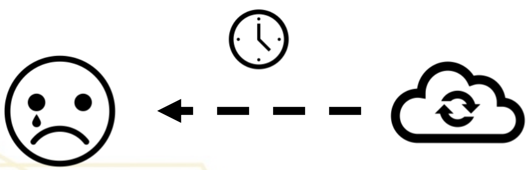
Lyon, France, 25-29 April 2022

Robust System Instance Clustering for Large-Scale Web Services

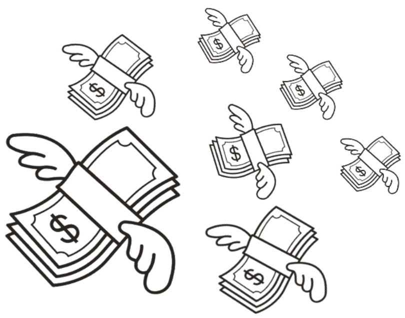
Shenglin Zhang, **Dongwen Li**, Zhenyu Zhong, Jun Zhu, Minghan Liang, Jiexi Luo, Yongqian Sun, Ya Su, Sibo Xia, Zhongyou Hu, Yuzhi Zhang, Dan Pei, Jiyan Sun, Yinlong Liu

Background

The Reliability of Web Services is Important



High Latency

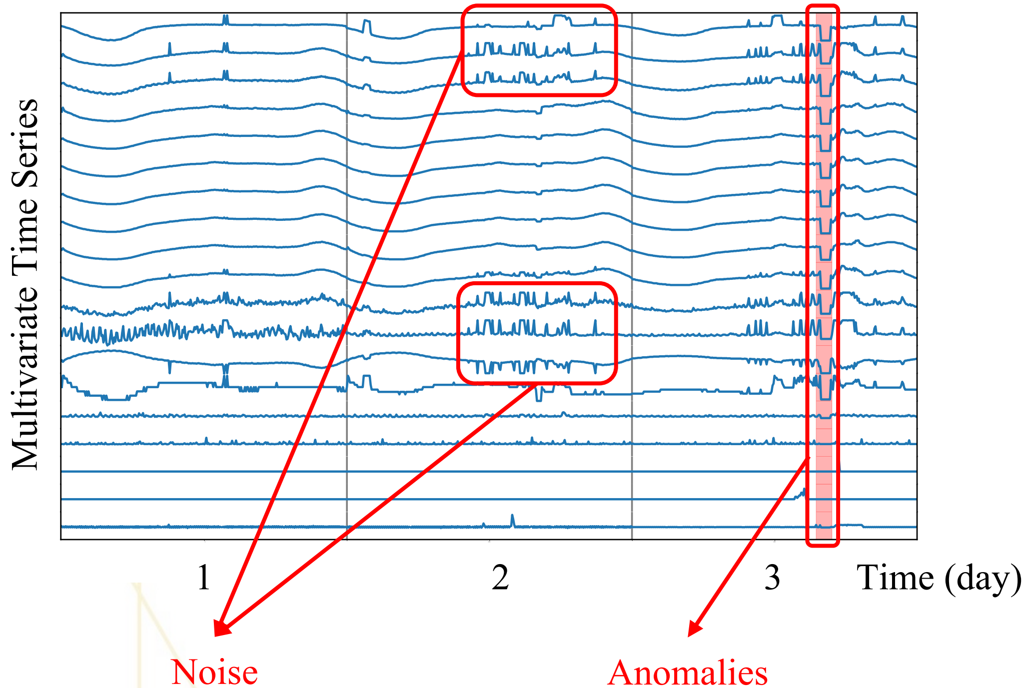


Users

Service Providers

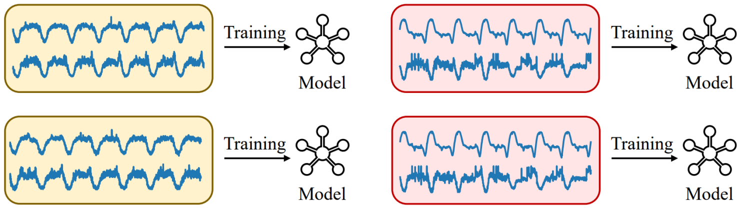
MTS Anomaly Detection

- Many **multivariate time series (MTS)** anomaly detection algorithms are proposed.
- Determine whether the behaviors of system instances deviate from the *normal patterns*.



MTS Anomaly Detection is Challenging

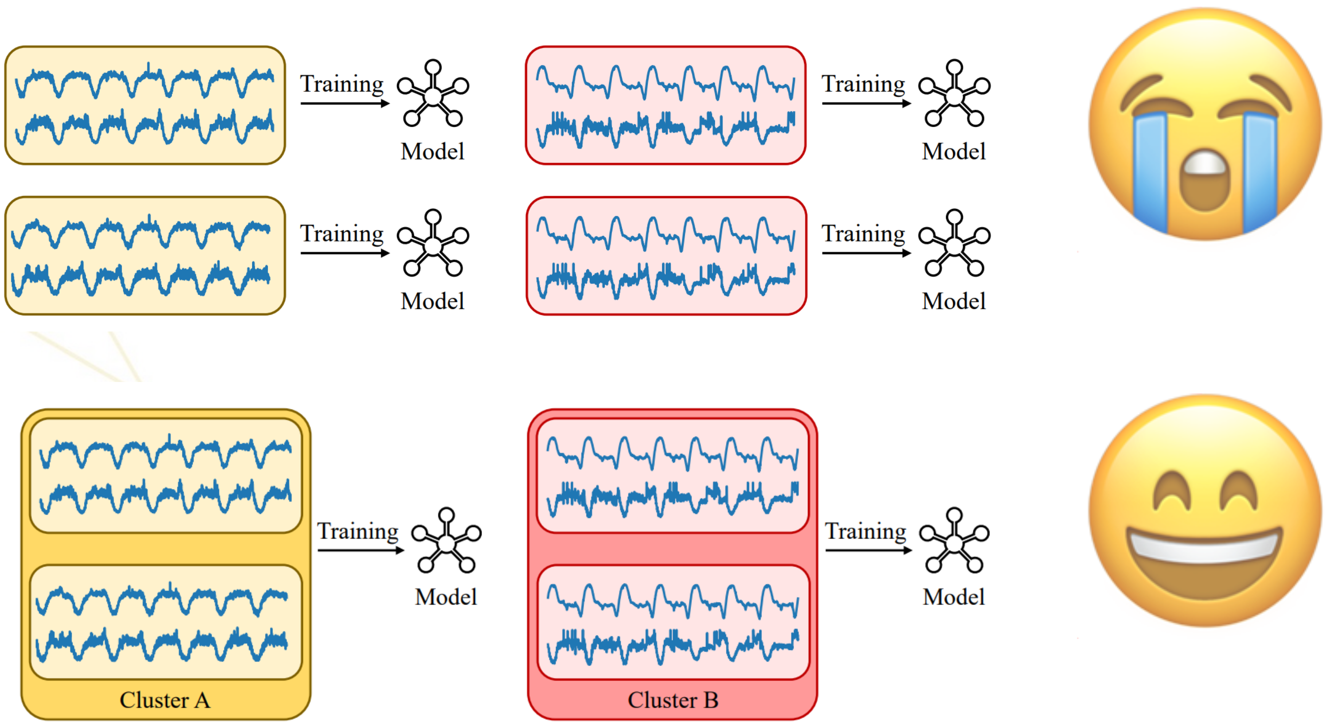
- Huge number of system instances in large-scale Web services.
- Training an MTS anomaly detection model for **each system instance** is resource-consuming.



Method	Time (1 M Instances)
USAD (KDD '20)	99.87 days
OmniAnomaly (KDD '19)	5.70 years
SDFVAE (WWW '21)	7.70 years
InterFusion (KDD '21)	15.40 years
DAGMM (ICLR '18)	206.16 days

Clustering Helps

- Training an MTS anomaly detection model for **each cluster** can significantly reduce the training overhead.



Previous Work

- **Traditional clustering methods:**

  Not suitable for such high-dimensional data.

- **Univariate time series clustering algorithms:**

  Lose important information and difficult to represent MTS data with a consistent model.

- **MTS clustering algorithms:**

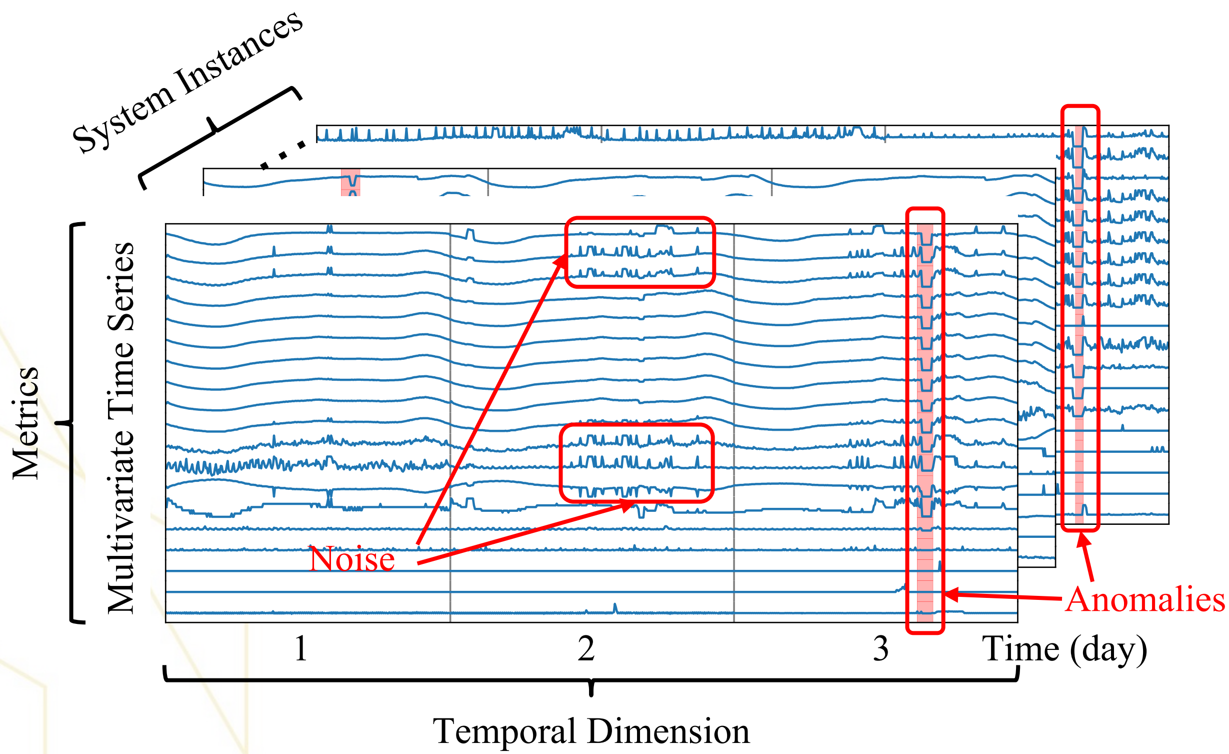
  Time- and space-consuming.

We need a more effective and efficient approach!

Challenges

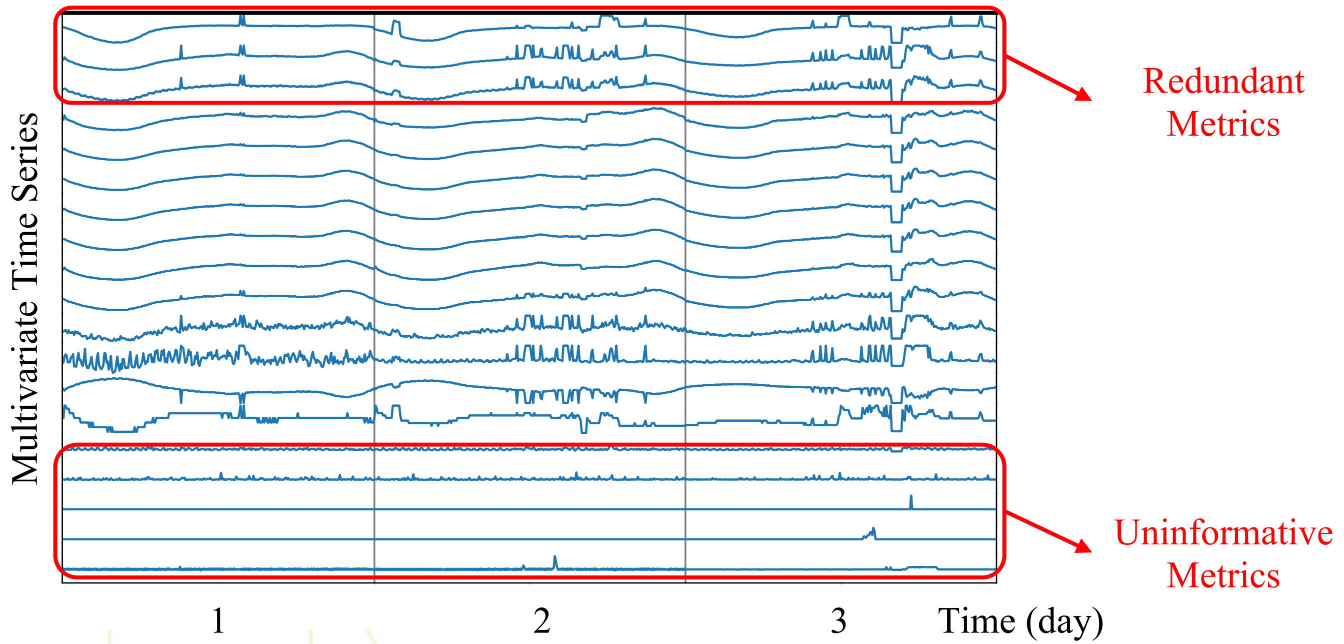
#1 Dimensionality Explosion!

- A vast number of system instances with massive time points containing noise and anomalies.



#2 Redundant and Uninformative Metrics

- Some metrics can be highly correlated.
- Non-periodic metrics are uninformative for MTS clustering.



#3 Lack of Labeling Tool

- Labeled data is required to evaluate models' performance.
- It is challenging to label such large-scale data manually without the help of a user-friendly tool.

The Solution: OmniCluster

The Core Ideas of OmniCluster

To address challenge #1:

- A vast number of system instances
- Massive time points
- Noise and anomalies

➡ Apply **1D-CAE** to extract low dimensional temporal features from each metric separately.

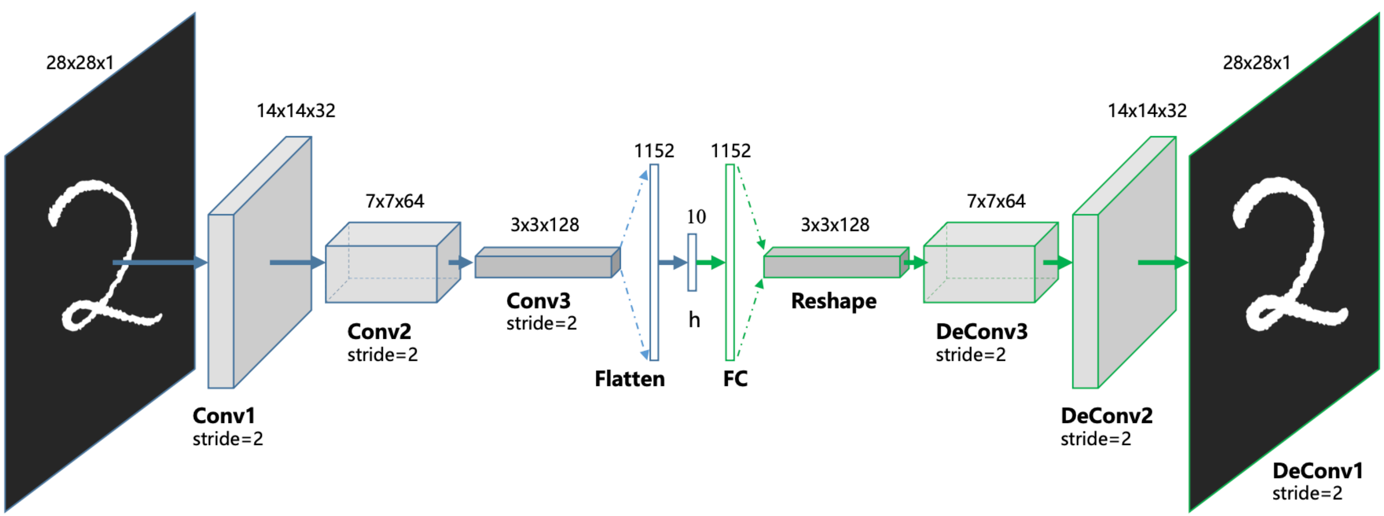
Convolutional Autoencoders (CAE)

A Convolutional Autoencoder (CAE) is a type of neural network that uses convolutional layers to encode input data into a compact representation and then decode it back to its original form. It is particularly effective for image data.

For Feature Extraction: CAE Learns useful features in an unsupervised manner, which can be used for further machine learning tasks.

Encoder: Compresses the input into a latent-space representation. It typically consists of convolutional layers followed by pooling layers which reduce spatial dimensions.

Decoder: Reconstructs the input from the latent representation. It uses deconvolutional layers (or up-sampling) to increase spatial dimensions back to that of the original input.



The Core Ideas of OmniCluster

To address challenge #2:

- Redundant and uninformative metrics

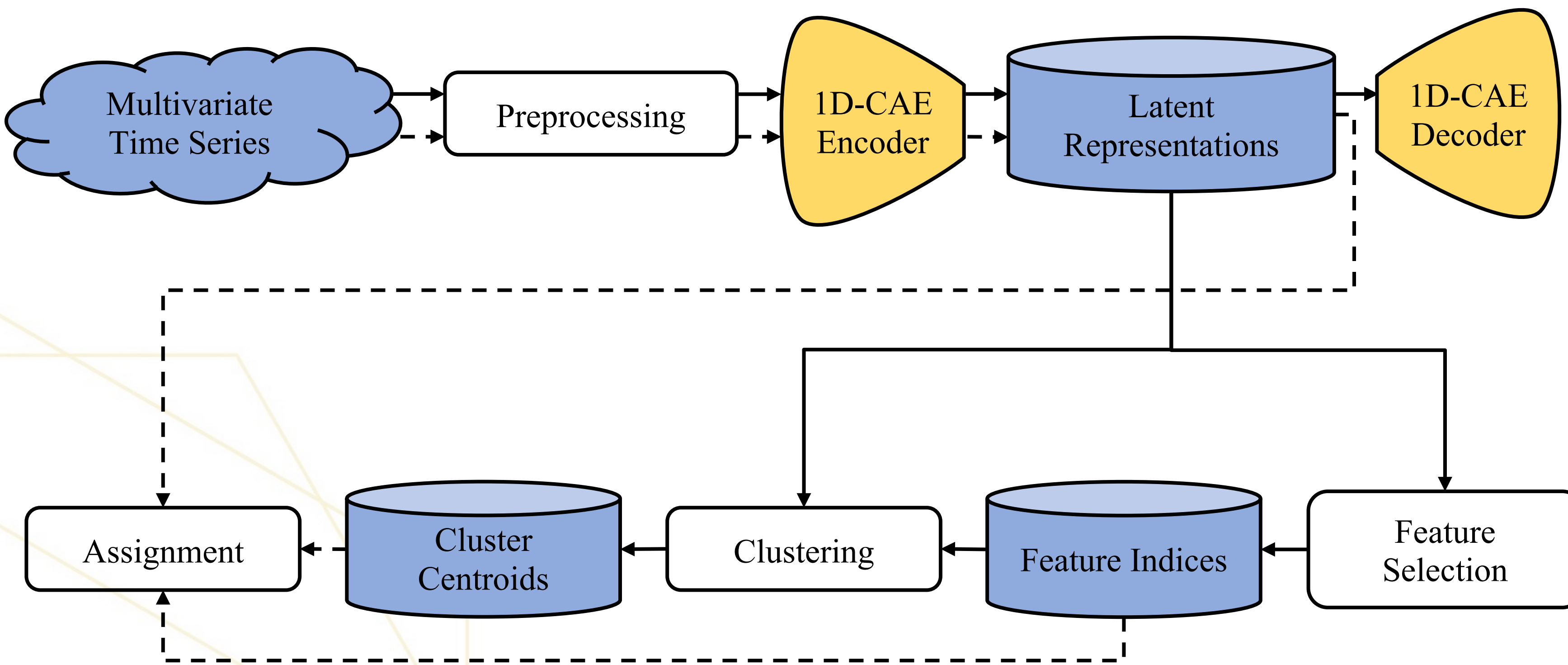
➡ Select periodic and representative metrics through **feature selection.**

The Core Ideas of OmniCluster

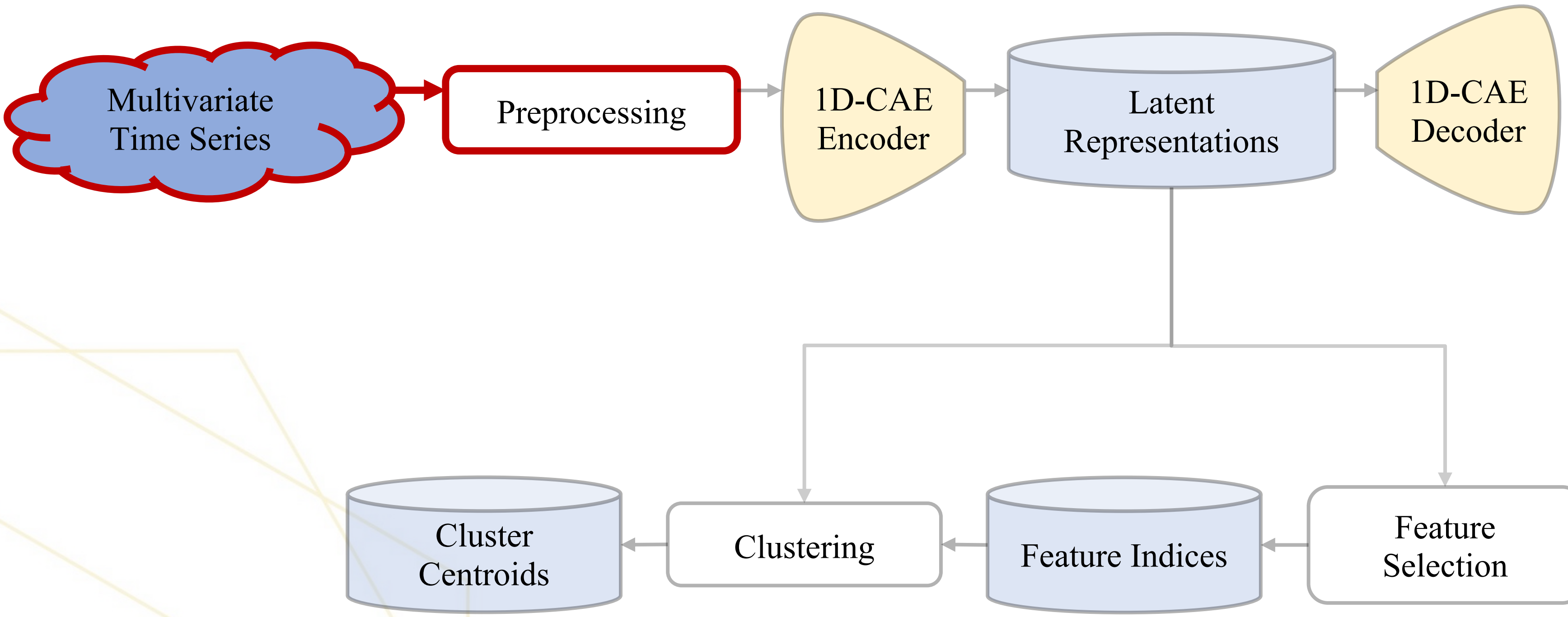
To address challenge #3:

- Lack of labeling tool ➡ Develop a **labeling tool for MTS clustering** with a user-friendly interface.

OmniCluster Overview



Offline Clustering (1/4)

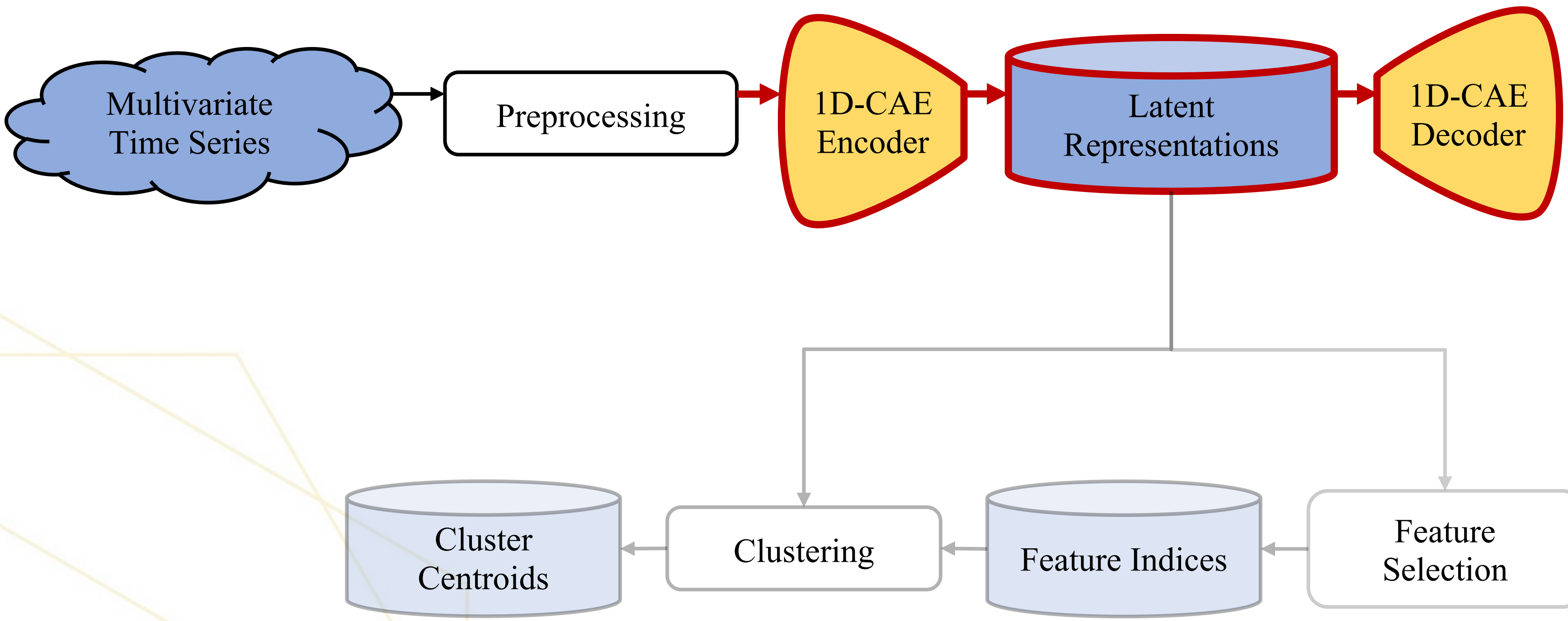


Preprocessing

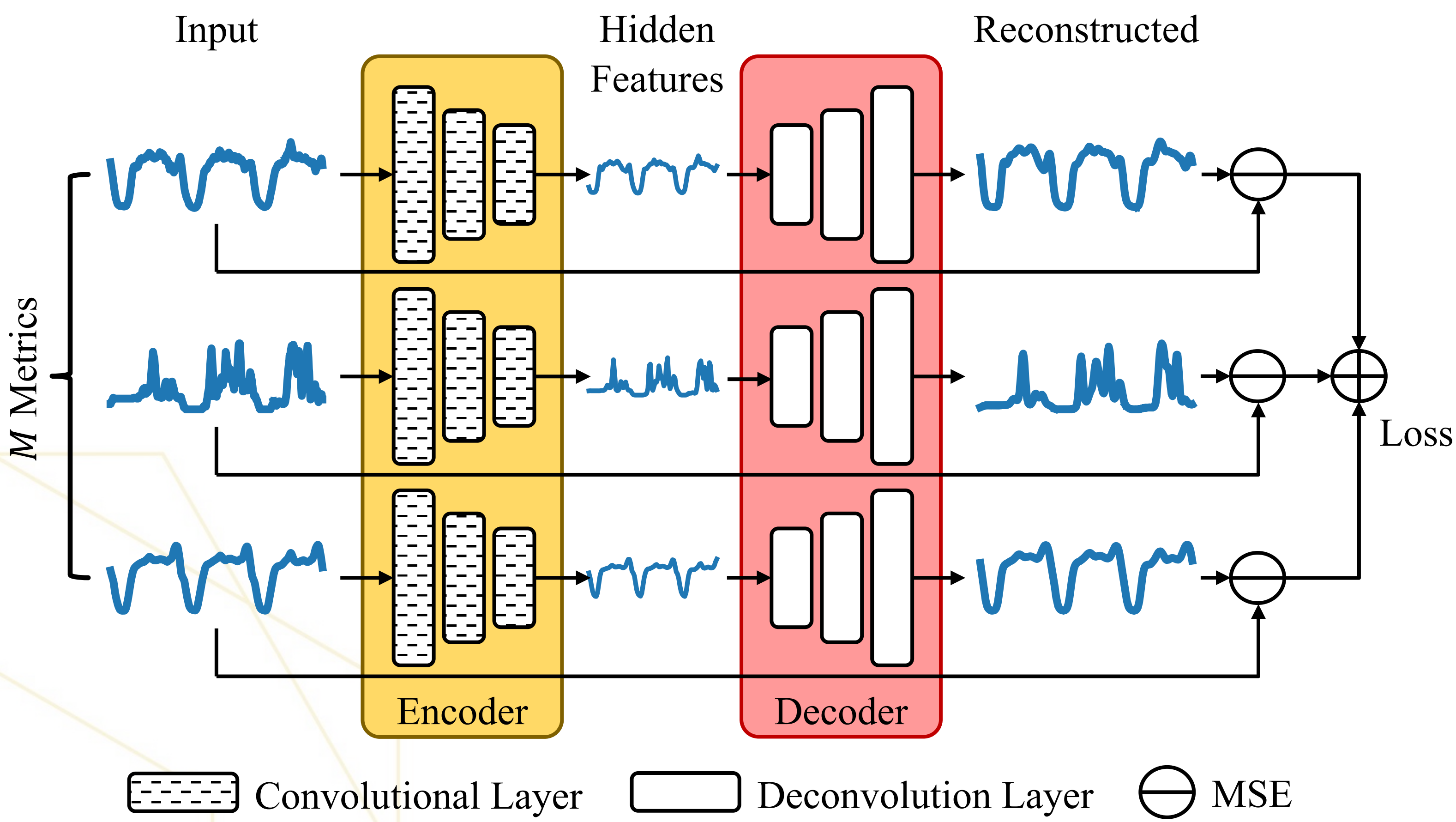
- **Remove the top 5% data** deviating from the mean value.
- **Linear interpolation** to fill the removed or missing values.
- **Moving average** with a small sliding window.
- **Normalization** to remove amplitude differences.

Minimize the negative impact brought by anomalies, noise, and missing values.

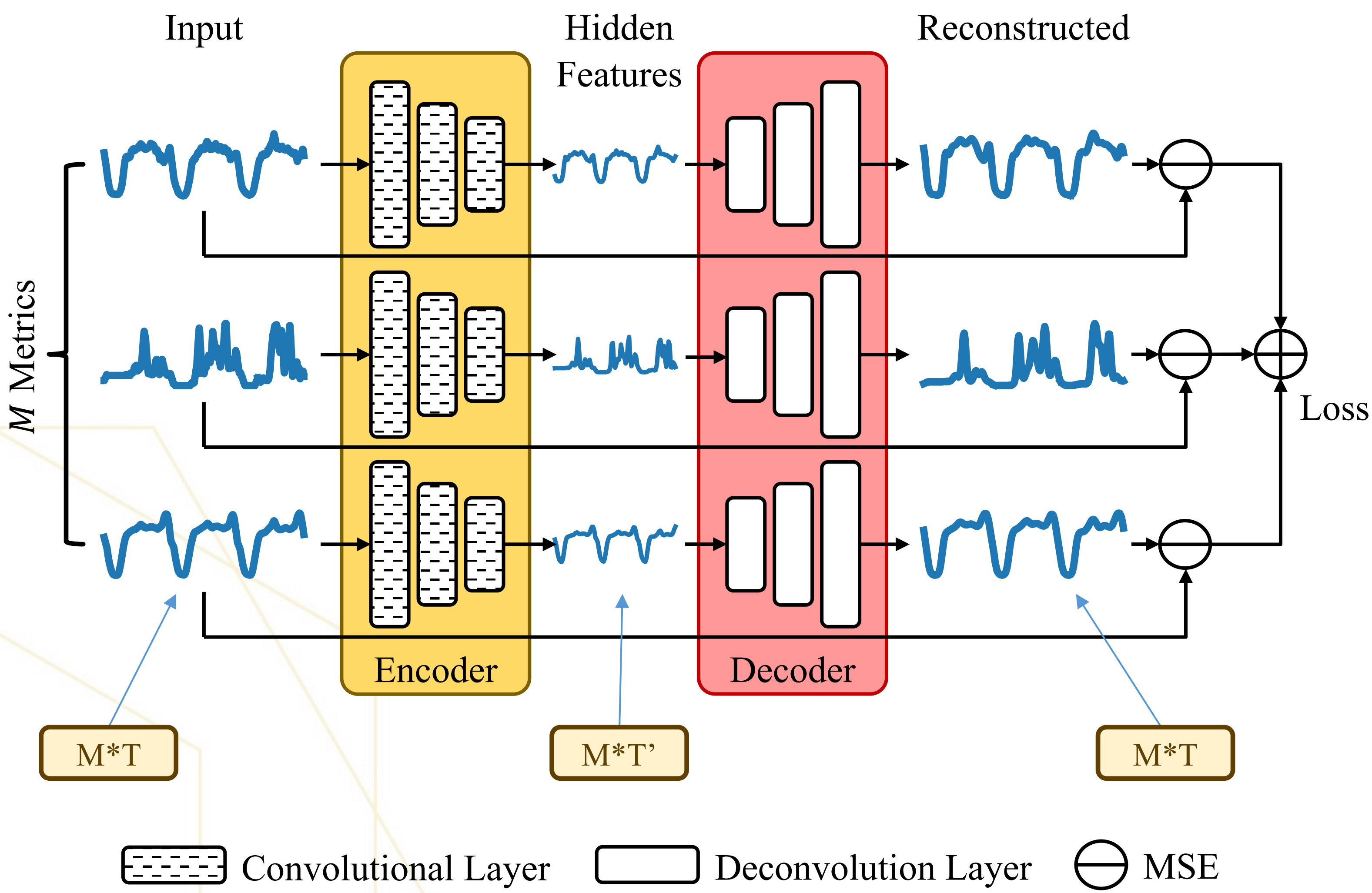
Offline Clustering (2/4)



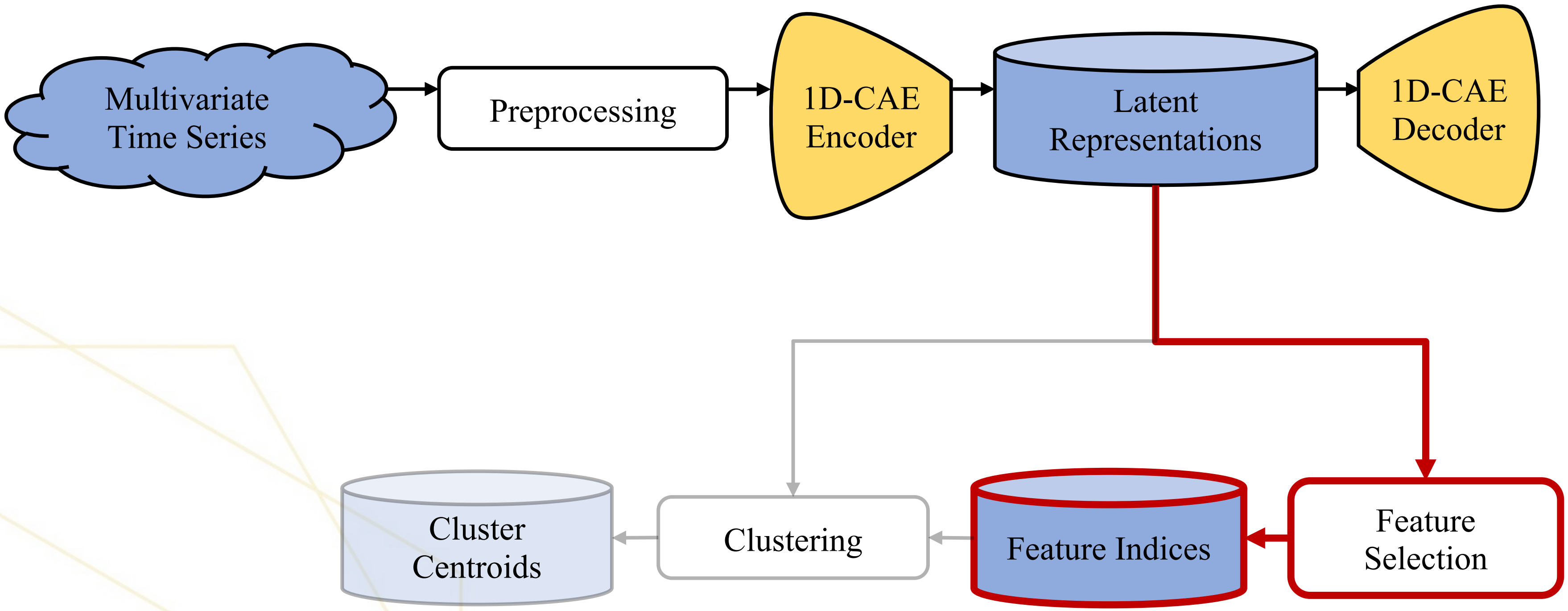
Temporal Feature Extraction



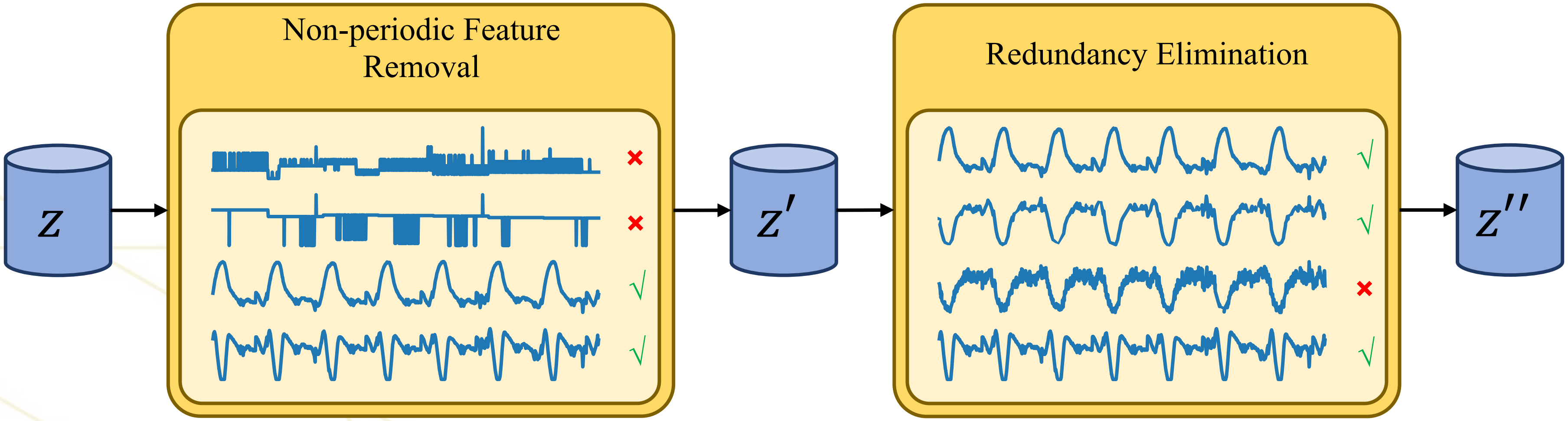
How 1D-CAE works



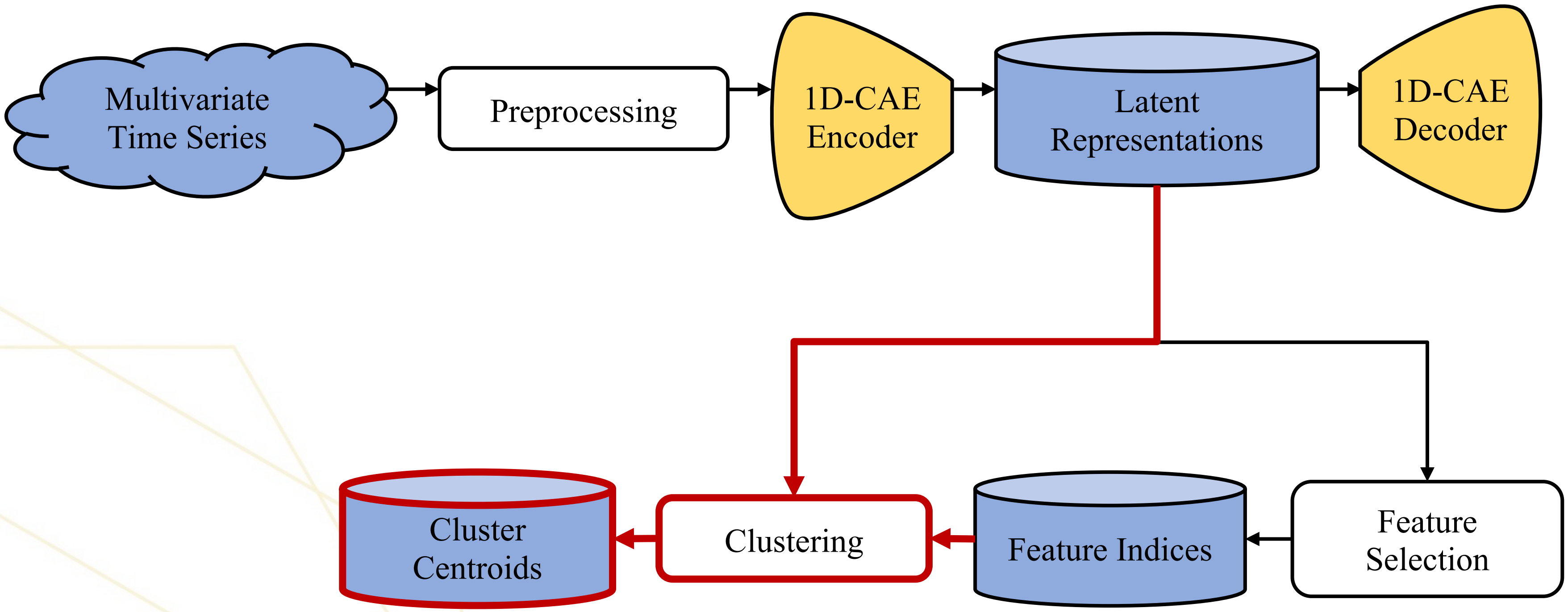
Offline Clustering (3/4)



Feature Selection



Offline Clustering (4/4)

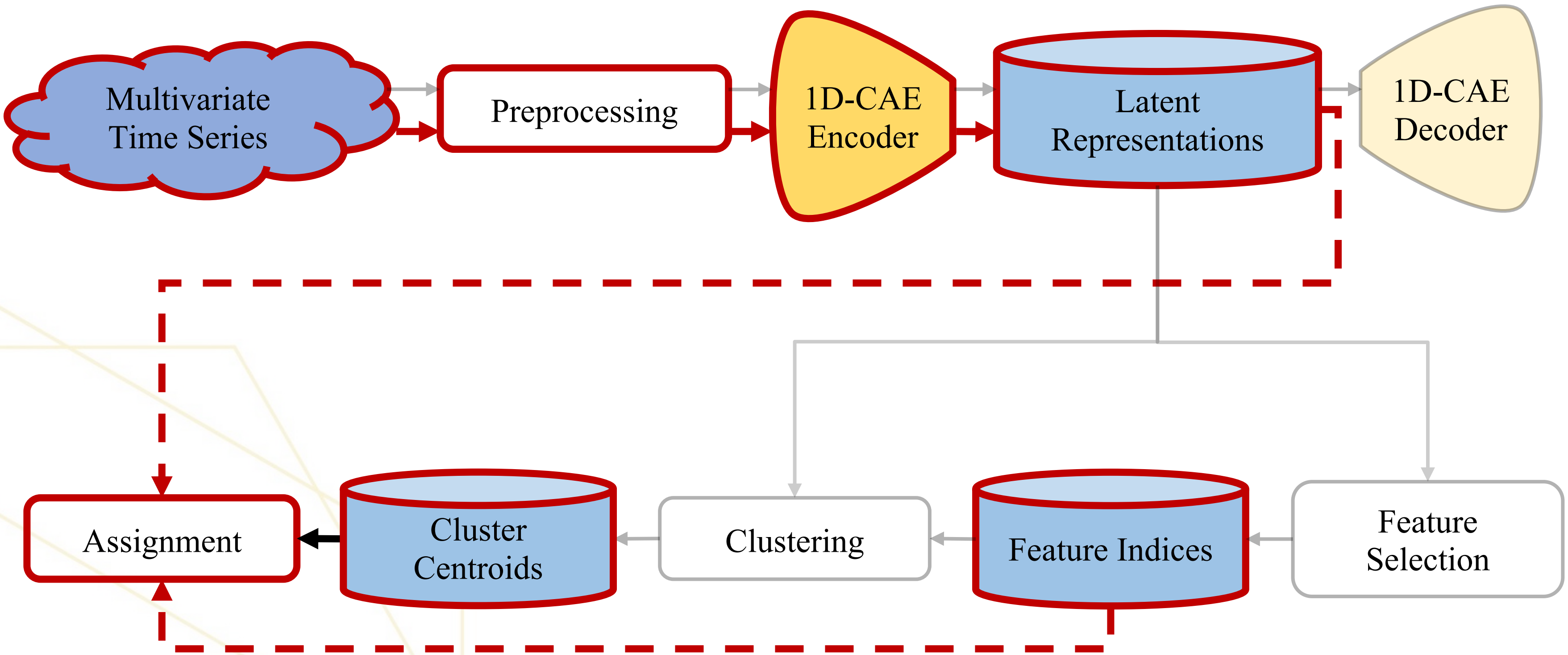


HAC Clustering

- Hierarchical agglomerative clustering (HAC) can use the **distance threshold** as a hyperparameter.
- **Average linkage** makes the distance measurement transitive.
- Determine whether a system instance is an **outlier**.

HAC with average linkage is adopted.

Online Classification

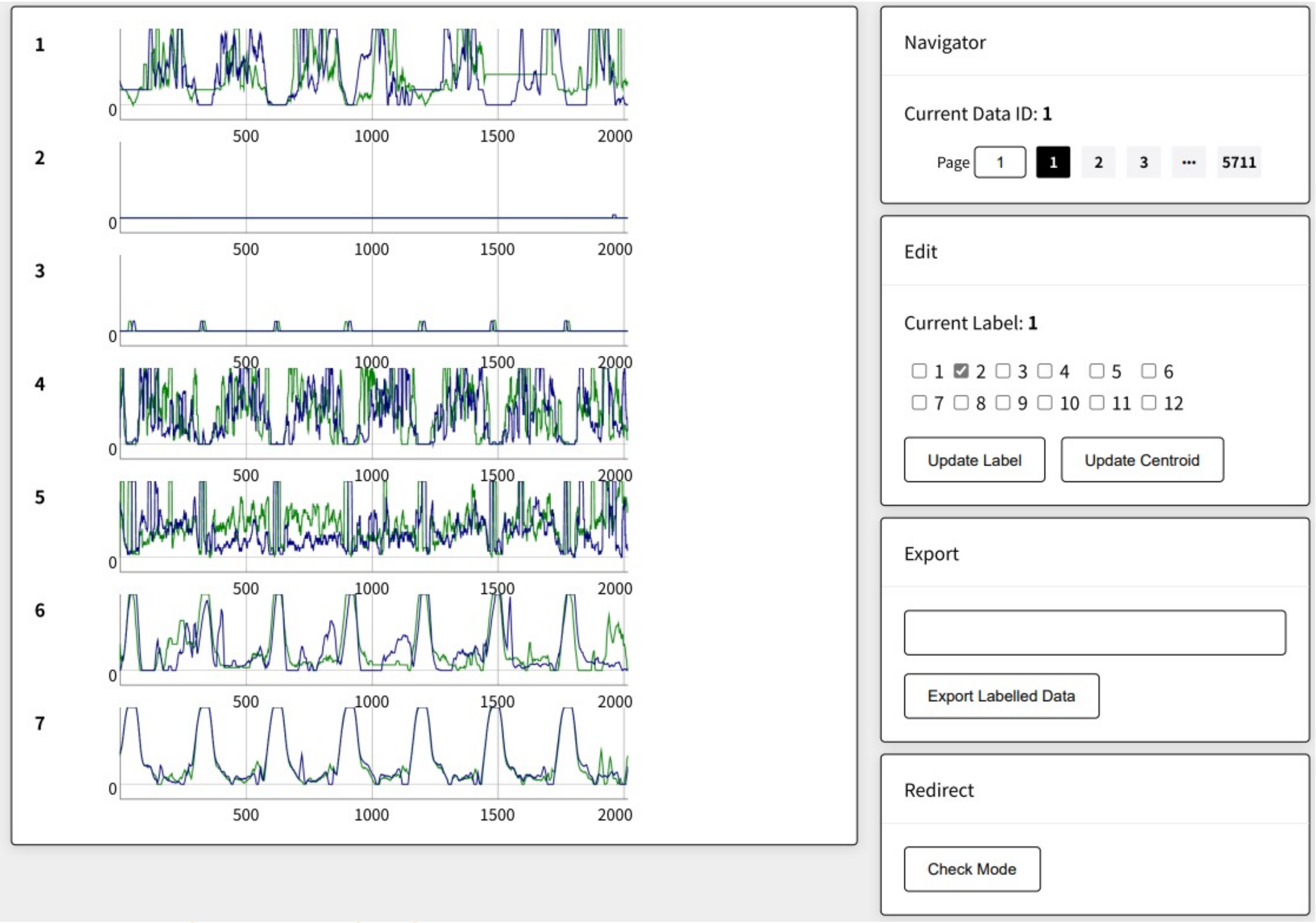


Assignment

- Calculate the centroid of each cluster and assign the newly coming data based on centroids.

$$c = \arg\min_{a \in C} \sum_{b \in C} D_{MTS}(a, b)$$

Labeling Tool



Evaluation

Dataset and Metrics

- **Dataset**
  - System-related dataset collected from ByteDance
  - 3175 system instances
  - 19 metrics
  - 7-day-long
- **Metrics**
  - Normalized Mutual Information (NMI)
  - Accuracy (ACC)
  - F_1-score

Overall Performance

Method	NMI	ACC	F_1	# Clusters	Avg. Time
OmniCluster	**0.9160**	**0.7990**	**0.9057**	19	**11.69 min**
TICC (IJCAI '18)	0.4826	0.3798	-	40	104.17 h
Mc2PCA (Neurocomputing '19)	0.2703	0.2306	-	10	22.03 min
FCFW (KBS '20)	0.6236	0.4117	-	10	195.86 h
SPCA+AED (ISA Transaction '17)	0.4084	0.2746	-	40	4.91 h

Effective and efficient

Ablation Study

- **C1: The 1D-CAE is replaced by 2D-CAE**
- **C2: w/o 1D-CAE**
- C3: w/o non-periodic feature removal
- C4: w/o redundancy elimination

Method	NMI	ACC	F_1	# Clusters	Avg. Time
OmniCluster	**0.9160**	0.7990	0.9057	19	11.69 min
C1	0.8511	0.6406	0.5243	46	**6.65 min**
C2	0.9102	**0.8009**	0.9057	23	135.60 min
C3	0.7602	0.4387	0.3022	117	9.09 min
C4	0.8742	0.6548	**0.9455**	35	11.48 min

1D-CAE improves efficiency.

Ablation Study

- C1: The 1D-CAE is replaced by 2D-CAE
- C2: w/o 1D-CAE
- **C3: w/o non-periodic feature removal**
- C4: w/o redundancy elimination

Method	NMI	ACC	F_1	# Clusters	Avg. Time
OmniCluster	**0.9160**	0.7990	0.9057	19	11.69 min
C1	0.8511	0.6406	0.5243	46	**6.65 min**
C2	0.9102	**0.8009**	0.9057	23	135.60 min
C3	0.7602	0.4387	0.3022	117	9.09 min
C4	0.8742	0.6548	**0.9455**	35	11.48 min

Non-periodic feature removal improves effectiveness.

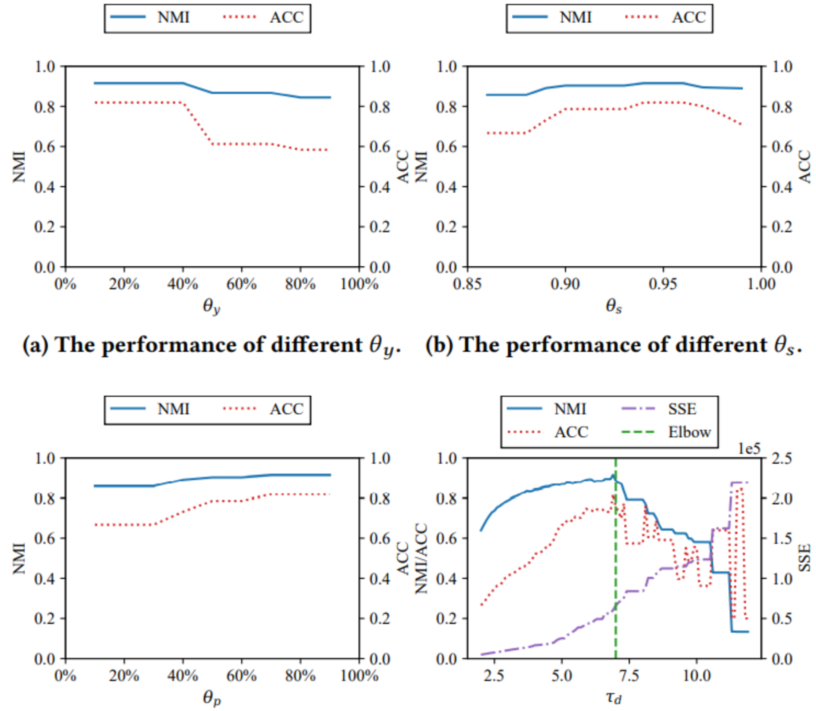
Ablation Study

- C1: The 1D-CAE is replaced by 2D-CAE
- C2: w/o 1D-CAE
- C3: w/o non-periodic feature removal
- **C4: w/o redundancy elimination**

Method	NMI	ACC	F_1	# Clusters	Avg. Time
OmniCluster	**0.9160**	0.7990	0.9057	19	11.69 min
C1	0.8511	0.6406	0.5243	46	**6.65 min**
C2	0.9102	**0.8009**	0.9057	23	135.60 min
C3	0.7602	0.4387	0.3022	117	9.09 min
C4	0.8742	0.6548	**0.9455**	35	11.48 min

Redundancy elimination improves effectiveness.

Effect of Hyperparameters



(a) The performance of different θ_y. (b) The performance of different θ_s.

(c) The performance of different θ_p. (d) Use SSE to select the optimal τ_d.

Robust to hyperparameters

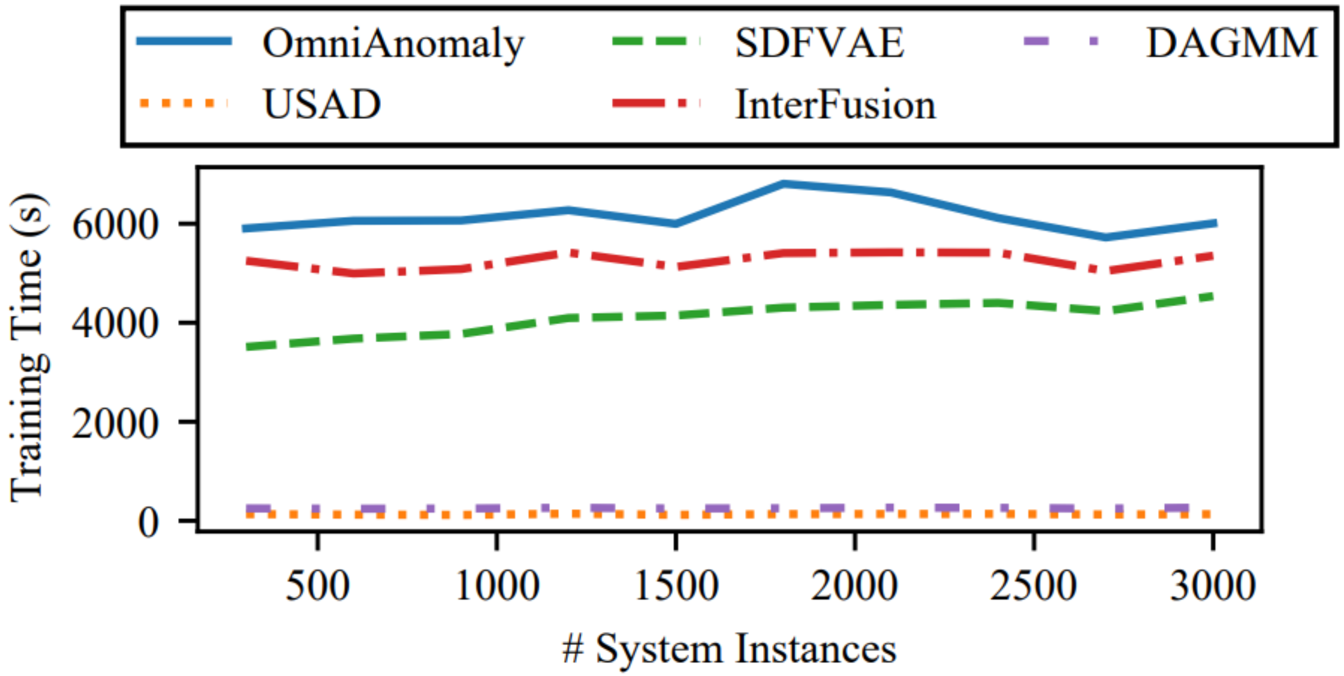
OmniCluster Helps

- E1: Sharing no model
- E2: Sharing one model
- E3: OmniCluster

Achieves satisfactory performance within an acceptable training time

Method	E1		E2		E3	
	F_1	Time (s)	F_1	Time (s)	F_1	Time (s)
USAD (KDD '20)	**0.926**	2726.80	0.841	8.99	0.923	133.88
OmniAnomaly (KDD '19)	0.842	56773.77	0.833	219.36	**0.845**	2748.88
SDFVAE (WWW '21)	**0.893**	76740.62	0.831	242.86	0.886	3511.04
InterFusion (KDD '21)	**0.836**	153378.84	0.680	295.35	0.827	9370.28
DAGMM (ICLR '18)	0.872	5628.57	0.826	18.50	**0.873**	254.89

OmniCluster Helps



Robust to the number of system instances

Conclusion

Conclusion

- **OmniCluster** is an **efficient** and **robust** algorithm for clustering high-dimensional MTS with noise, anomalies, and redundant features.
- **1D-CAE** improves efficiency and removes noise and anomalies.
- A **three-step feature selection strategy** prevents redundant and non-periodic features from degrading OmniCluster's performance.
- We have published a **labeling tool for MTS clustering** and a **labeled dataset** for further studies.

Thank You

Improvements

- Compared to existing methods, OmniCluster improves on different dimensions like volume, length of MTS, and noise.
- Volume & length of MTS **[Efficiency]**: 1D-CAE embeds high-dimensional data into low-dimensional features. Therefore, it can be applied to cluster a vast number of MTS.
- Noise **[Effectiveness]**: Moving average and 1D-CAE can filter out most noise. The impact of noise is thus reduced.

Dimensionality Explosion

- **1D-CAE** to reduce the number of time points in each metric.
- **Feature selection** techniques to reduce the number of metrics in each MTS.

System Instance

- Physical machines
- Virtual machine instances
- Dockers
- Containers
- …

Application Scenarios

- Large-scale Web services housing many system instances.