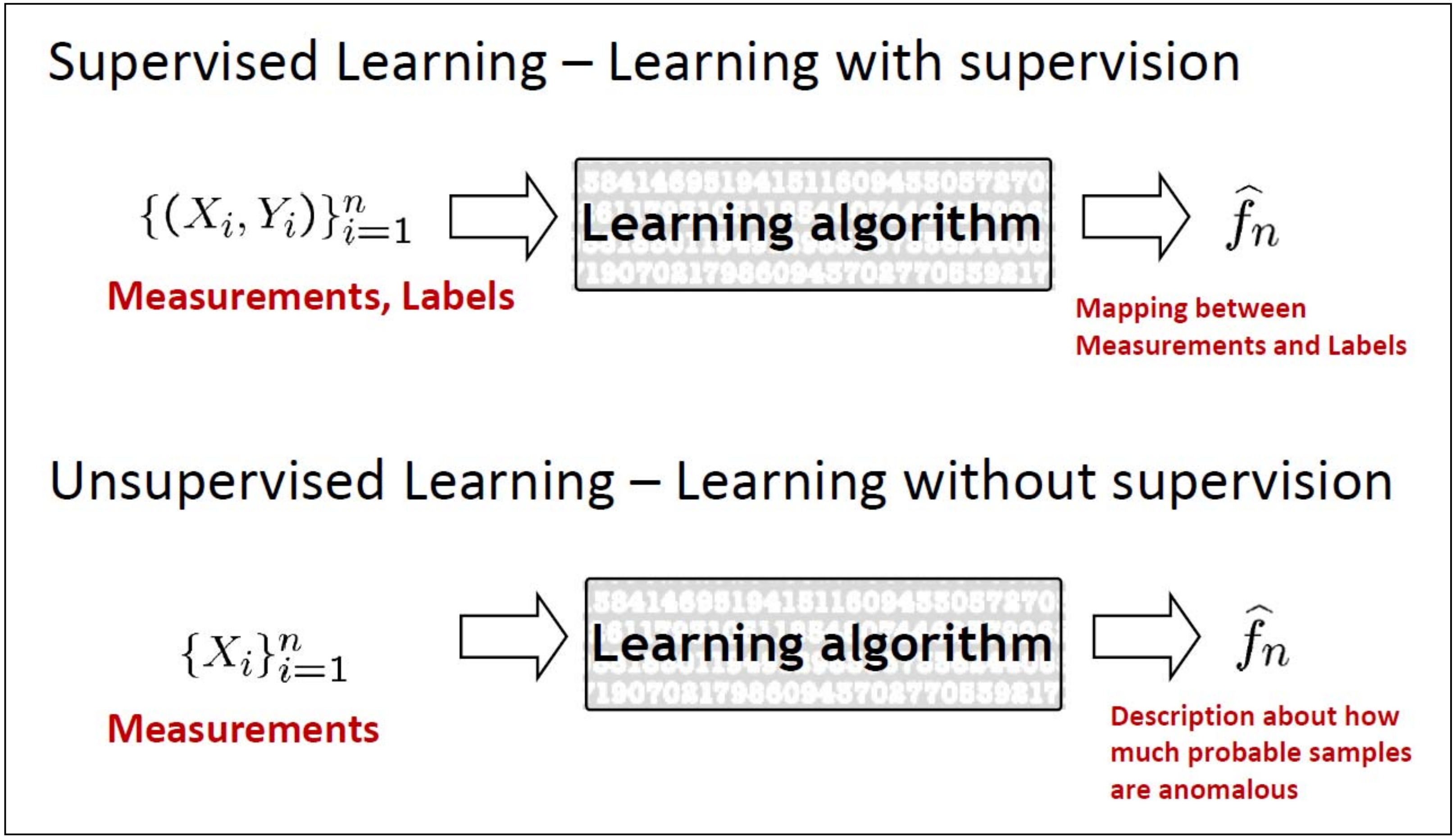


1. Background

- Fault detection
  - Fault detection aims to identify defective states and conditions within complex industrial systems, subsystems and components
  - Detect “something has gone wrong”
  - From a machine learning viewpoint, it answers “yes or no”, skewed binary classification problem

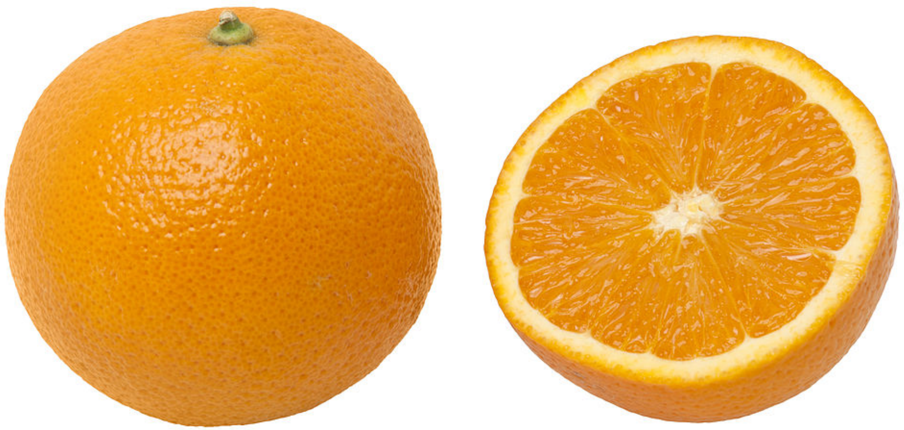
1 Background

- Machine Learning, models are fixed once learned.



2. Problem statement

- Curse of dimensionality
  - Number of training samples (exponentially increased)
  - Notions like proximity, distance, neighborhood become less meaningful (the concentration of norms)
  - Probability theory is counter-intuitive



2. Problem statement

- Requirements from data stream
  - Real-time or near real-time response
  - One-scan algorithms
  - Concept drift
    - System has time-varying characteristics due to seasonal flunctuation, equipment aging, process drifting, etc.

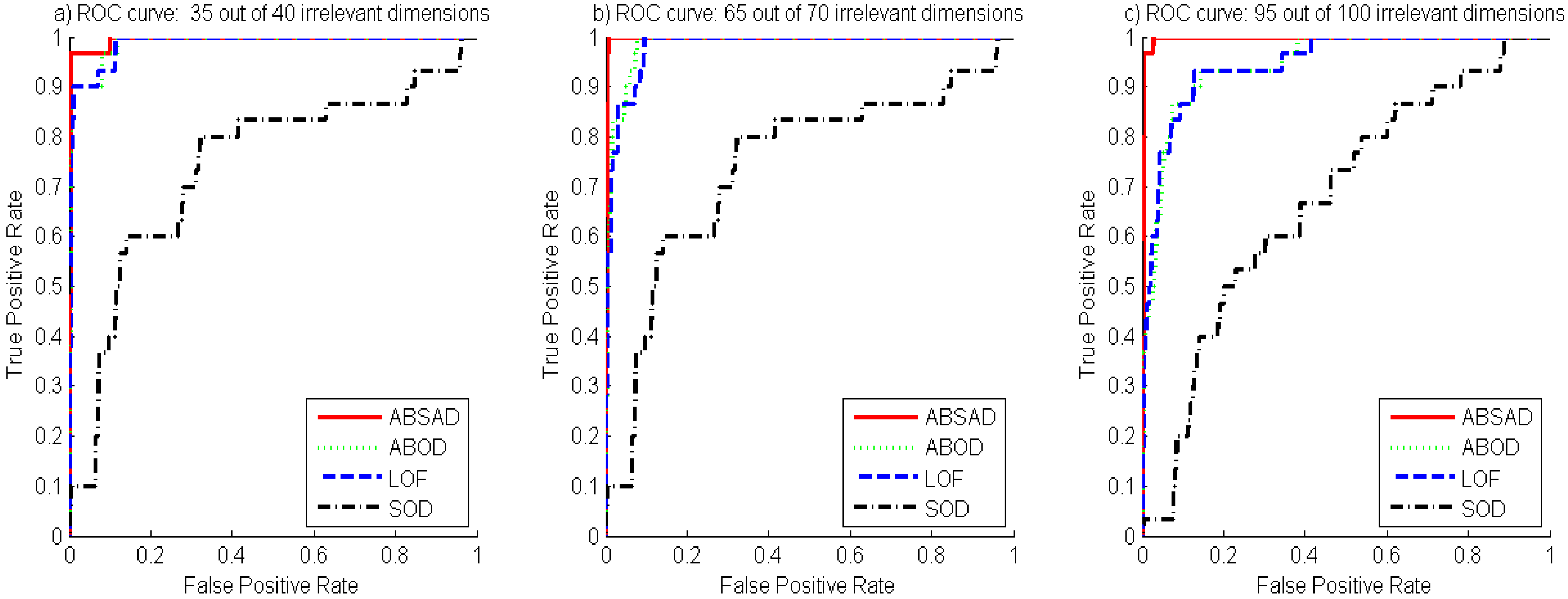
3. Existing work

- Typically, extension of existing algorithms
  - Exponential weighted PCA
  - Sliding window PCA
  - Recursive PCA

- The key is the learned model should be refined, enhanced, and personalized while stream evolves so as to accomodate the natural drift in data stream.

- Research in fault detection from high-dimensional data stream is under-explored.

4.1 ABSAD: fault detection from high-dimensional data



Zhang, Liangwei, Jing Lin, and Ramin Karim. "An angle-based subspace anomaly detection approach to high-dimensional data: With an application to industrial fault detection." *Reliability Engineering & System Safety* 142 (2015): 482-497.

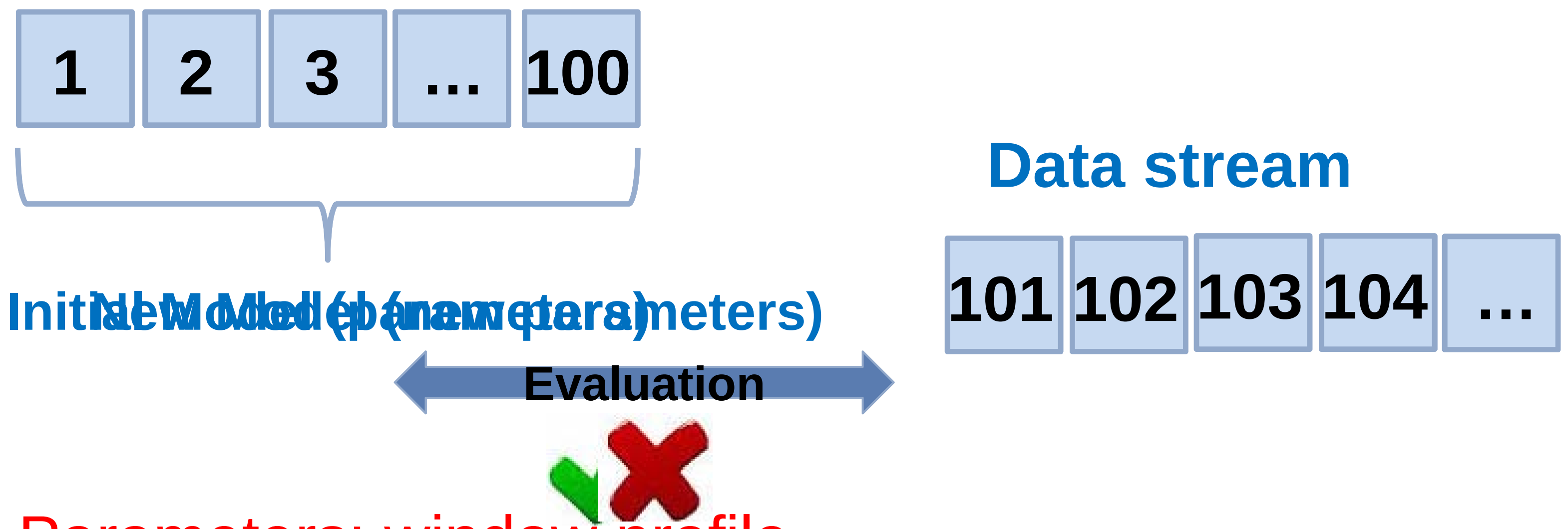
4.2 Sliding window-based ABSAD

- sliding window strategy

Stage 1: Offline model training

Stage 2: Online fault detection

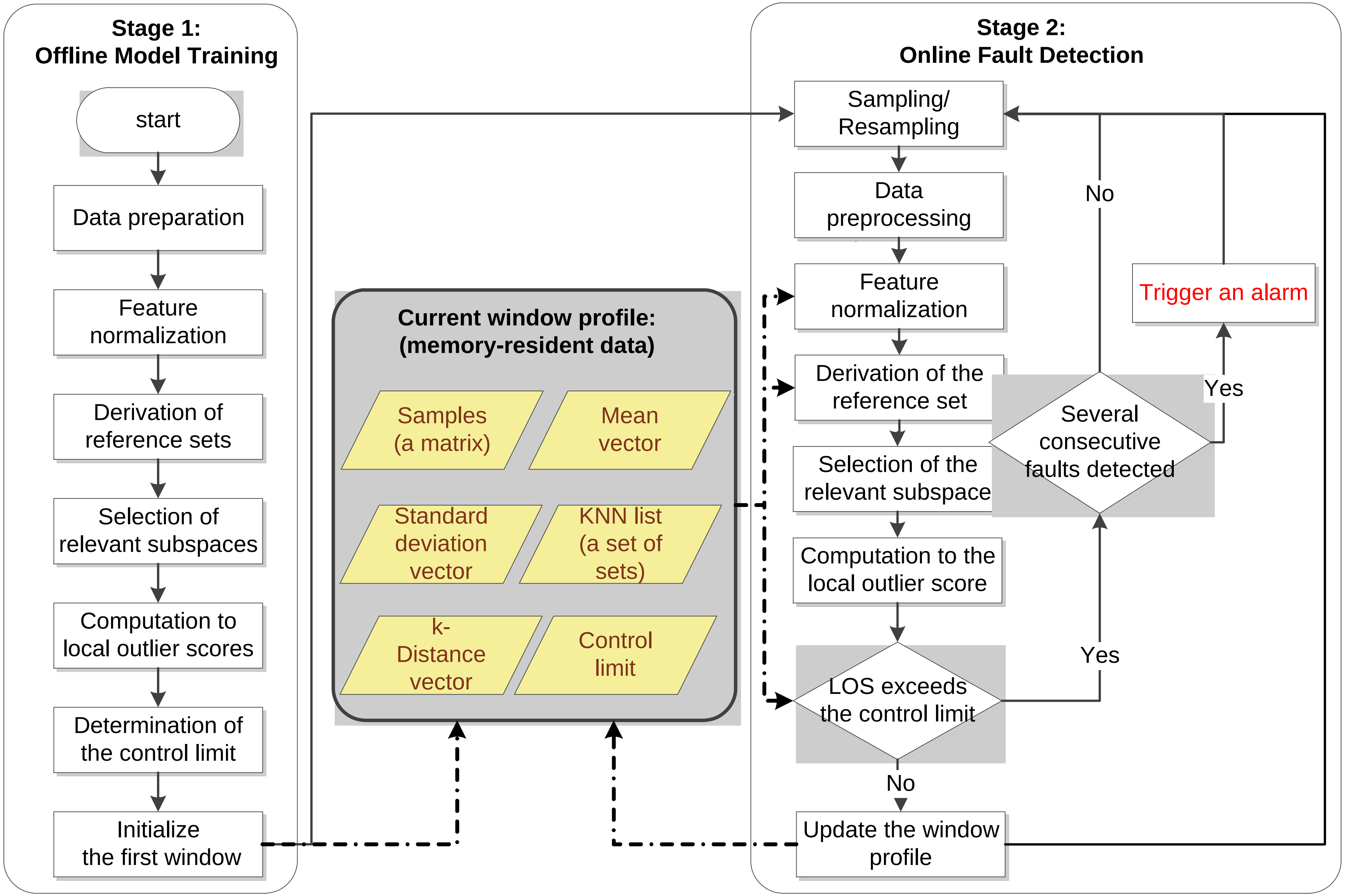
Window size L: 100



- Parameters: window profile
- ”Primitive model”: initial model is built once for all

4.2 Sliding window-based ABSAD

- Model structure



4.2 Sliding window-based ABSAD

- Computational complexity
  - The first stage takes the same complexity as the ABSAD approach, O$(m^2 \cdot \max(n, k))$. If some structure is employed, O$(m \log m \cdot \max(n, k))$
  - The time complexity of the second stage for processing a single sample is O$(L \cdot \max(n, k))$ and the space complexity is O$(L \cdot \max(n, k))$

4.2 Sliding window-based ABSAD

- Numerical illustration: data
  - System modeled on the input-output form

$$\boldsymbol{O}(t) = \boldsymbol{A} \cdot \boldsymbol{I}(t) + \boldsymbol{E}(t) = \begin{bmatrix} 0.86 & 0.79 & 0.67 & 0.81 \\ -0.55 & 0.65 & 0.46 & 0.51 \\ 0.17 & 0.32 & -0.28 & 0.13 \\ -0.33 & 0.12 & 0.27 & 0.16 \\ 0.89 & -0.97 & -0.74 & 0.82 \end{bmatrix} \cdot \begin{bmatrix} I_1(t) \\ I_2(t) \\ I_3(t) \\ I_4(t) \end{bmatrix} + \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ e_4(t) \\ e_5(t) \end{bmatrix}$$

$$I_1(t) = 2 \cdot \cos(0.08t) \cdot \sin(0.006t)$$

$$I_2(t) = \mathrm{sign}[\sin(0.03t) + 9 \cdot \cos(0.01t)]$$

$$I_3(t) \sim U(-1,1)$$

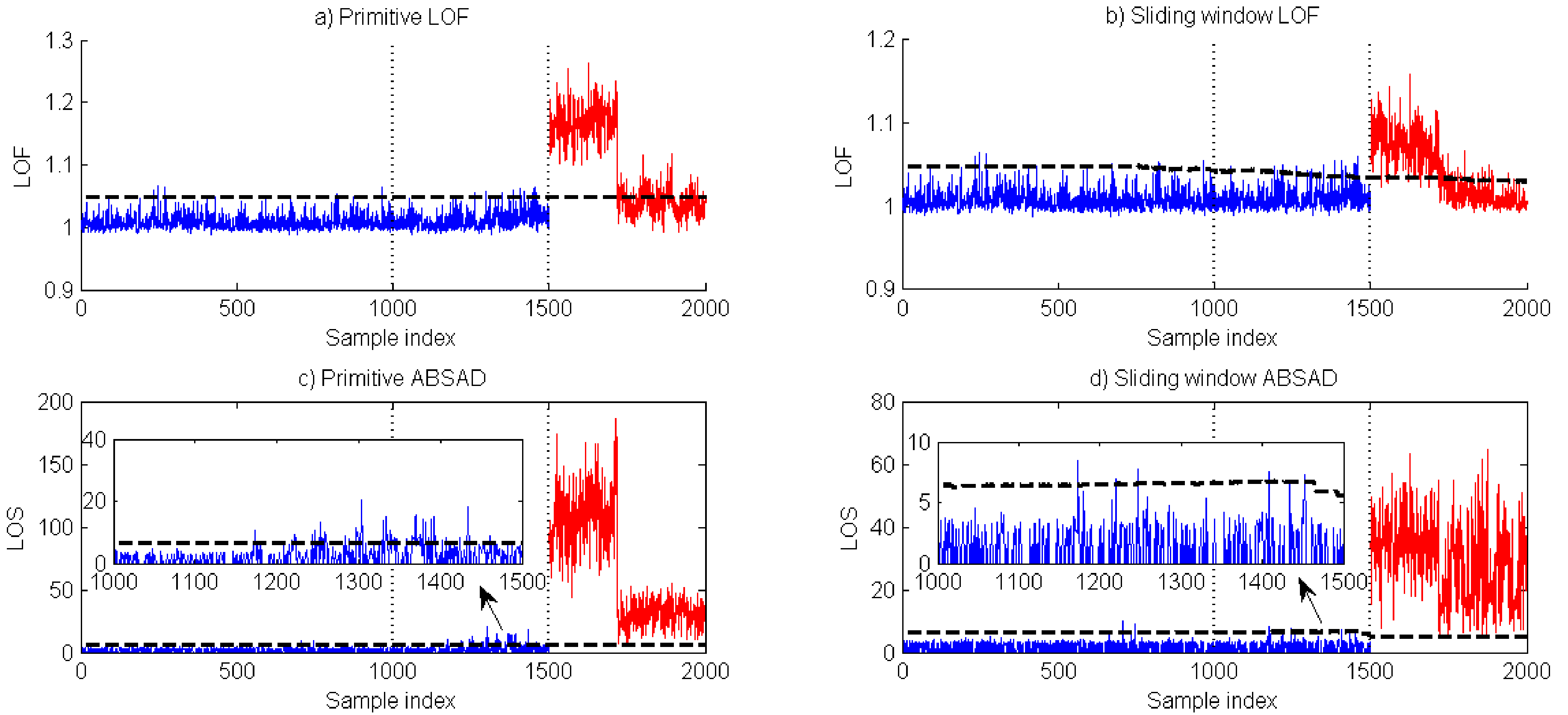
$$I_4(t) \sim N(2,0.1)$$

4.2 Sliding window-based ABSAD

- Numerical illustration: data
  - 95 fault-irrelevant dimensions were added to create a high-dimensional setting
  - 4 datasets (4 faults), 2000 by 100 matrix
  - All faults were induced starting from the 1501st sample
  - slow drifts were added starting from the 1001st sample

4.2 Sliding window-based ABSAD

- Numerical illustration: results and discussion
  - Fault detection results on scenario 2



4.2 Sliding window-based ABSAD

- Numerical illustration: results and discussion

Dataset and error type		Primitive LOF	Sliding window LOF	Primitive ABSAD	Sliding window ABSAD
Fault 1	Type I error	1.73[a]	1.73	8.4	0.67
	Type II error	32.2	91.8	0.2	4.4
Fault 2	Type I error	2.4	3.73	8.4	0.8
	Type II error	38.8	51	0	0
Fault 3	Type I error	2.8	2.27	8.13	1.2
	Type II error	0	36.4	0	0
Fault 4	Type I error	2.13	1.87	8.8	0.67
	Type II error	4.8	6.8	3.8	4.2

[a] Units of the decimal numbers in this table are in percent (%)

5. Conclusion

- The sliding window-based ABSAD approach can be adaptive to the time-varying behavior of the concerned system
- The sliding window ABSAD can perform isochronously online fault detection by trading space for time.

Acknowledgement

Thank you!