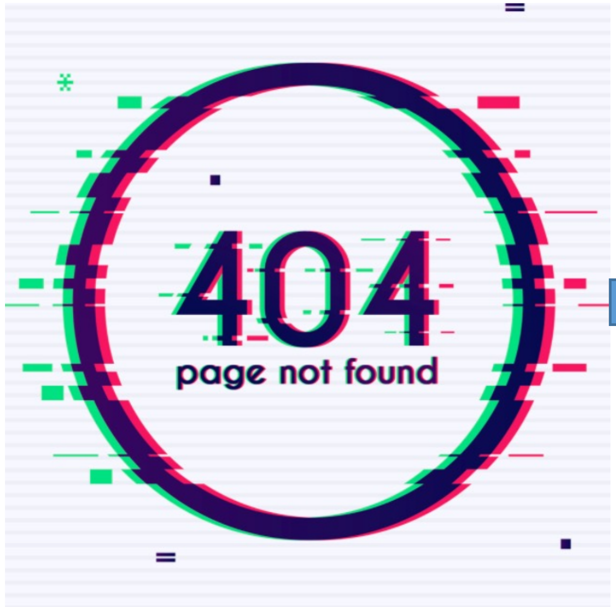
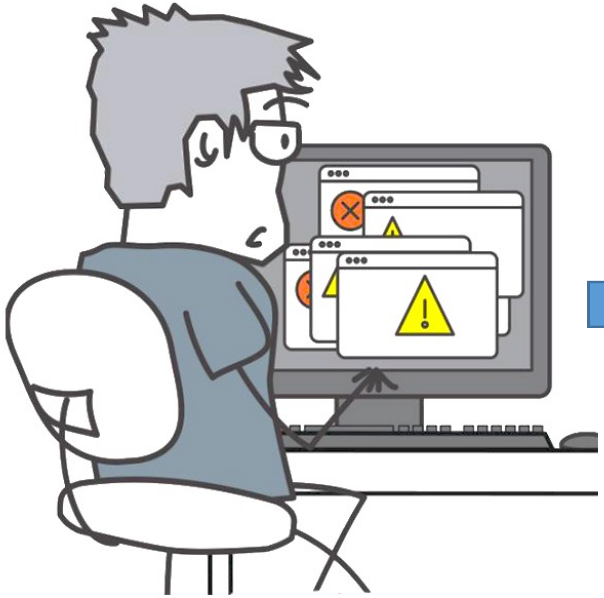
Jump-Starting Multivariate Time Series Anomaly Detection for Online Service Systems

Minghua Ma, Shenglin Zhang, Junjie Chen, Jun Xu, Dan Pei, et. al.

Service Reliability is Important



Users



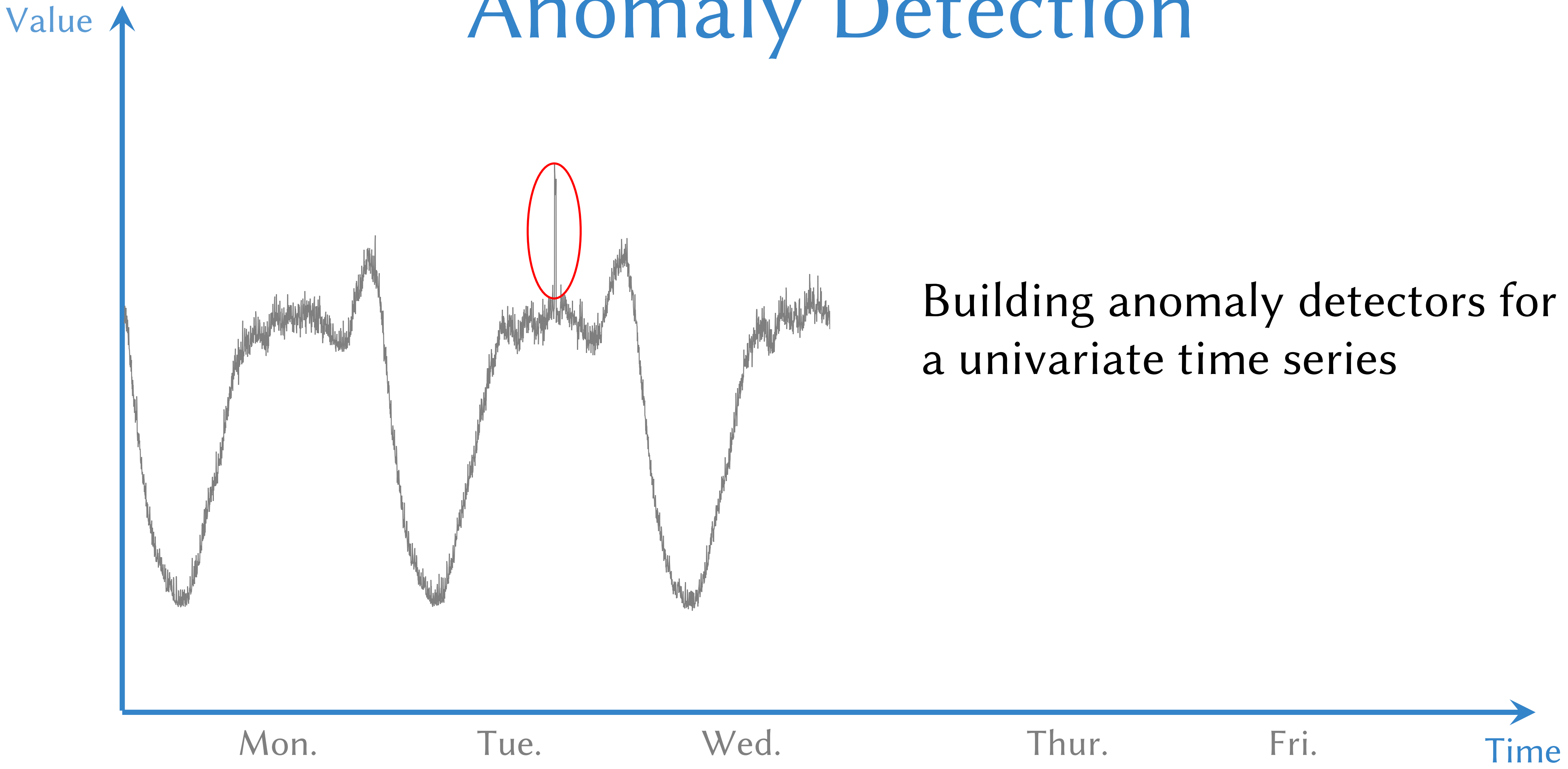
Operators

Companies

Real-World Revenue Loss

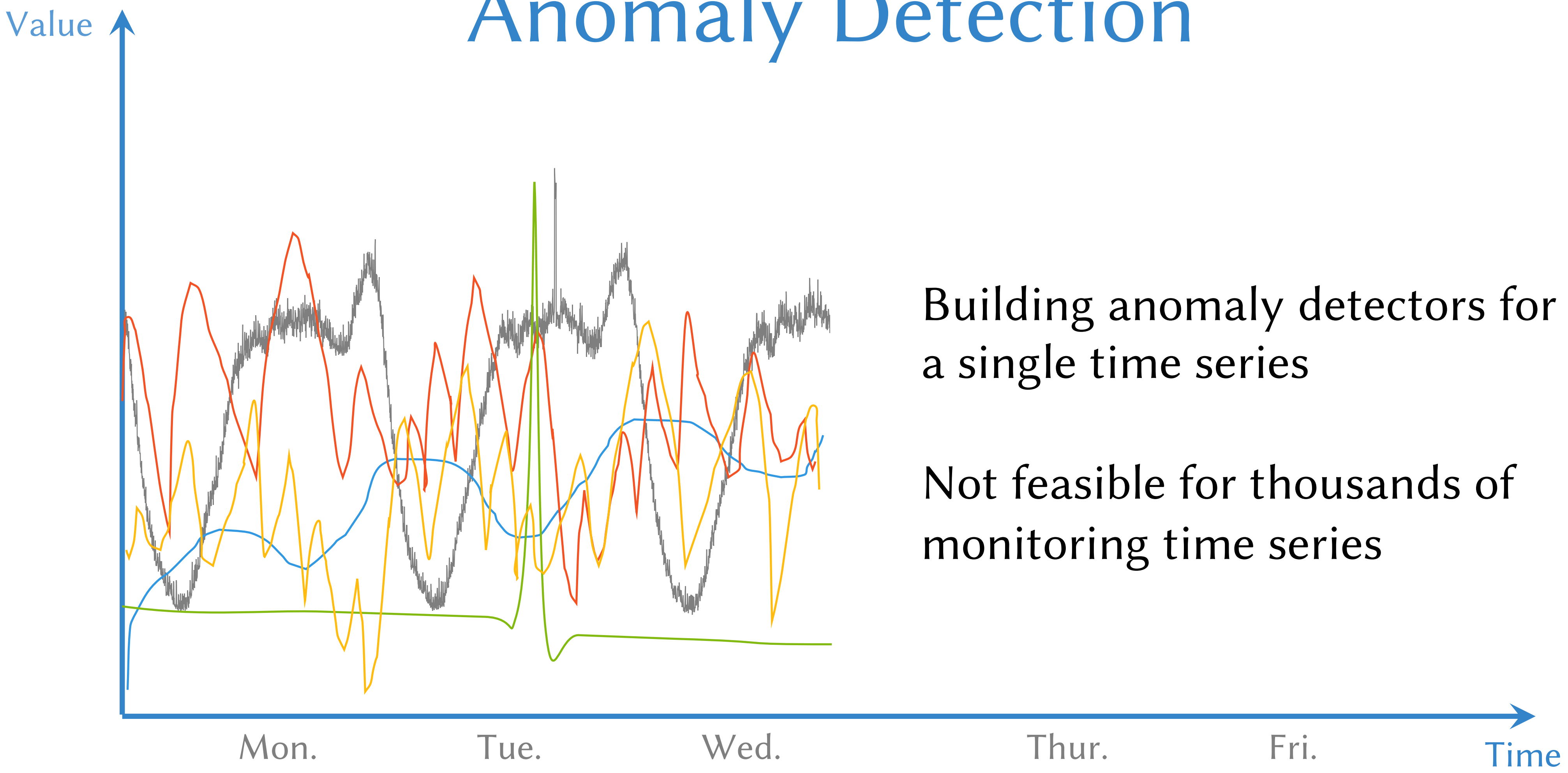
[Evolven: GAD COHEN]

Univariate Time Series (UTS) Anomaly Detection



Building anomaly detectors for a univariate time series

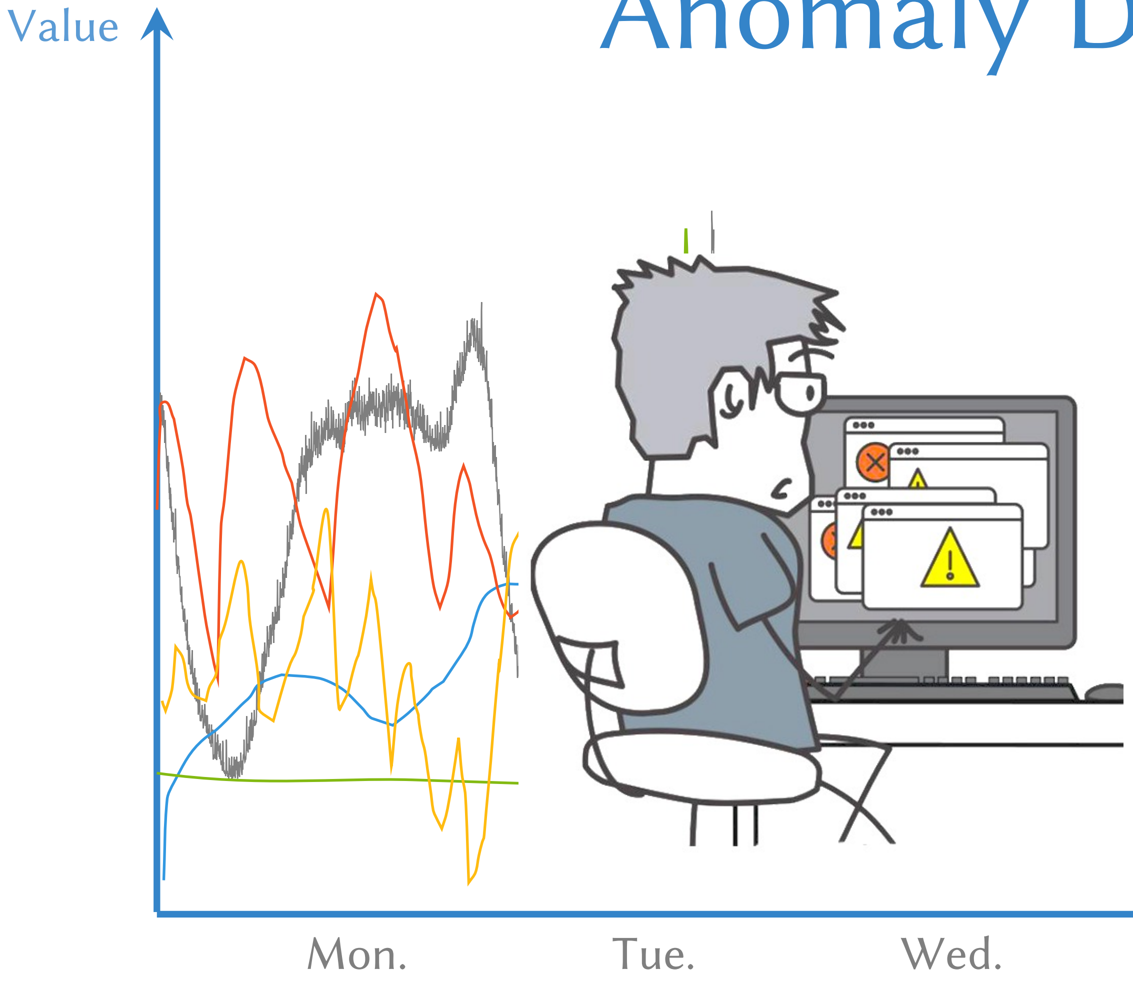
Univariate Time Series (UTS) Anomaly Detection



Building anomaly detectors for a single time series

Not feasible for thousands of monitoring time series

Univariate Time Series (UTS) Anomaly Detection

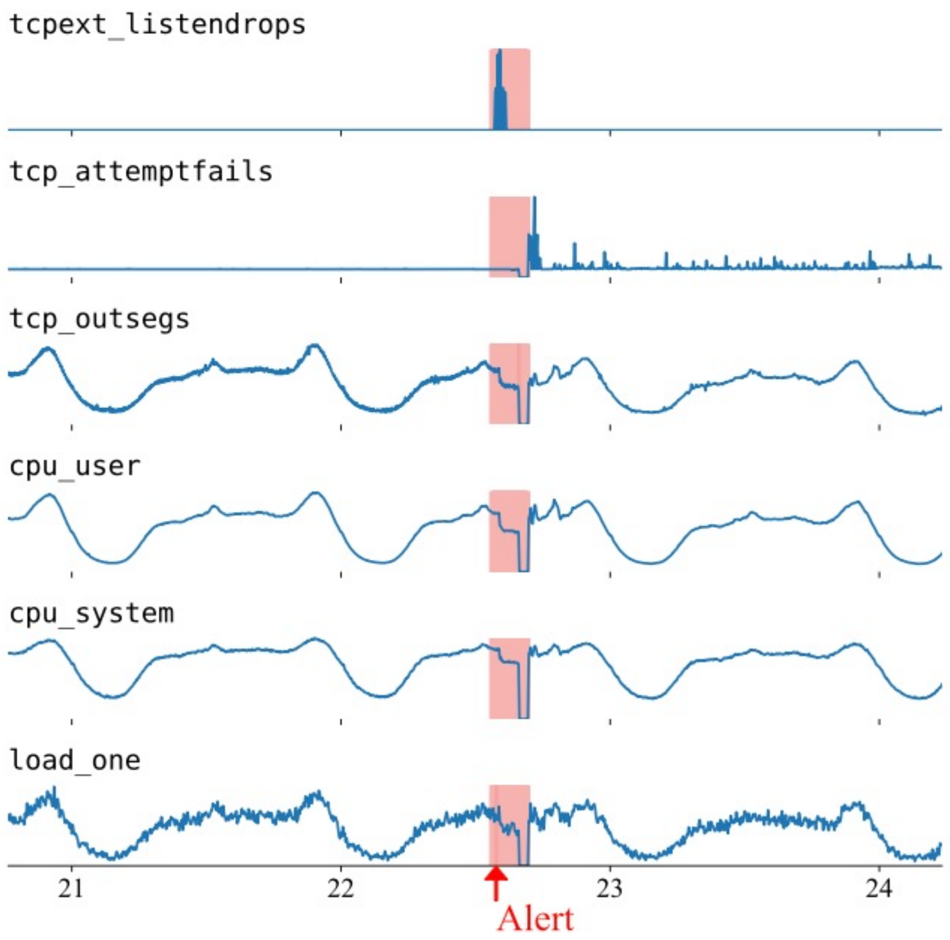


Building anomaly detectors for a single time series

Not feasible for thousands of monitoring time series

May lead to alert storms [SEIP20]

Multivariate Time Series (MTS) Anomaly Detection

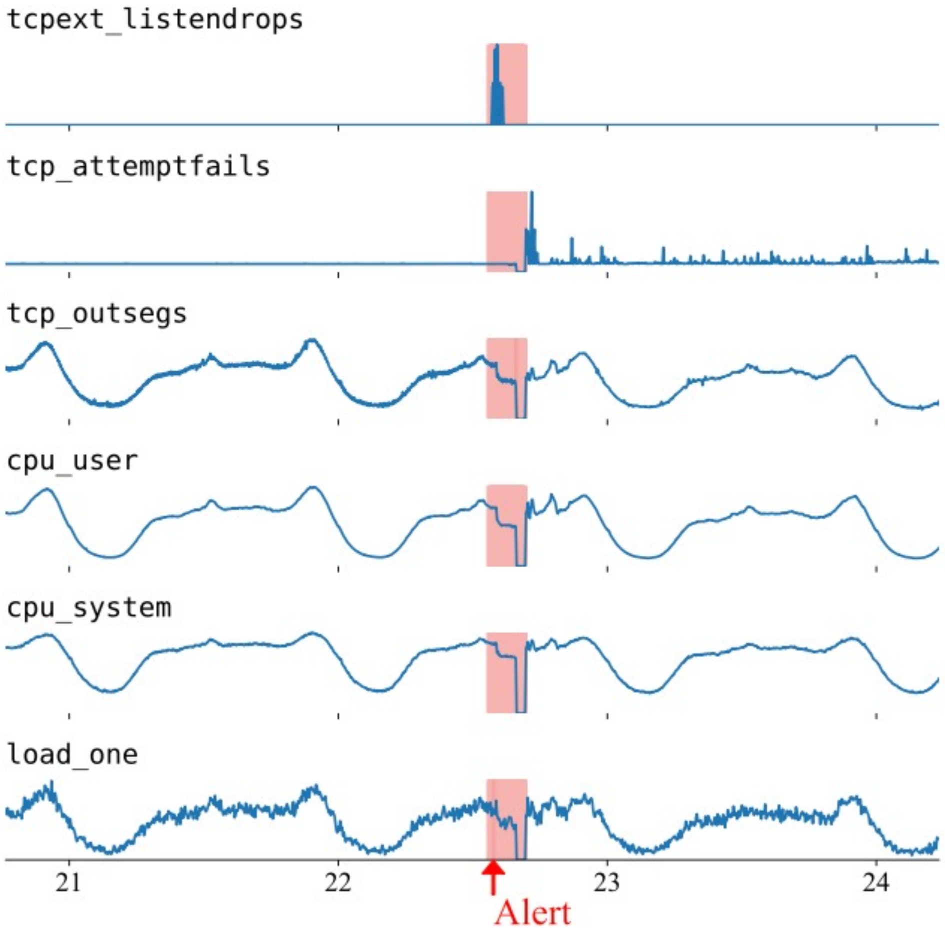


Capture status of the overall service system

Intuitive & effective & efficient

[KDD18, KDD19, KDD20, KDD21, AAAI19, AAAI21, NeurIPS20]

Multivariate Time Series (MTS) Anomaly Detection



Capture status of the overall service system

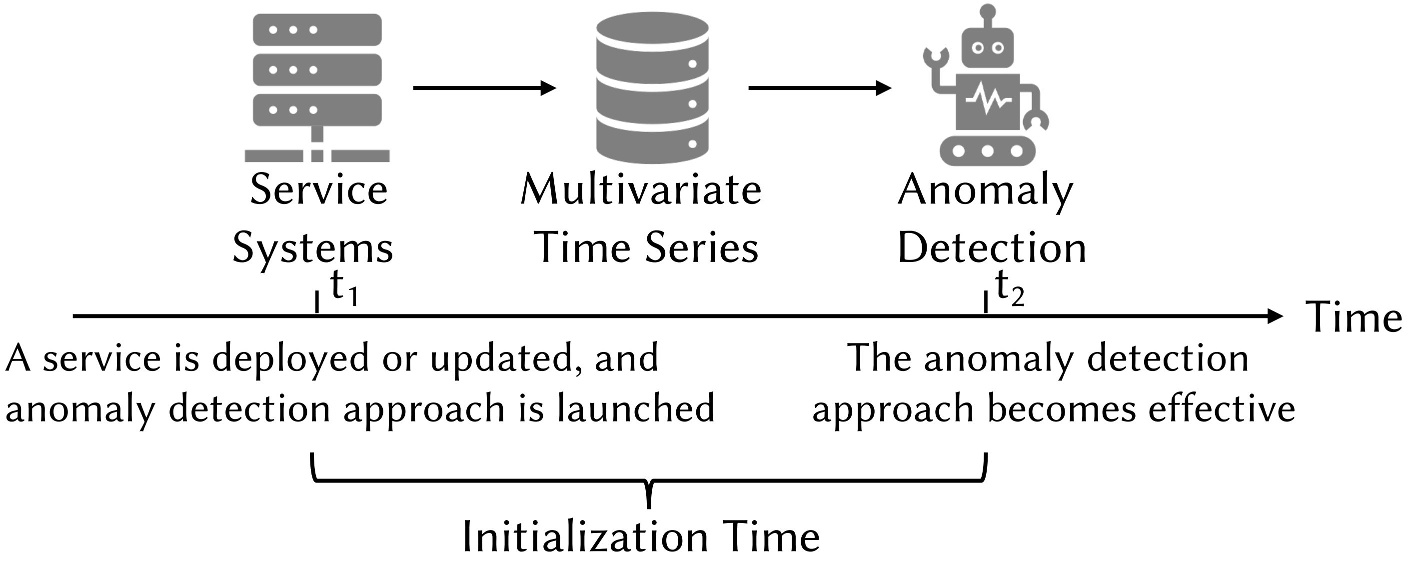
Intuitive & effective & efficient

[KDD18, KDD19, KDD20, KDD21, AAAI19, AAAI21, NeurIPS20]

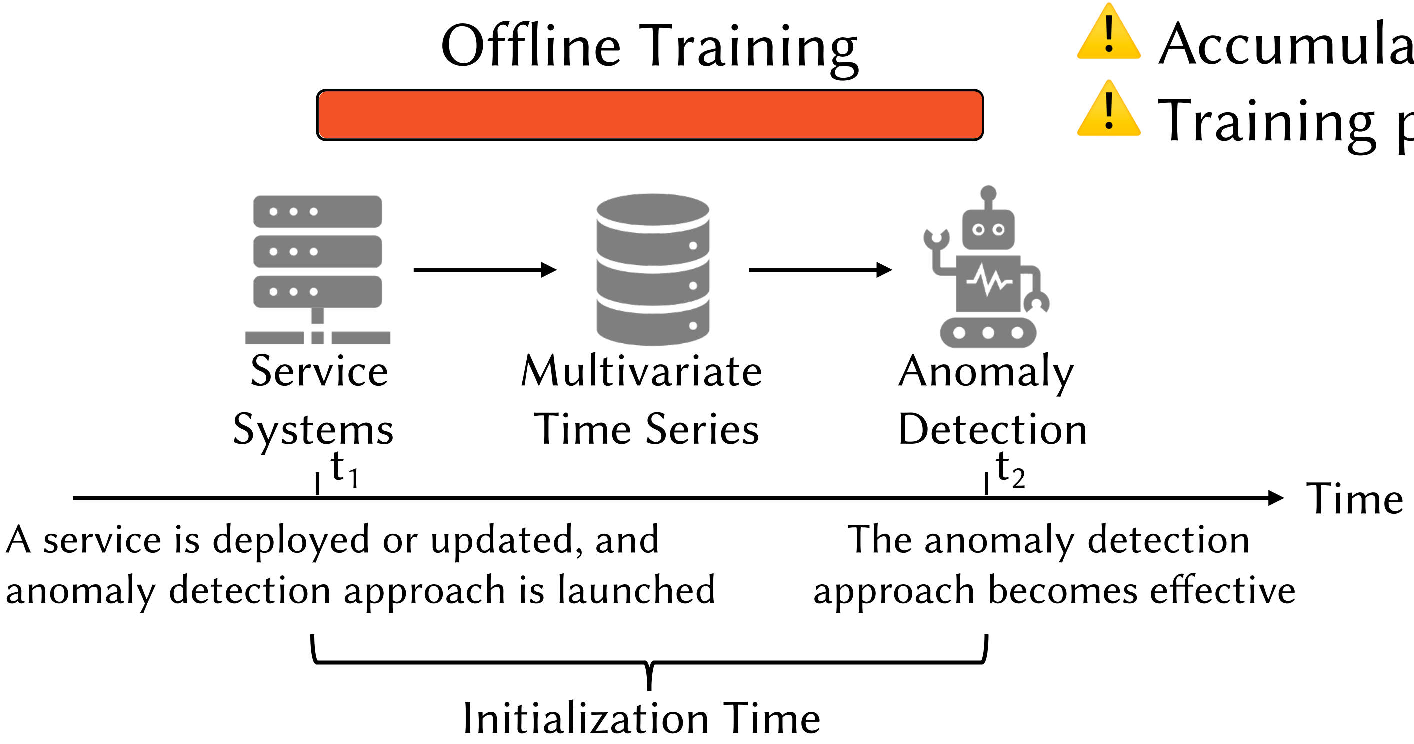
Deep learning based approaches (LSTM, LSTM-VAE, ConvLSTM…)

Initialization Time

Software change (concept drift) -> Anomaly detection -> Initialize



Deep Learning Based Approaches: Long Initialization Time



- ⚠ Accumulating training data
- ⚠ Training process

Deep Learning Based Approaches: Long Initialization Time

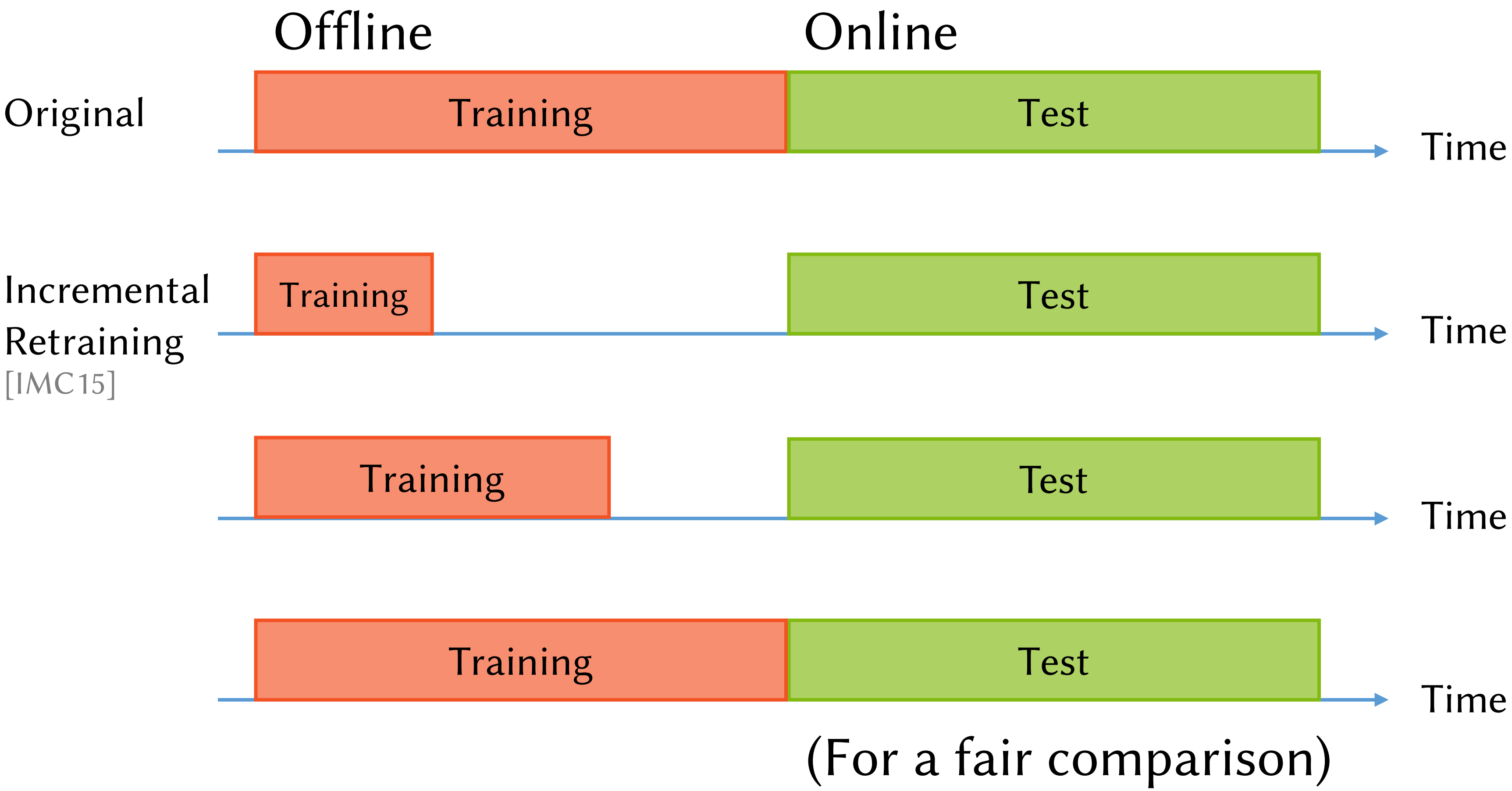
Approach	S1	S2	S3	Avg.
MSCRED [AAAI19]	7	13	-	10
OmniAnomaly [KDD19]	17	15	17	16.3
LSTM-NDT [KDD18]	69	36	-	52.5
Donut* [WWW18]	102	110	99	103.6

Days!

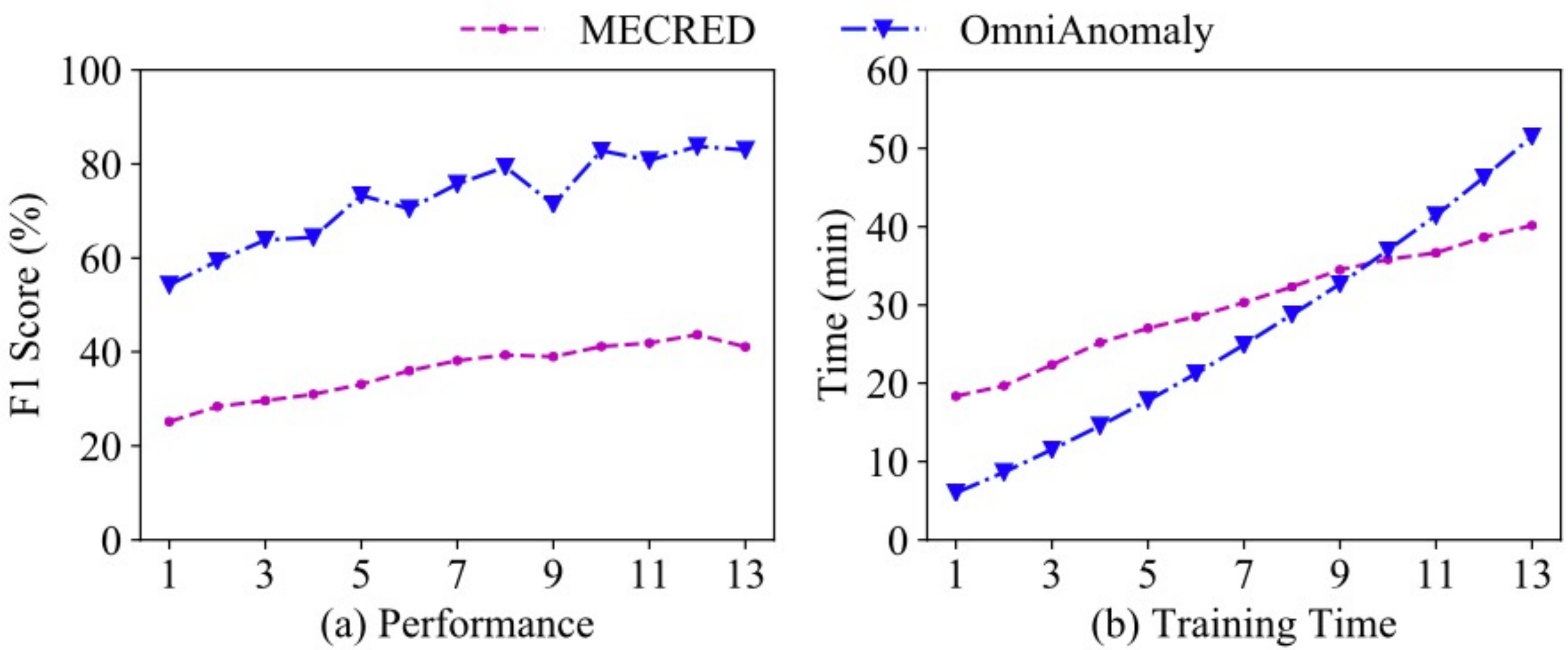
* denotes UTS anomaly detector, which can be used for MTS by combining it with majority vote

Inappropriate for newly deployed or updated systems

Incremental Retraining



Incremental Retraining Cannot Ensure Satisfactory Performance



Non-robustness and considerable training cost

Outline

The drawback of deep learning based approaches
➔ Long initialization time

Our key idea of compressed sensing and its challenges

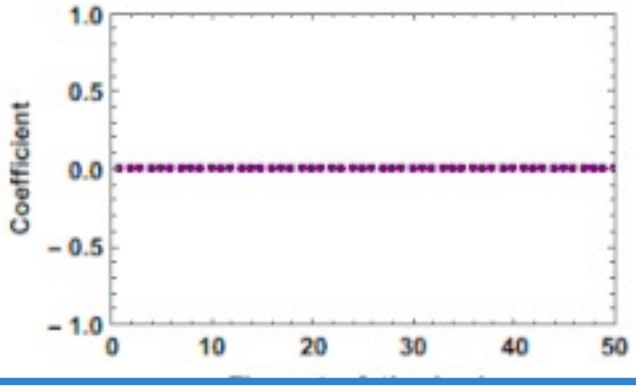
JumpStarter approach

Evaluation

Key Idea: Compressed Sensing (CS)

- CS can reconstruct time series with low energy components.
- Anomalies are always high energy components.
- CS uses a fixed-length window to initialize.

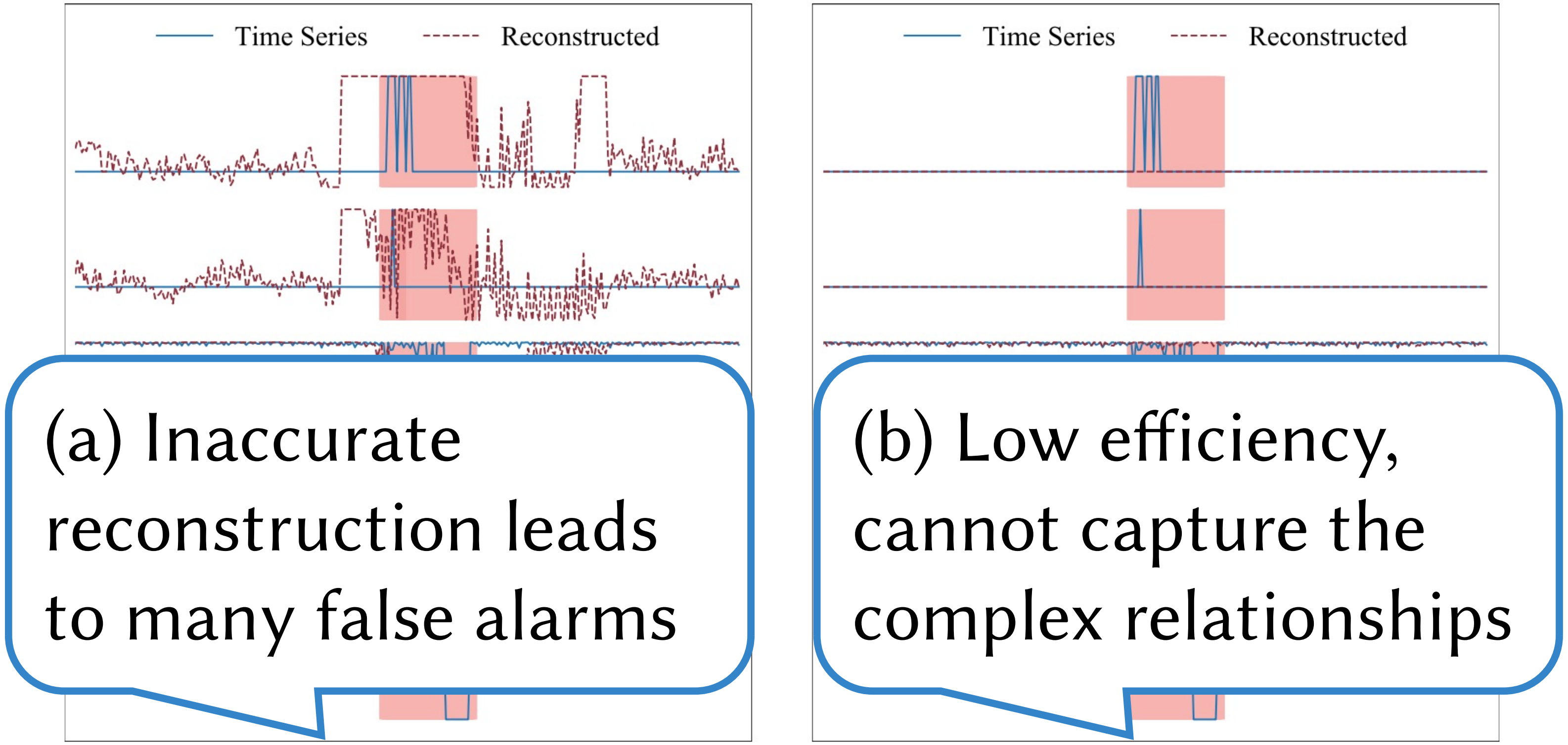
[wikipedia]



First attempt to use CS for multivariate time series anomaly detection

Two Strawman Solutions Using CS

Examples of CS-based anomaly detection when the MTS is reconstructed as a whole matrix (a) or as separate UTS (b)

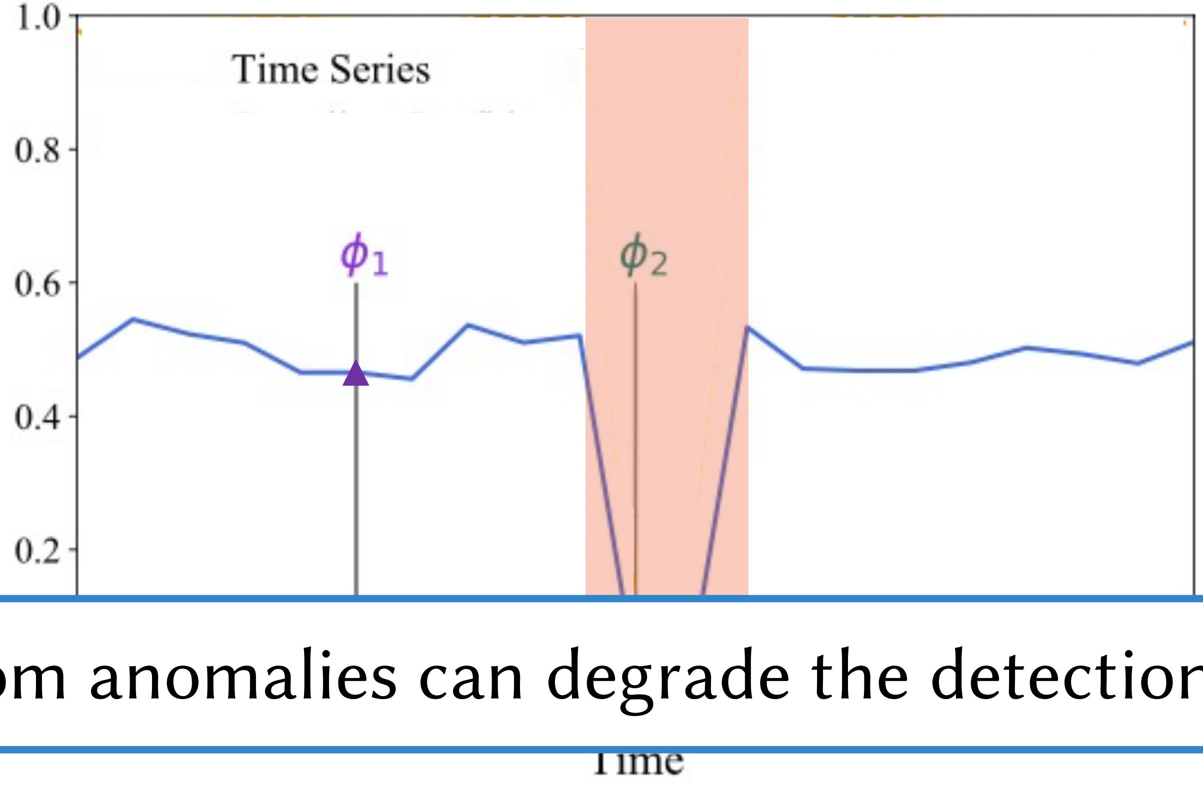


(a) As a Whole   (b) Separately

Problem of Random Gaussian Sampling

- The sampled matrix: guarantee Restricted Isometry Property (RIP)

[Information Theory 15]



Sampling from anomalies can degrade the detection performance

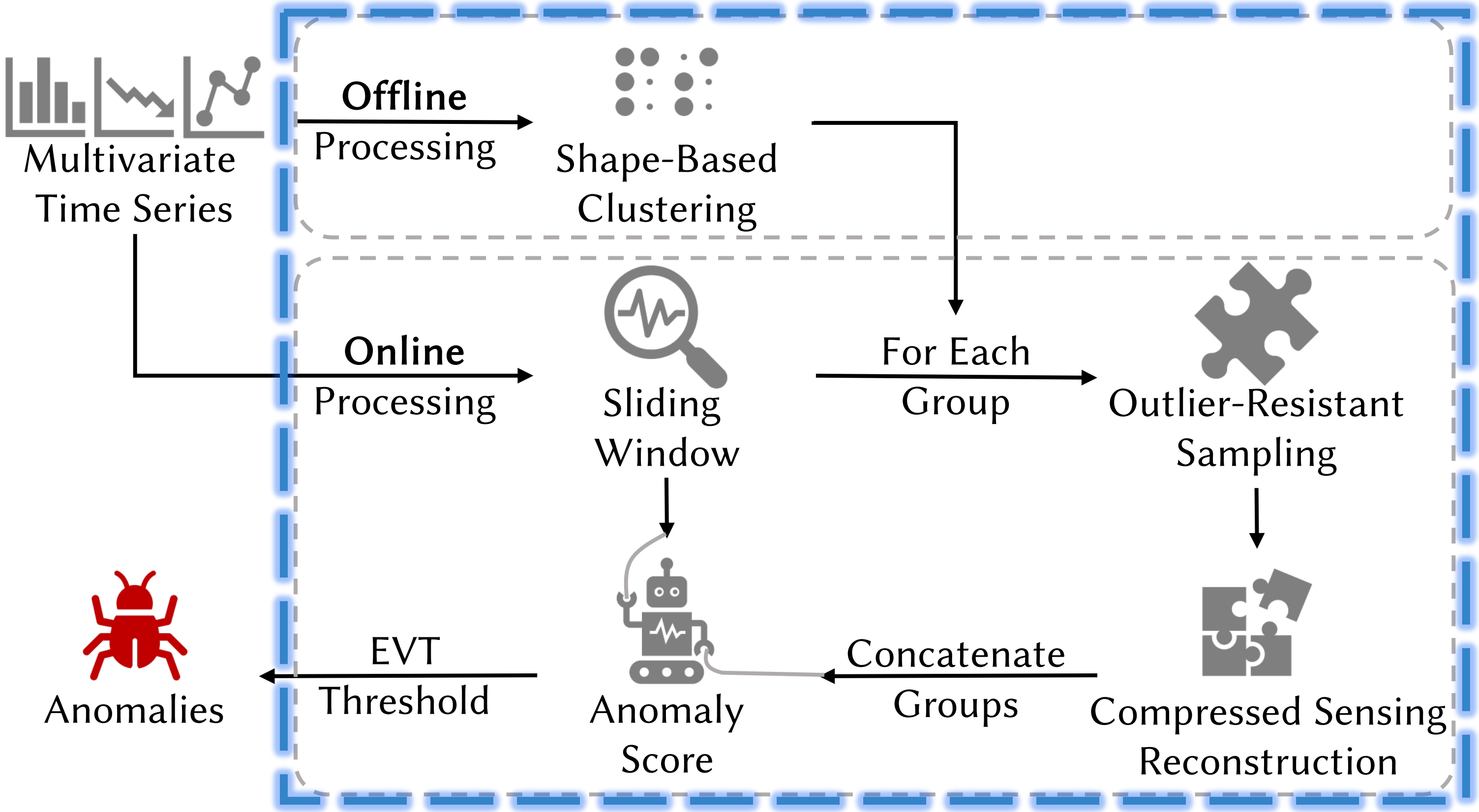
JumpStarter 🚀

Jump-Starting Multivariate Time Series

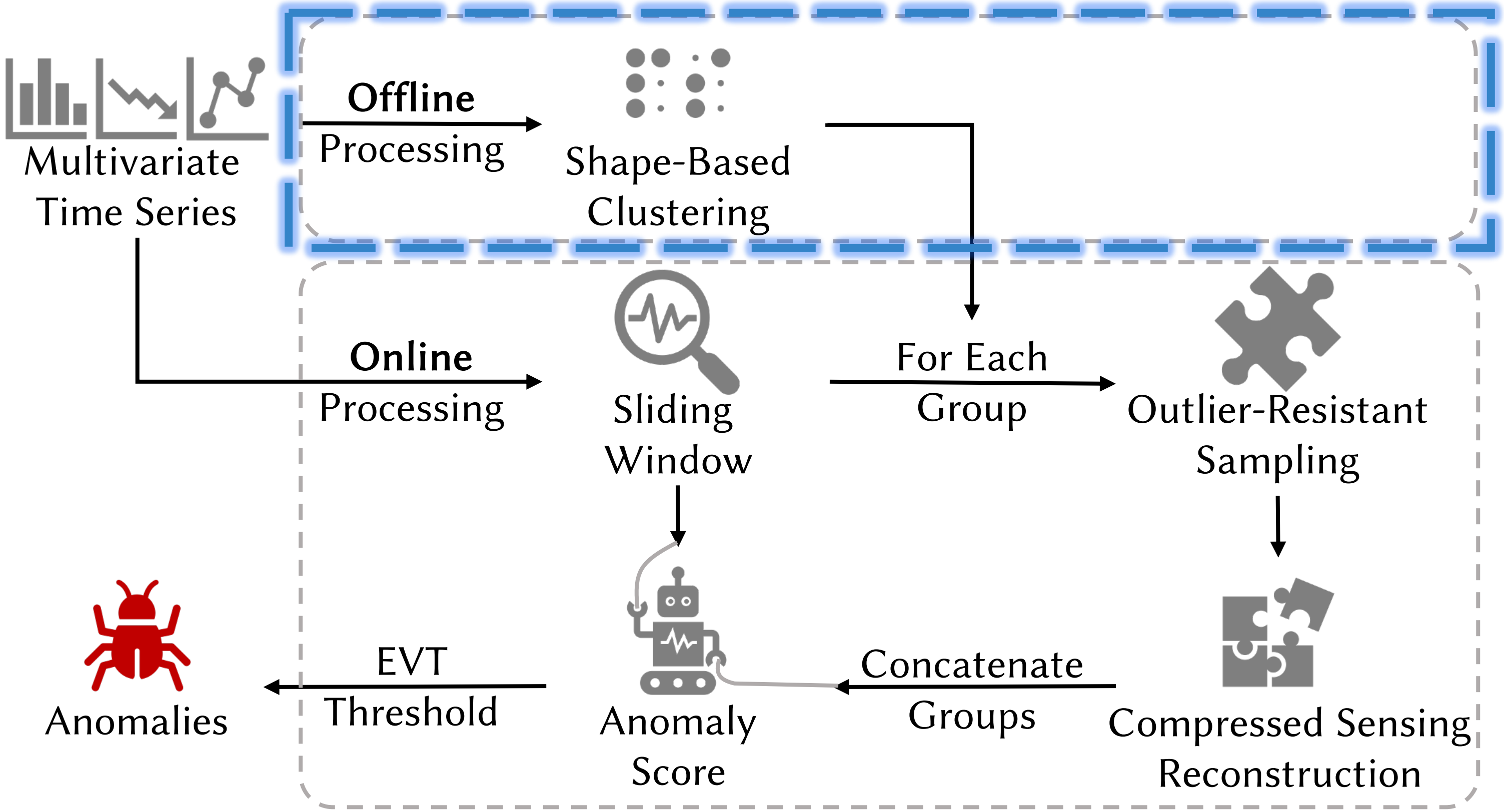
Anomaly Detection

for Online Service Systems

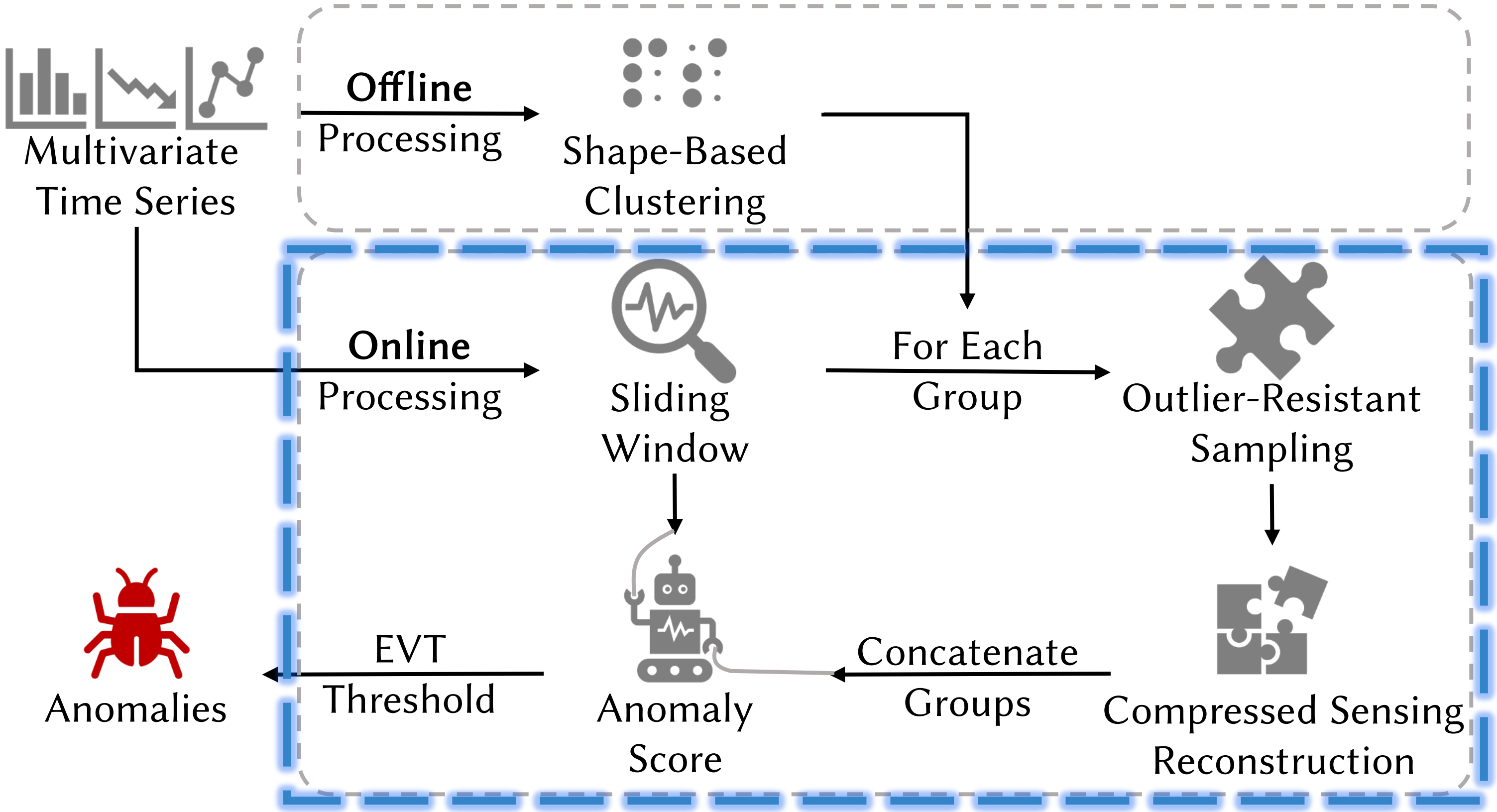
JumpStarter Overview



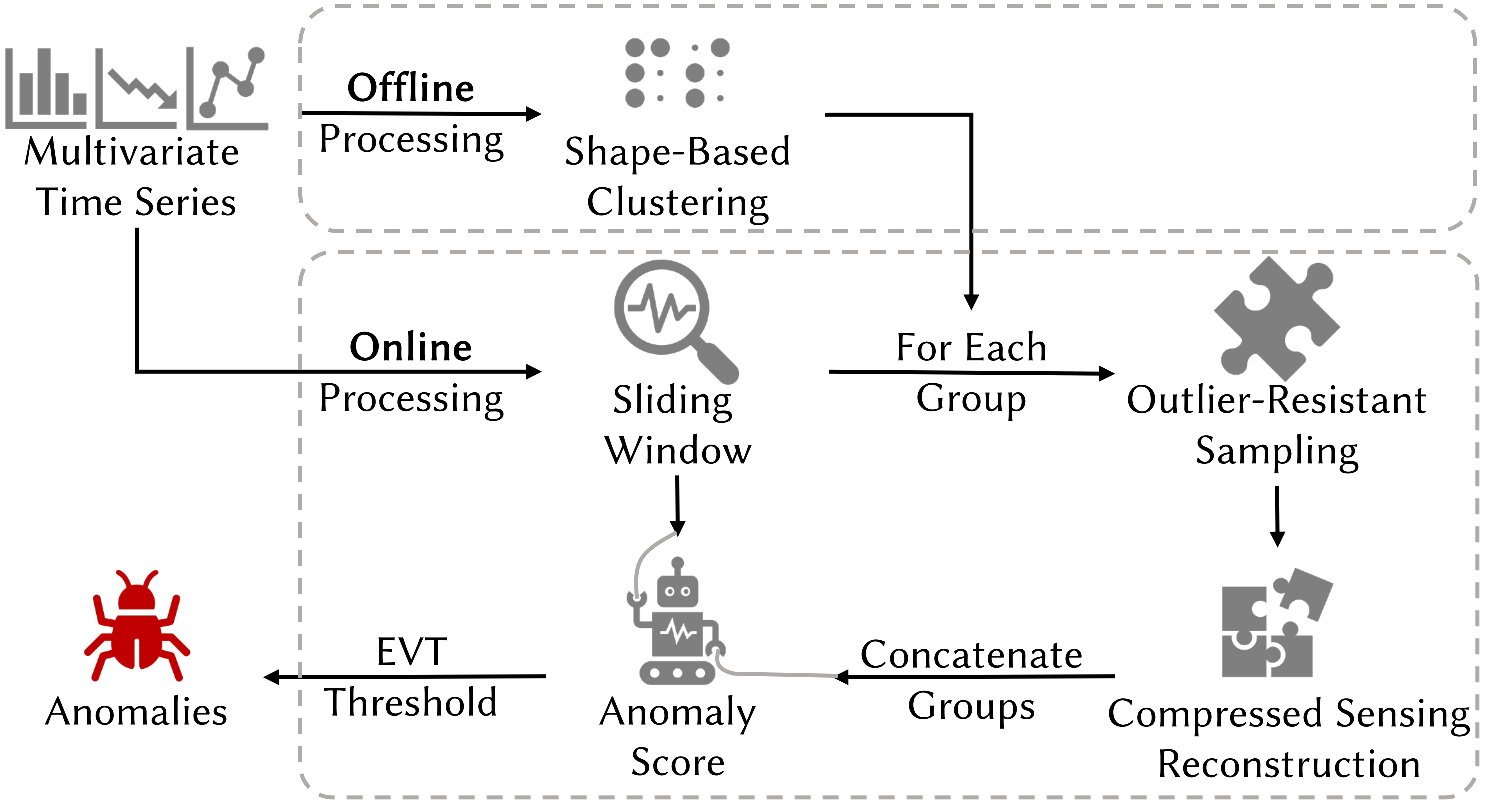
JumpStarter Overview



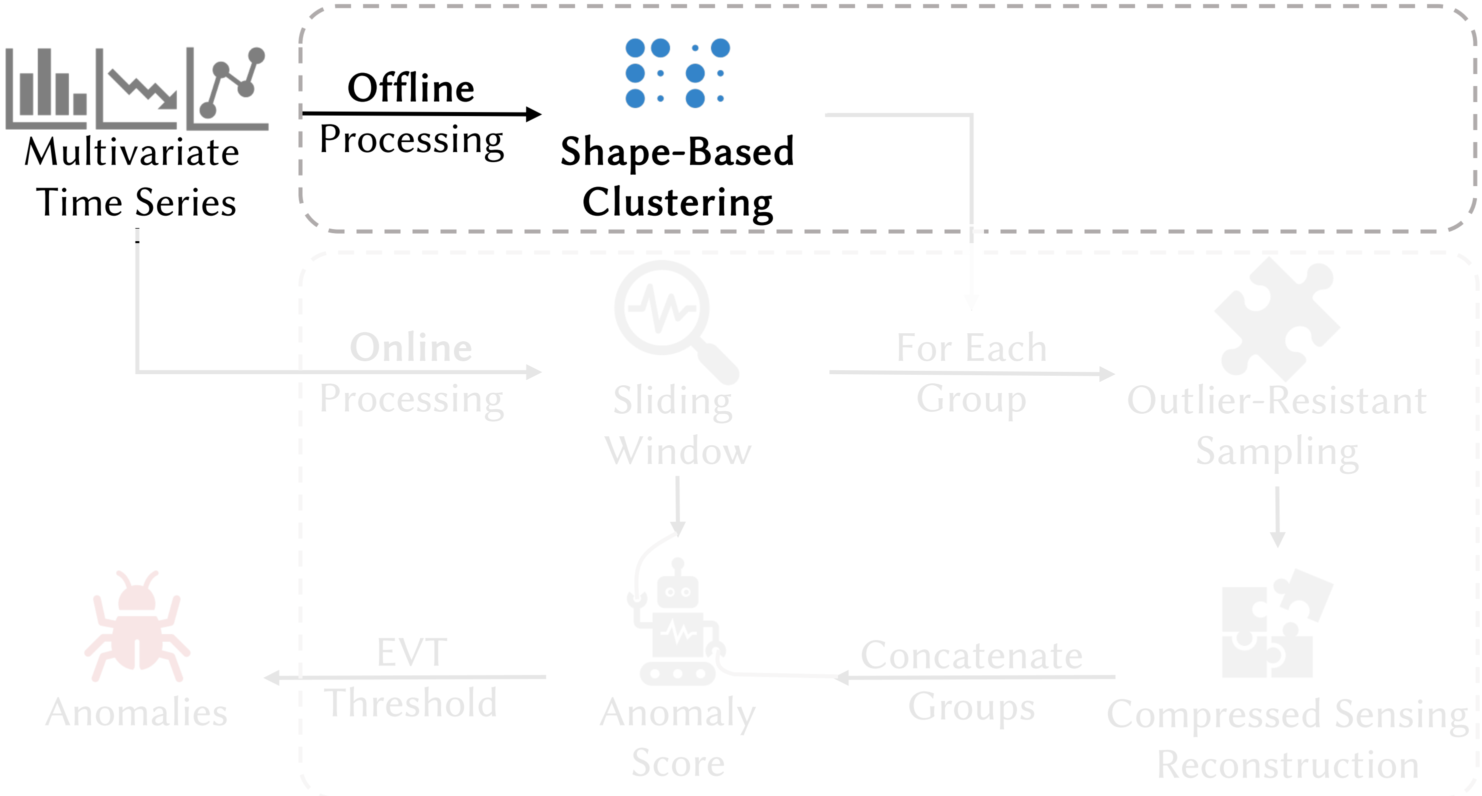
JumpStarter Overview



JumpStarter Overview

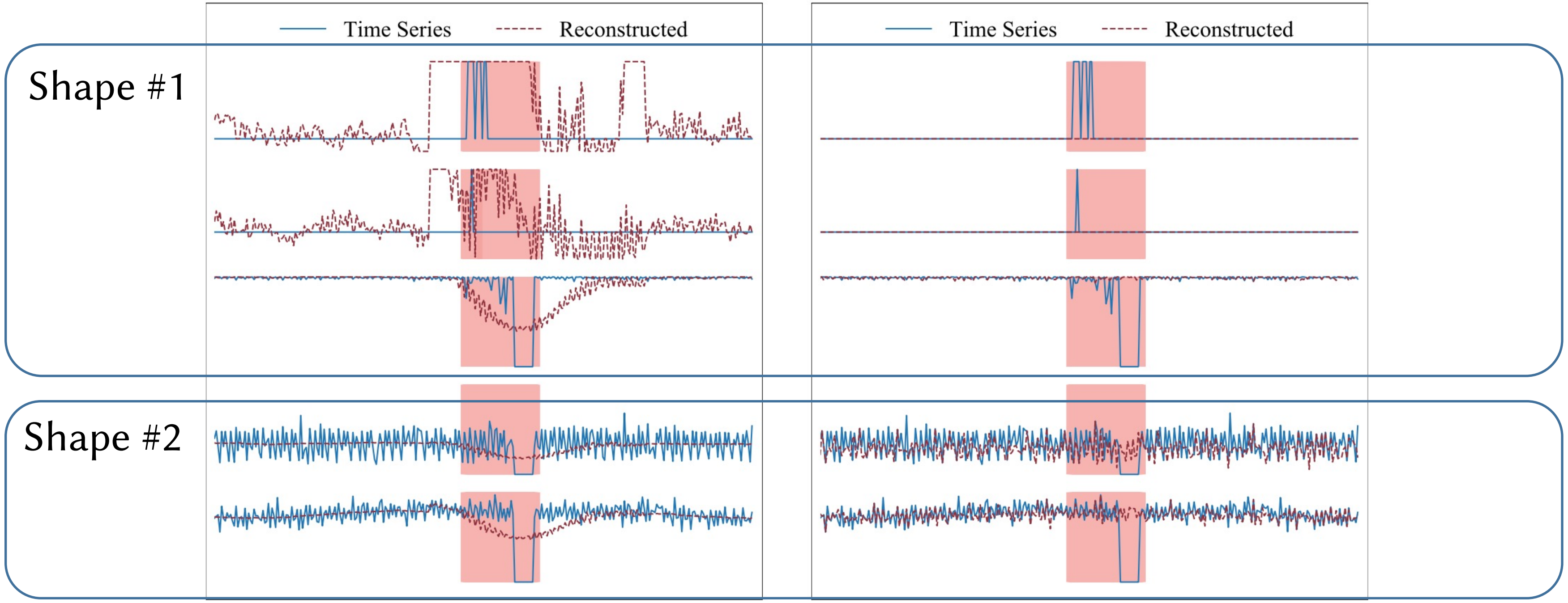


JumpStarter Offline Processing



Shape-Based Clustering

- Strawman (a) cannot deal with different shapes of time series
- Shape-based distance [sigmod15] + hierarchical clustering



(a) As a Whole (b) Separately

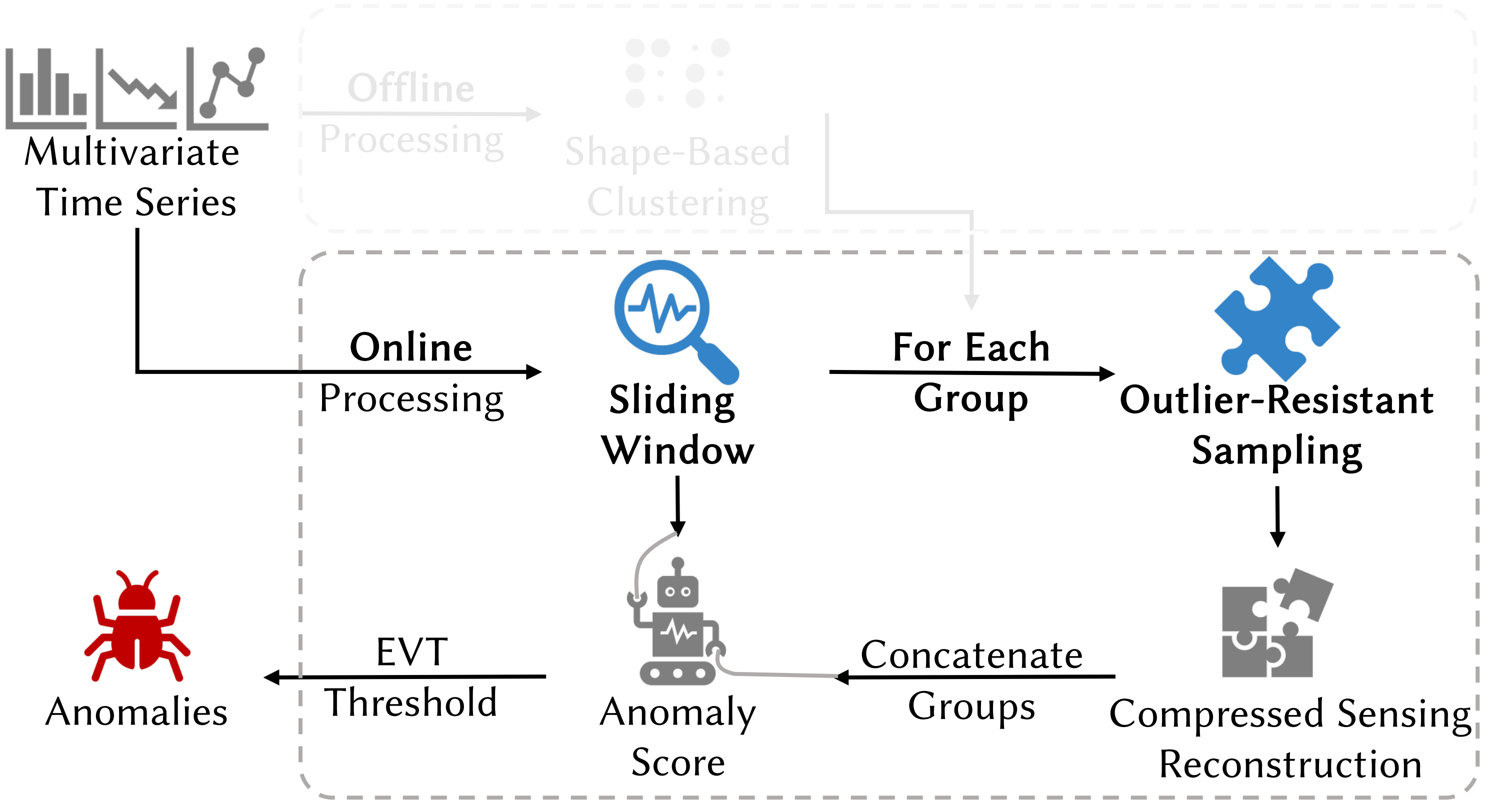
Shape-Based Clustering

- Strawman (a) cannot deal with different shapes of time series
- Shape-based distance [sigmod15] + hierarchical clustering

An example of clustering the MTS into three clusters

#	Cluster of Univariate Time Series	Explanation
1	rx-pkts-eth0, rx-bytes-eth0	# received packets/bytes
2	tcp-insegs, tcp-outsegs, tx-pkts-eth0	TCP network metrics
3	cpu-ctxt, cpu-user, cpu-system, cpu-nice	CPU utilization metrics

JumpStarter Online Processing

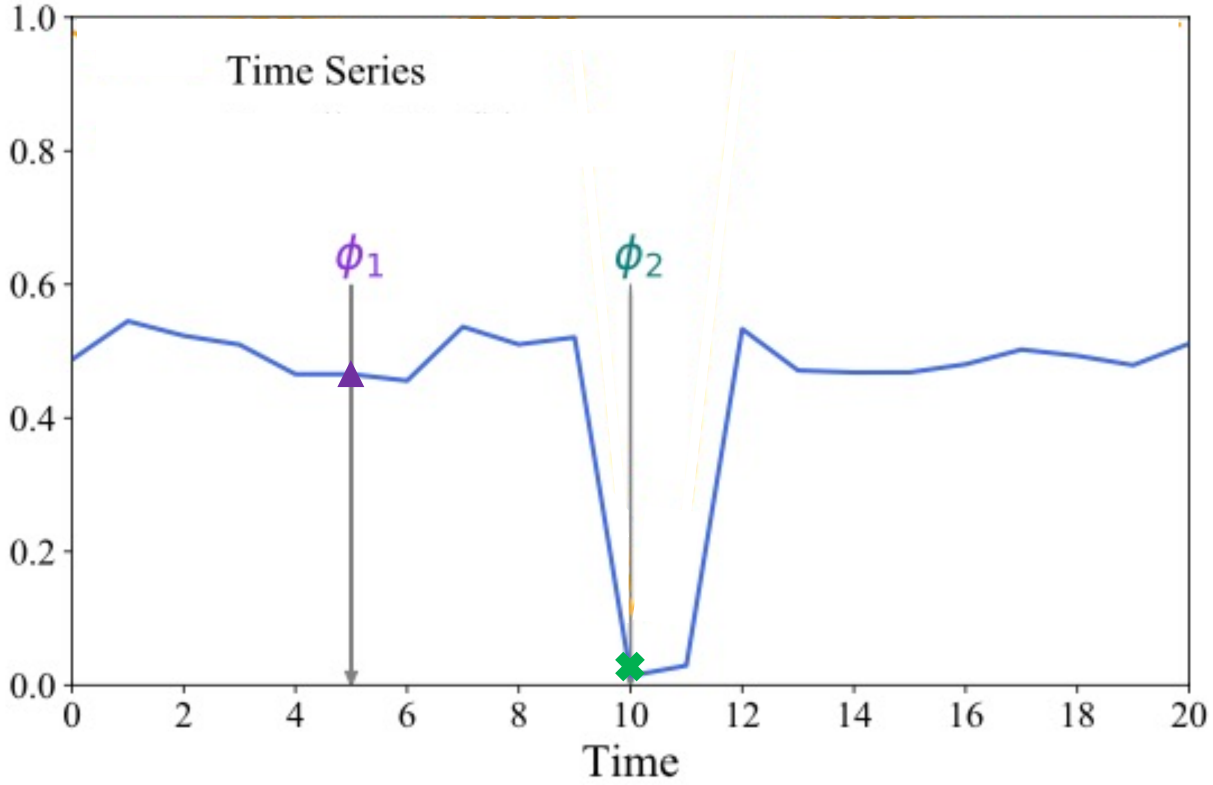


Outlier-Resistant Sampling

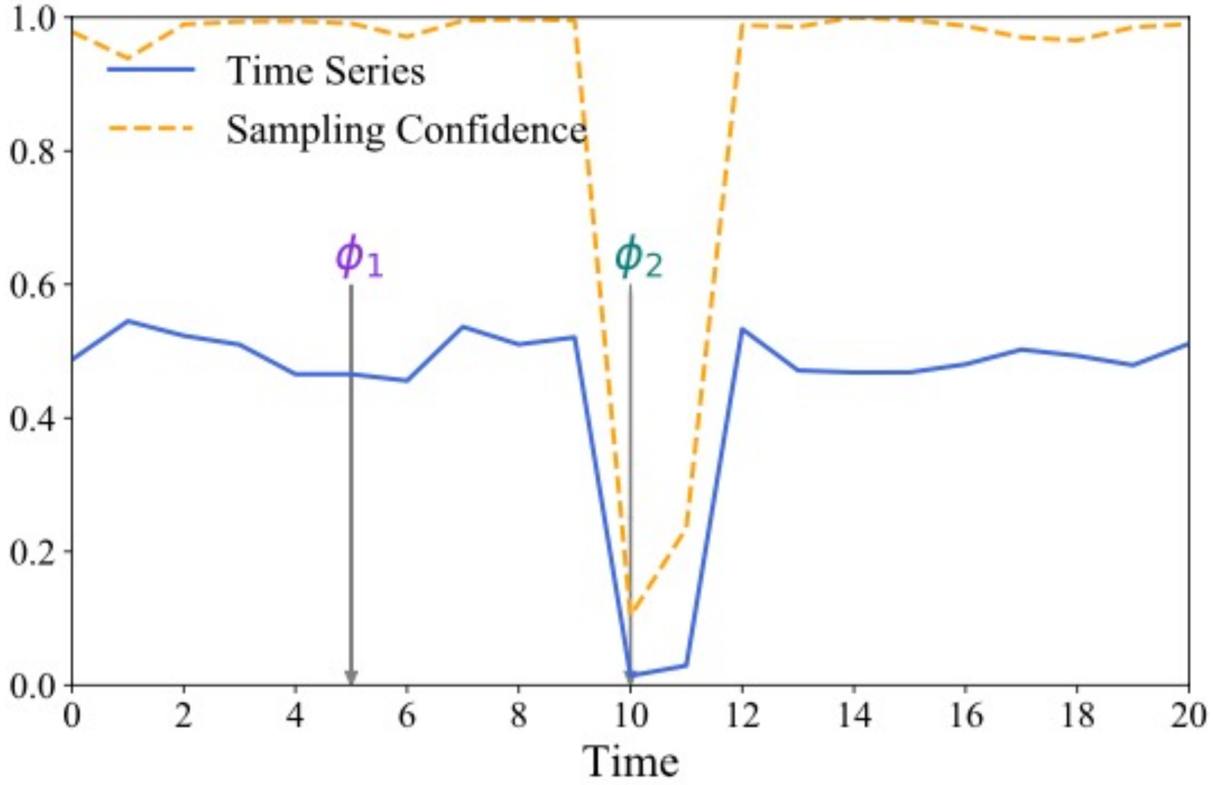
Domain-specific insights:

- Anomalies are usually outliers in an observation window.
- The value of time series has time locality.

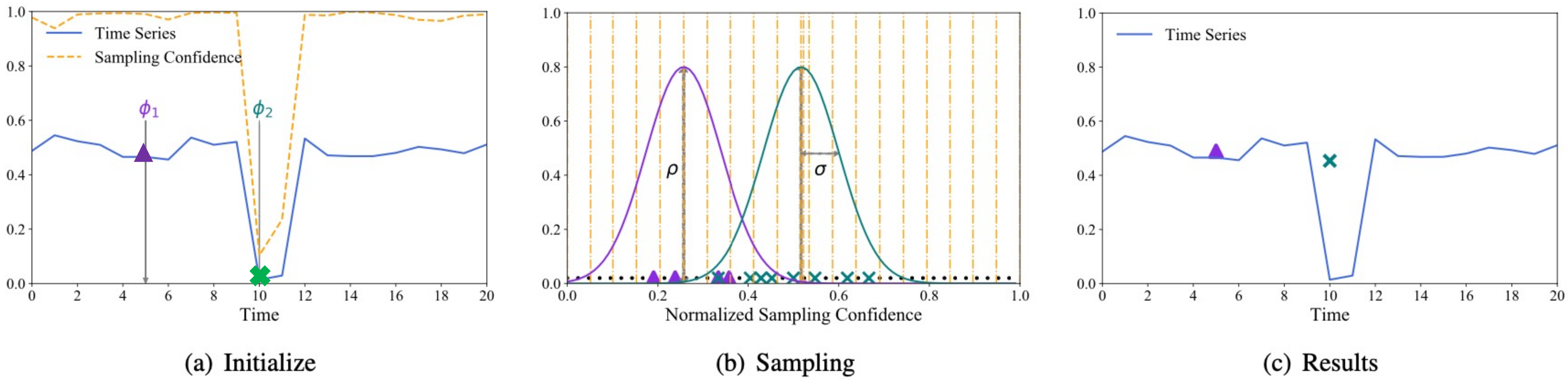
Outlier-Resistant Sampling



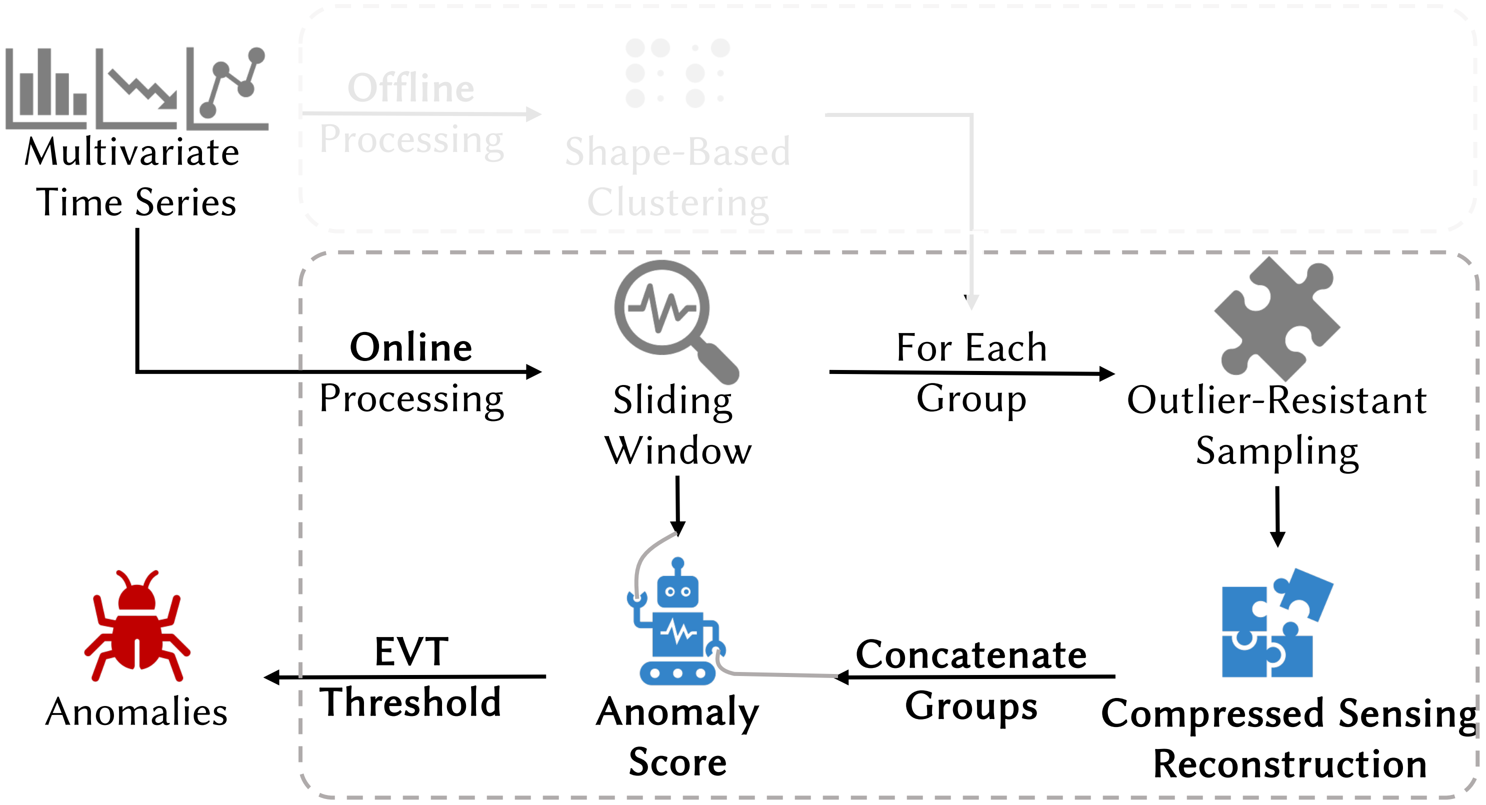
Outlier-Resistant Sampling



Outlier-Resistant Sampling



JumpStarter Online Processing



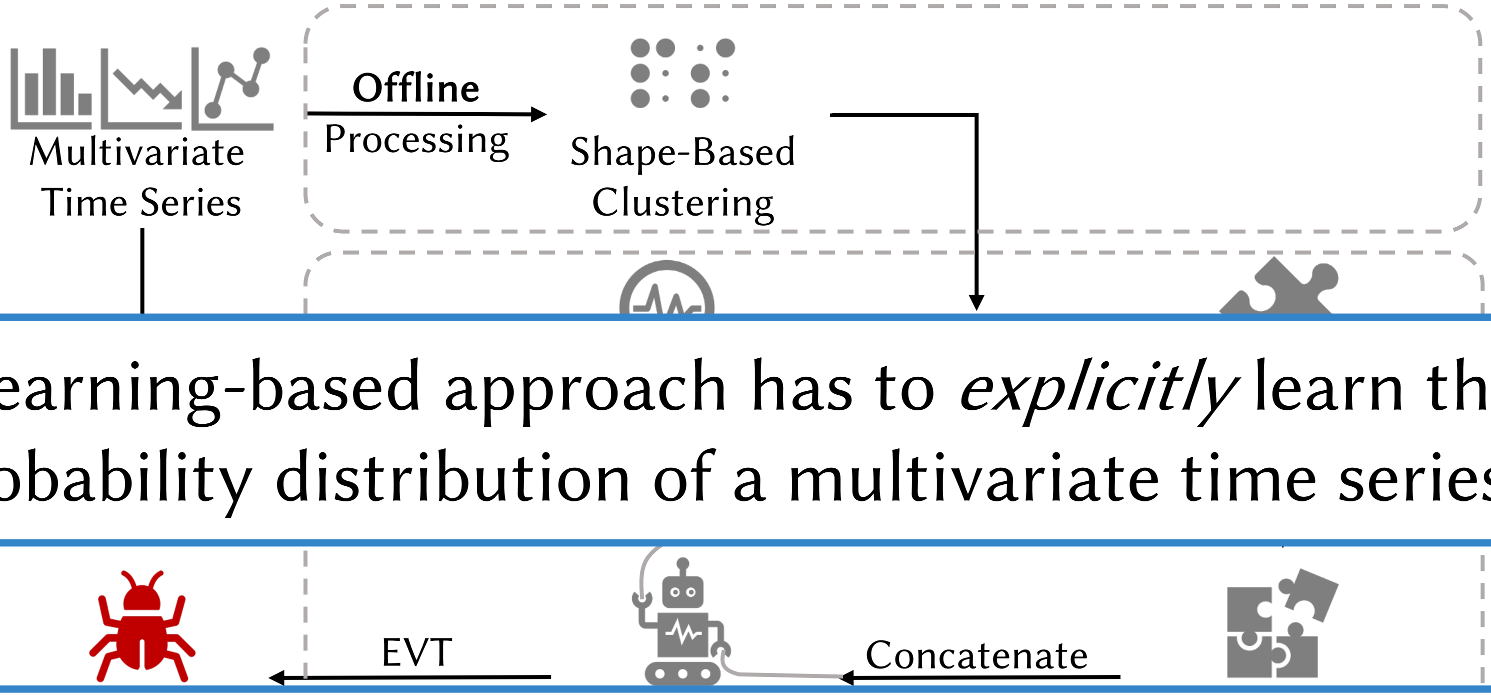
Compressed Sensing Reconstruction

- Multivariate time series: $\mathbf{X}_t = [\mathbf{x}_t^1, \mathbf{x}_t^2 ..., \mathbf{x}_t^n]^{\mathrm{T}}$
- Compressed sensing reconstruction: $\mathbf{A}\mathbf{X'}_t = \mathbf{B}$, calculating $\mathbf{X'}_t$
  - A is calculated as: $\mathbf{A} = \phi(\mathbf{D} \otimes \mathbf{D}^{\mathbf{T}})$, D is the transform of $\mathbf{X}_t$
  - B is the sampling result
- Calculation: CVXPY (convex optimization tool) [JMLR16]

- Anomaly score: measuring the differences between $\mathbf{X}_t$ and $\mathbf{X'}_t$

- Choosing threshold: Extreme Value Theory (EVT) [KDD17]

JumpStarter Initialization Time: 20 mins



A learning-based approach has to *explicitly* learn the probability distribution of a multivariate time series

Our JumpStarter: the reconstructed multivariate time series *implicitly* inherits the normal behavior

Outline

The drawback of deep learning based approaches
➔ Long initialization time

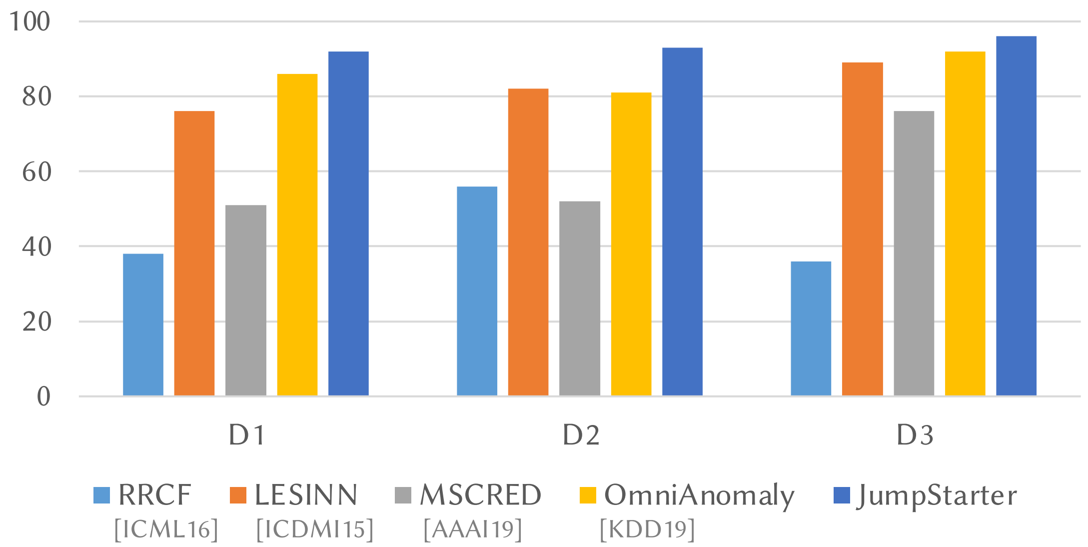
Our key idea of compressed sensing and its challenges
➔ Reconstruction & Sampling

JumpStarter approach
➔ Shape-Based Clustering & Outlier-Resistant Sampling

Evaluation
➔ Company A (28 service systems) & Company B (30 service systems)

Evaluation: Accuracy

Average F1 Score of JumpStarter and baseline approaches

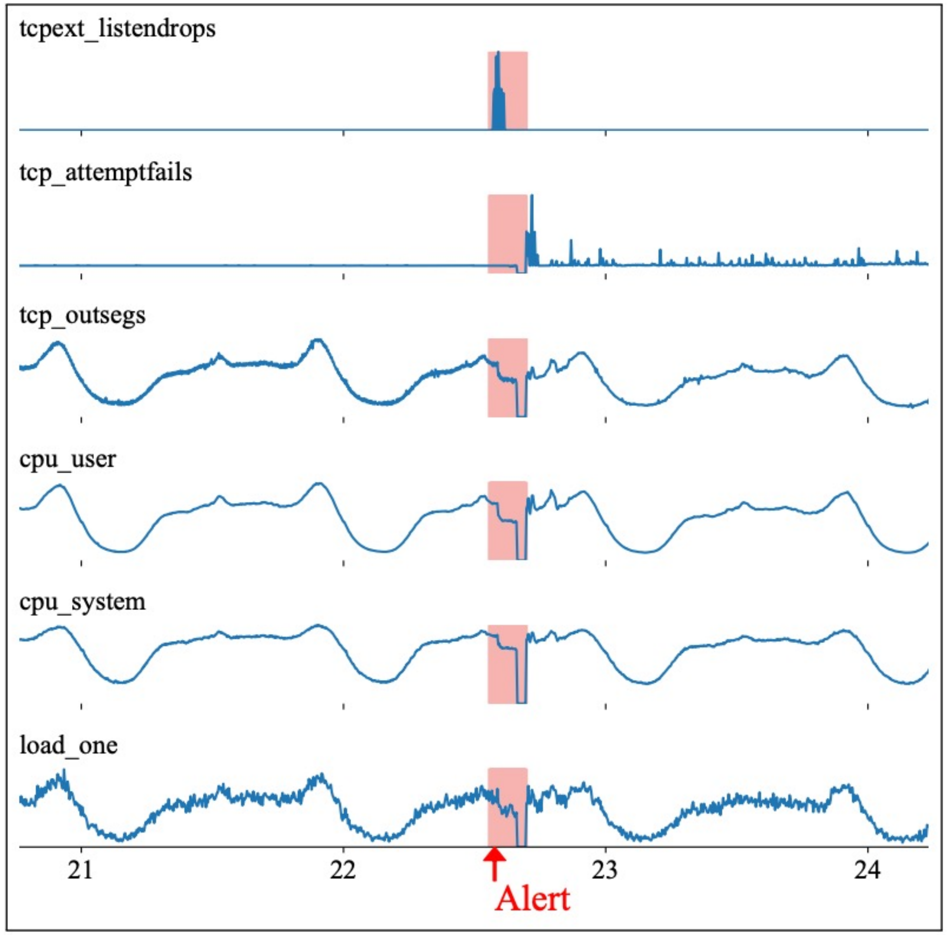


Evaluation: Efficiency

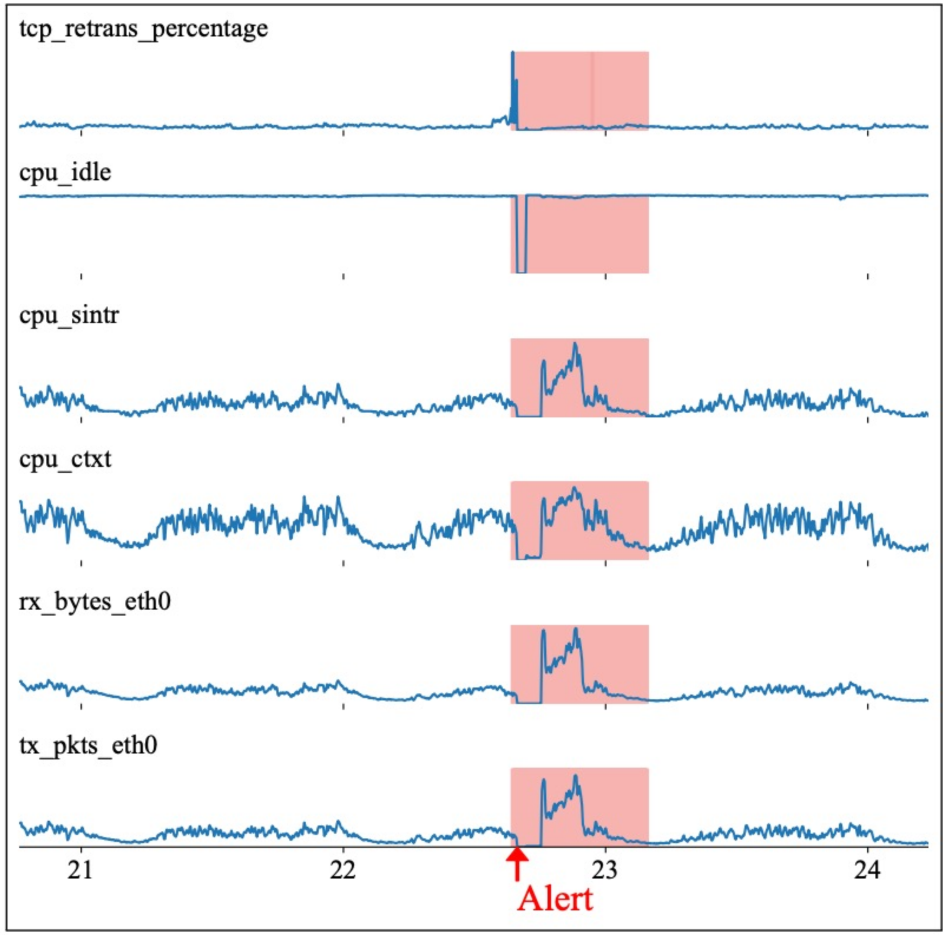
The initialization time (IT) and detection time (DT) comparison

Approach	RRCF	LESINN	MSCRED	Omni-Anomaly	*JumpStarter*
IT (min)	20	20	>86400	>86400	20
DT (ms)	41.24	118.63	122.82	191.86	127.13

Case Study



(a) Network Issue



(b) Software Change

Conclusion

To adapt to frequent changes in online service systems, multivariate time series, anomaly detection should be robust and can be **quickly initialized**.

JumpStarter adopts the **Compressed Sensing** technique

- Reconstruction challenge ➔ **Shape-based clustering**
- Sampling challenge ➔ **Outlier-resistant sampling**

Evaluation

- Real-world online service systems of two Internet companies
- Achieving an average F1 score of 94.1%, initialization time 20 minutes
- https://github.com/NetManAIOps/JumpStarter

Thanks

mmh16@mails.tsinghua.edu.cn