Tsinghua University

Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network

Ya Su, Youjian Zhao, Chenhao Niu,

Rong Liu, Wei Sun, Dan Pei

SIGKDD 2019

Outline

Background

Algorithm

Evaluation

Conclusion

Outline

Background

Algorithm

Evaluation

Conclusion

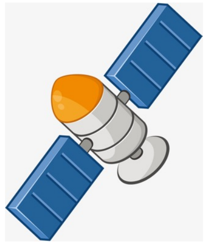
Anomaly Detection

- Graph [SIGKDD 2018, AI Magazine 2014]
- Log Messages [SIGKDD 2016, SIGKDD 2017]
- Time Series [SIGKDD 2015, SIGKDD 2017, SIGKDD 2018]
  - Univariate Time Series
  - Mutivariate Time Series

Entities with monitored multivariate time series

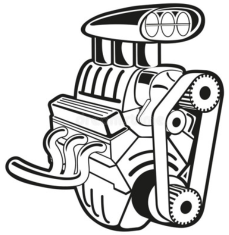
Entities

Server Machine



SpaceCraft

Robot-assisted System



Engine

. . .

Multi-metrics

Server Machine	SpaceCraft	Robot-assisted System	Engine
CPU Load Network Usage Memory Usage …	Radiation Temperature Power …	Kinematic Visual Haptic …	Accelerator Torque Temperature …

. . .

Entities with monitored multivariate time series

Entities

Server Machine

More intuitive

More effective

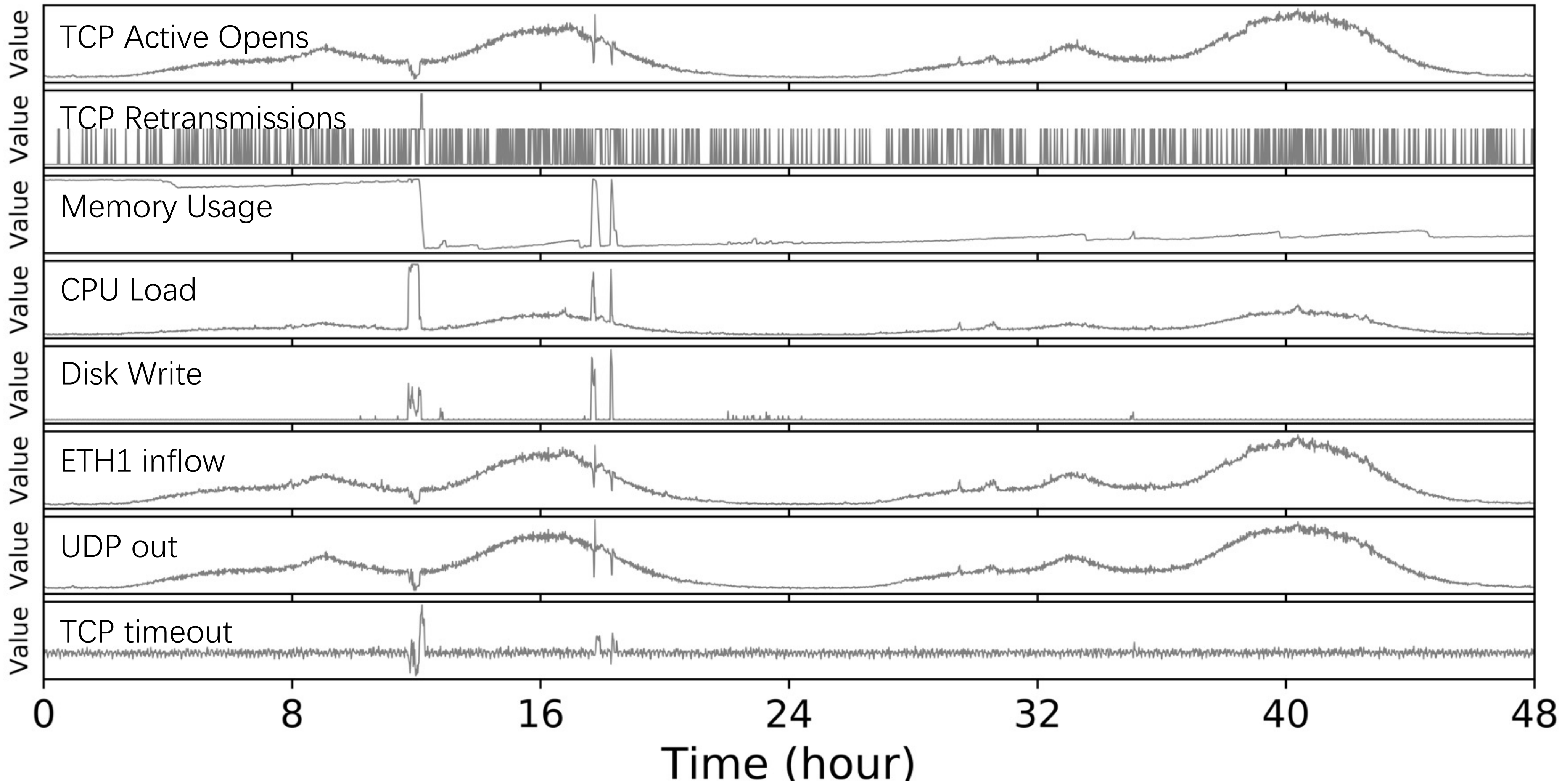
More efficient

Multi-metrics

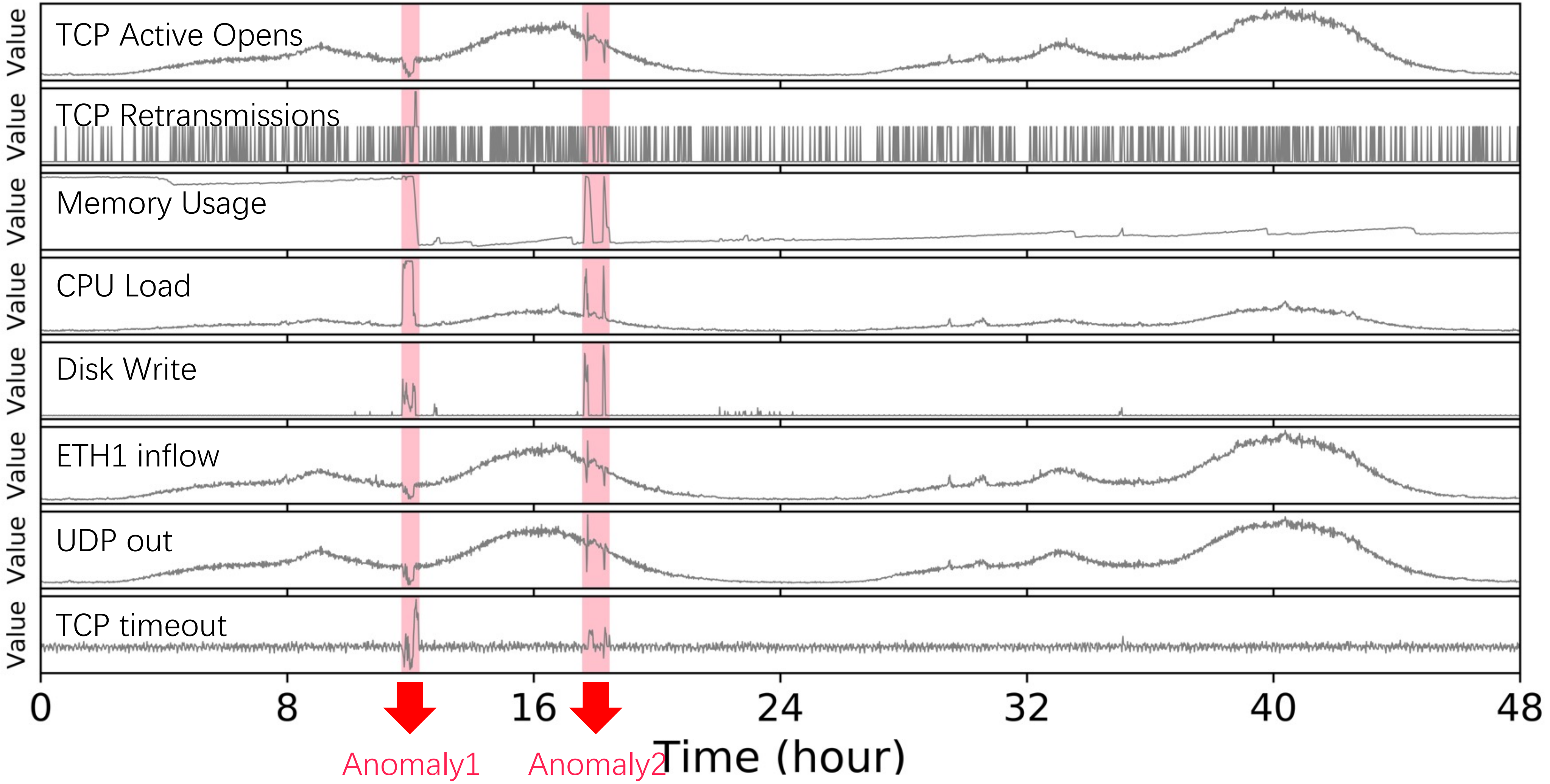
CPU Load
Network Usage
Memory Usage
...

Multivariate time series

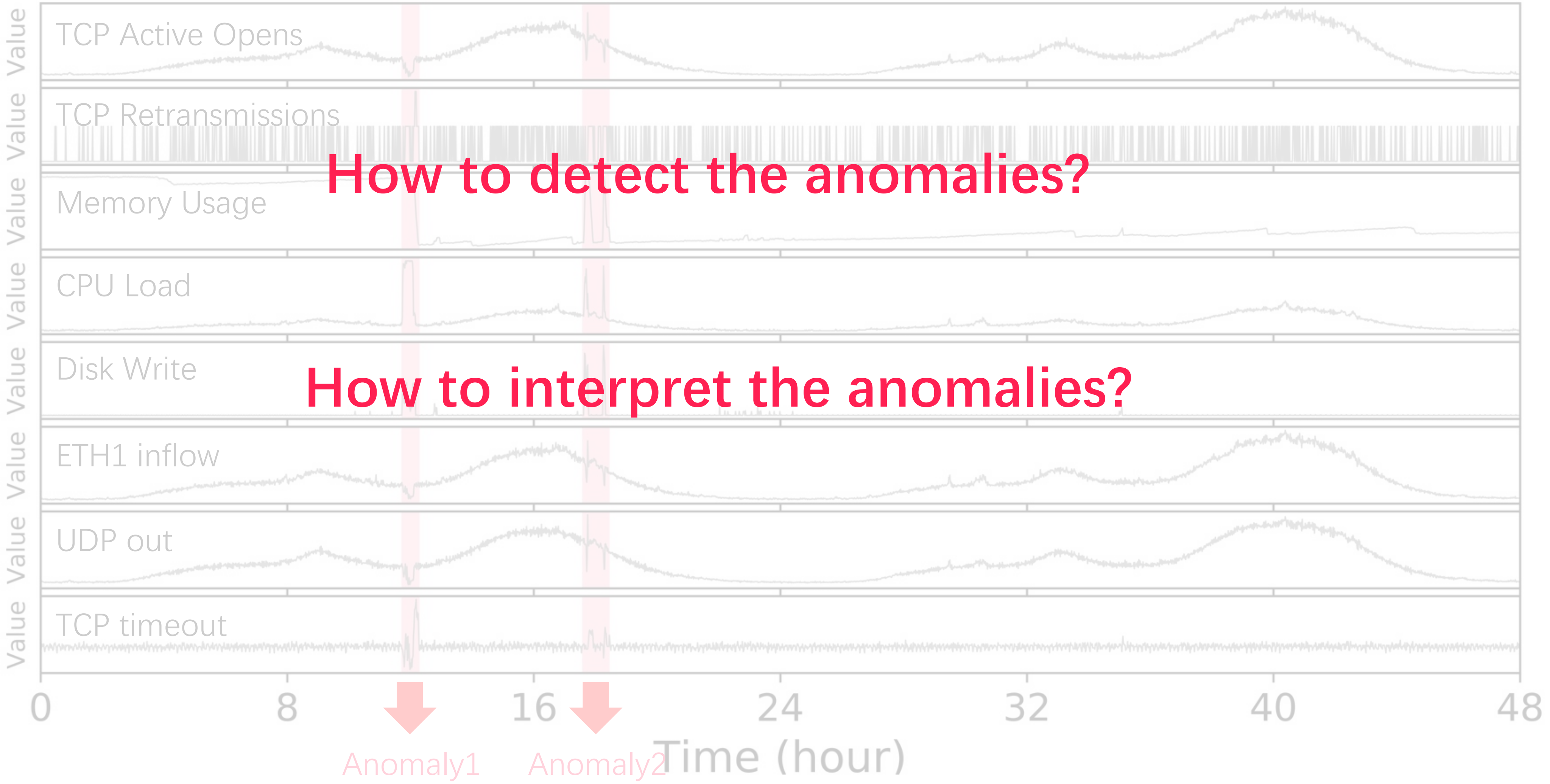
Machine with monitored multivariate time series



Machine with monitored multivariate time series



Motivations



How to detect the anomalies?

How to interpret the anomalies?

Challenges

- How to deal with the temporal dependence of multivariate time series ?

- How to deal with the stochasticity of multivariate time series ?

- How to provide interpretation to the detected entity-level anomalies ?

Related work

Deterministic models	Stochastic based models
LSTM、 LSTM-based Encoder-Decoder [SIGKDD2018，ICML workshop 2016，NIPS 2016]	DAGMM、LSTM-VAE [IEEE Robotics and Automation Letters 2018, ICLR 2018]
Deterministic models without stochastic variables	Ignore the dependence of time series or stochastic variables.

Outline

Background

Algorithm

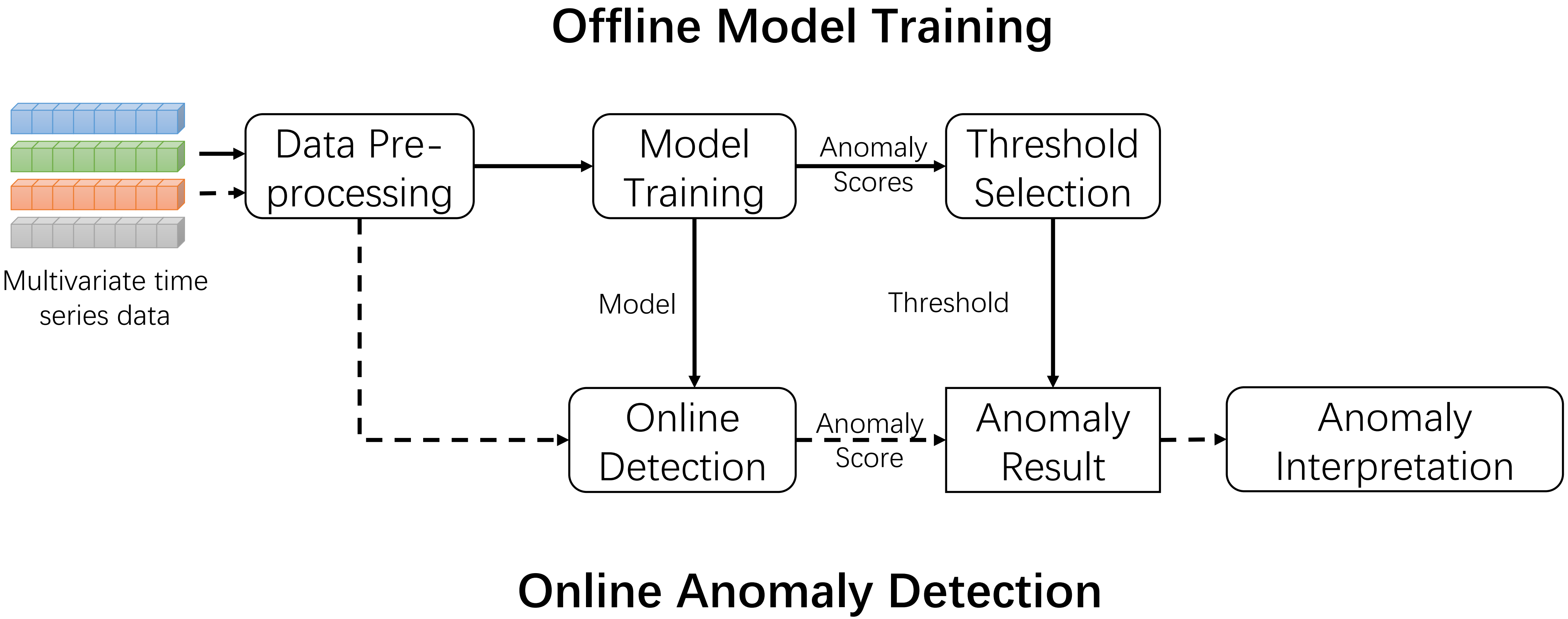
Evaluation

Conclusion

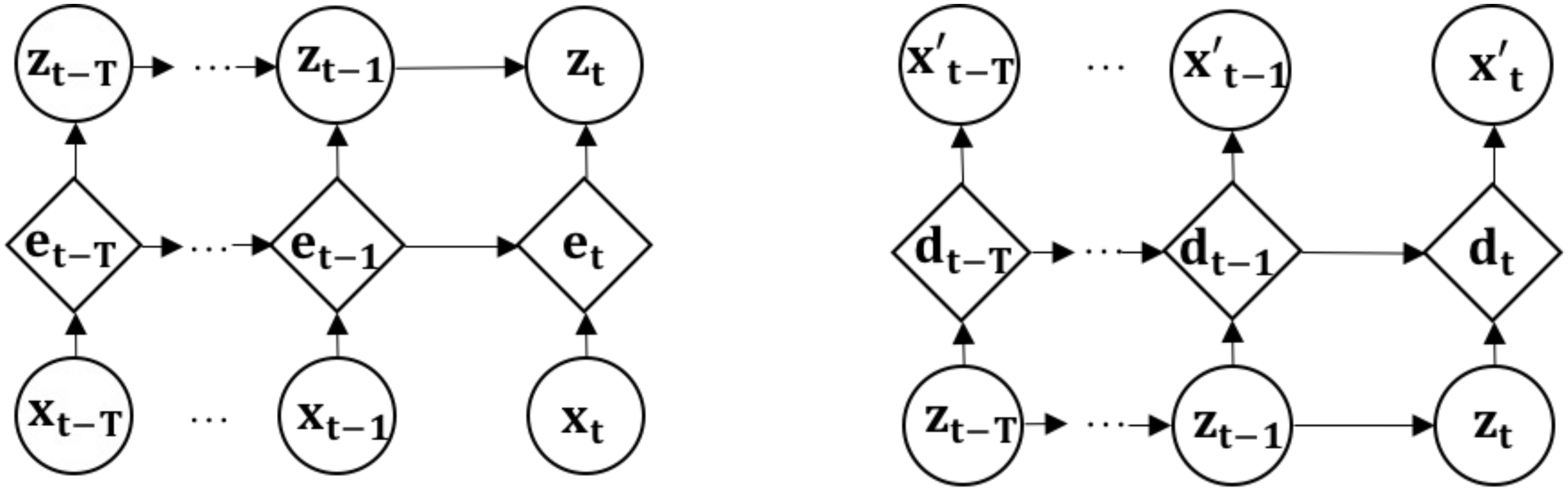
OmniAnomaly

Helps answer the questions

Structure of OmniAnomaly

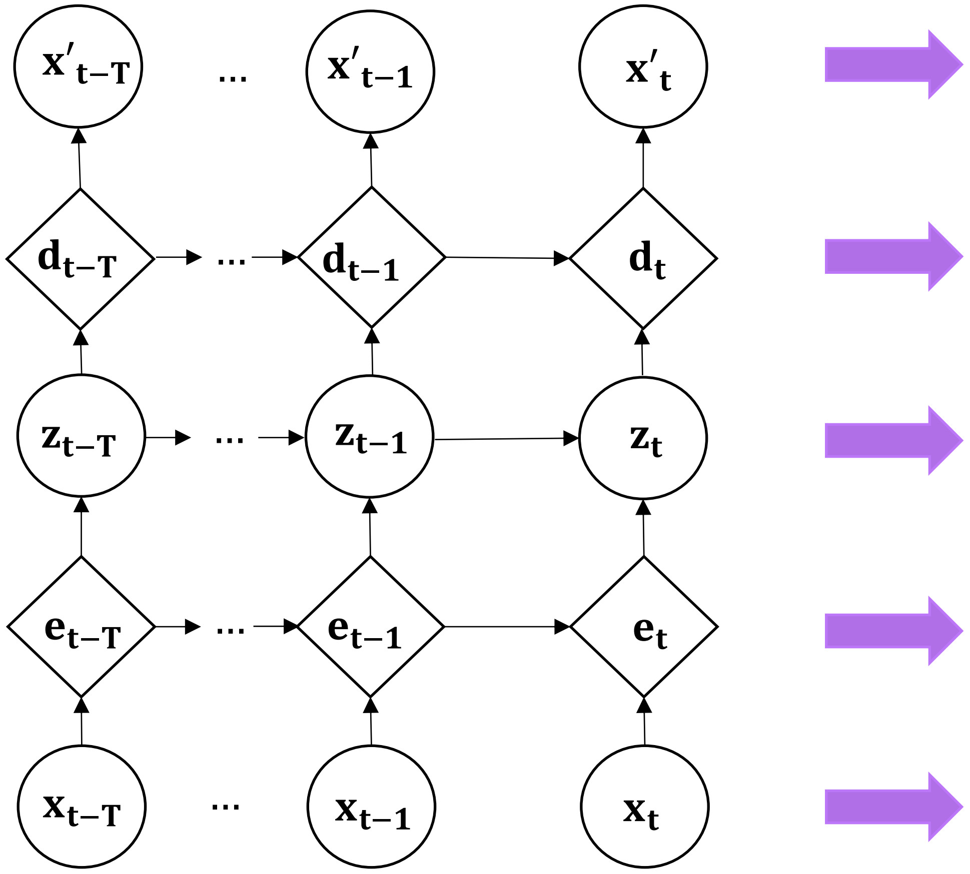


Model Architecture of OmniAnomaly



(a1) qnet (a2) pnet

Model Architecture of OmniAnomaly



Reconstructed data

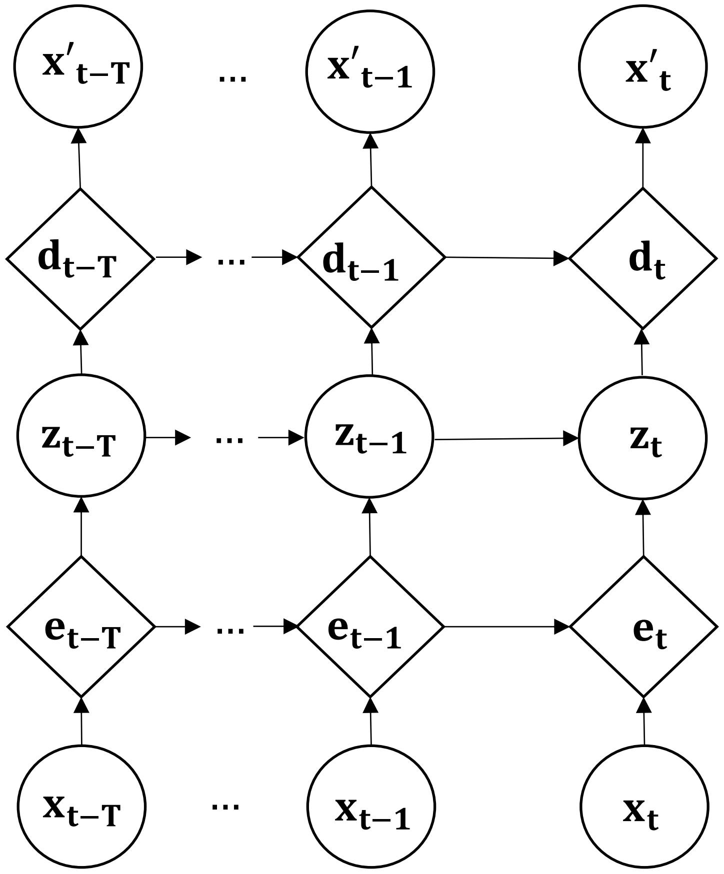
GRU cells for capturing temporal dependence

Stochastic cells for modeling data distribution

GRU cells for capturing temporal dependence

Input Sequence data

Core idea of OmniAnomaly



A good z_t can represent x_t well no matter x_t is anomalous or not.

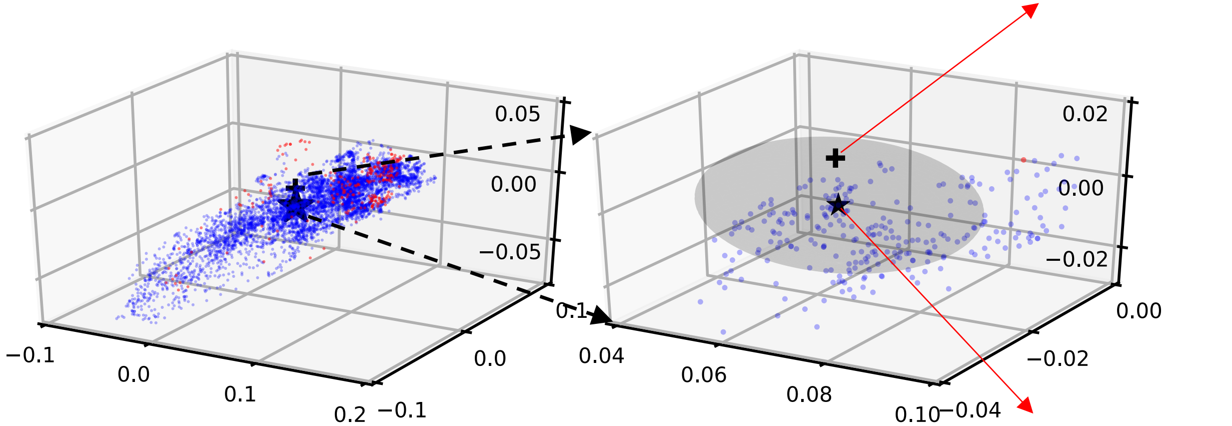
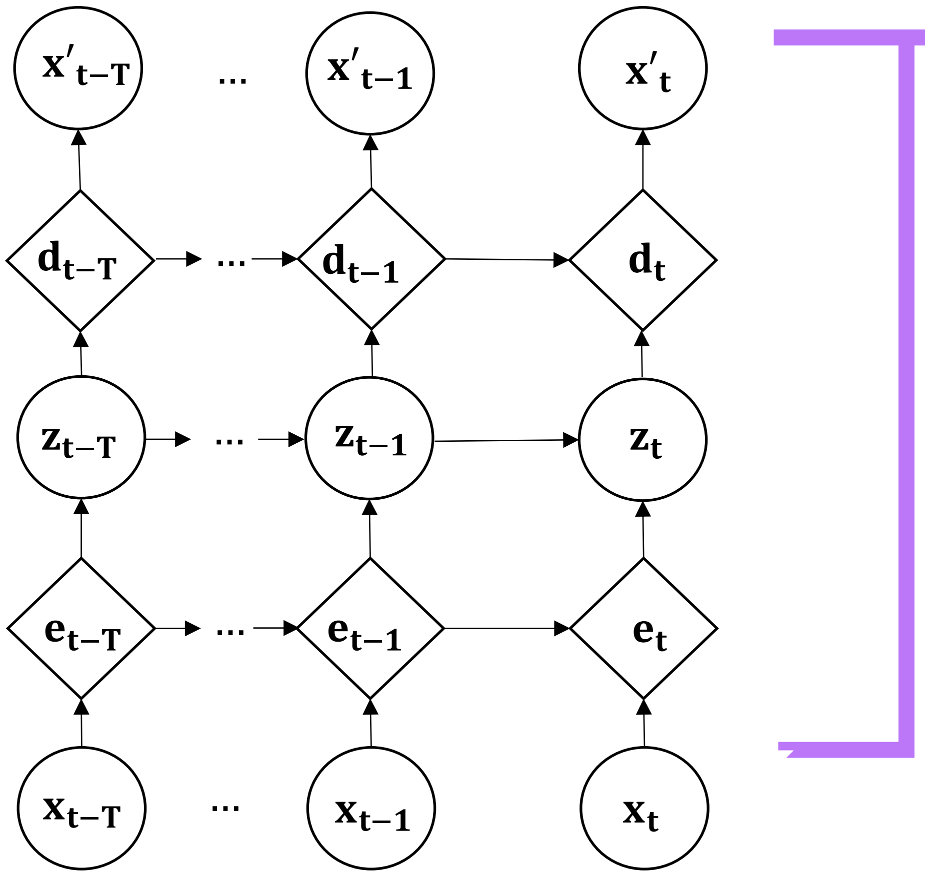


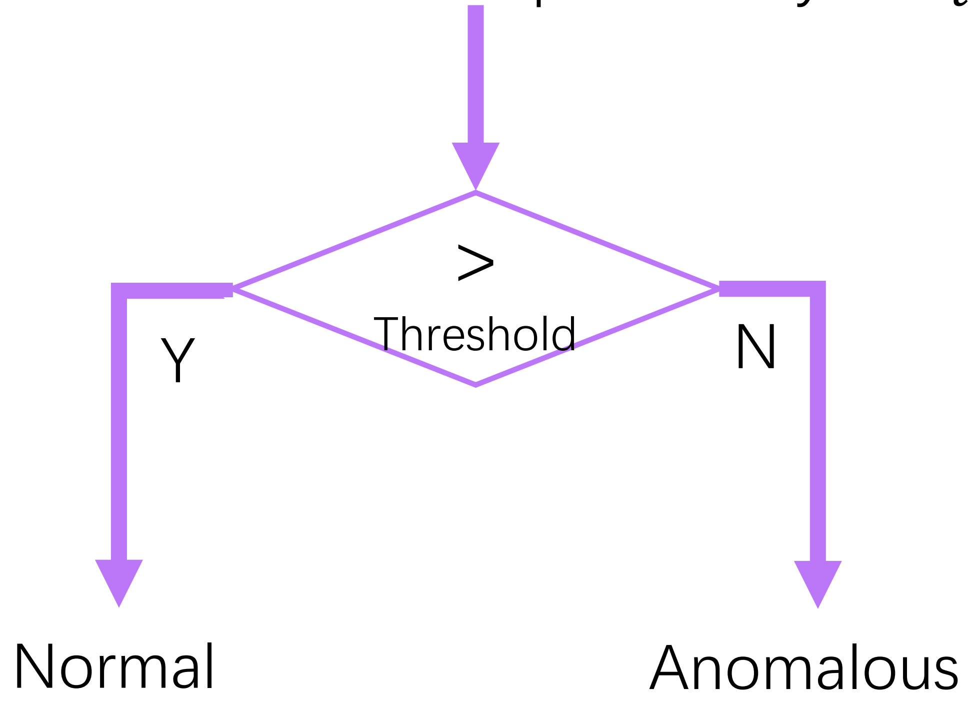
Fig: 3-dimensional z_t of x_t

When x_t is anomalous, its z_t can still represent its normal pattern and x'_t will be normal too.

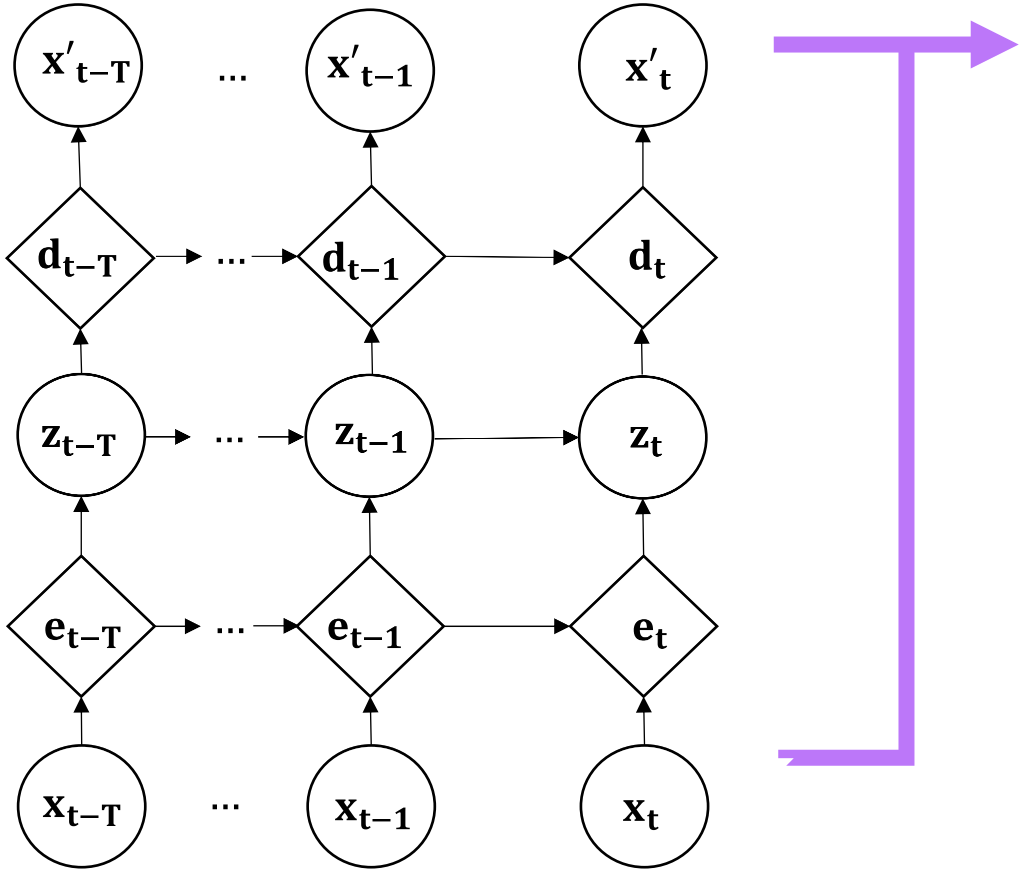
Anomaly detection of OmniAnomaly



Anomaly Score S_t = Reconstruction probability of x_t



Anomaly detection of OmniAnomaly



Anomaly Score S_t =
Reconstruction probability of x_t

$x_t = [x_t^1, x_t^2, \ldots, x_t^M]$, M is the dimension

$$S_t = \sum_{i=1}^{M} S_t^i$$

Sort the $[S_t^1, S_t^2, \ldots, S_t^M]$ in ascending order, and the Top K dimensions can interpret the anomaly.

Outline

Background

Algorithm

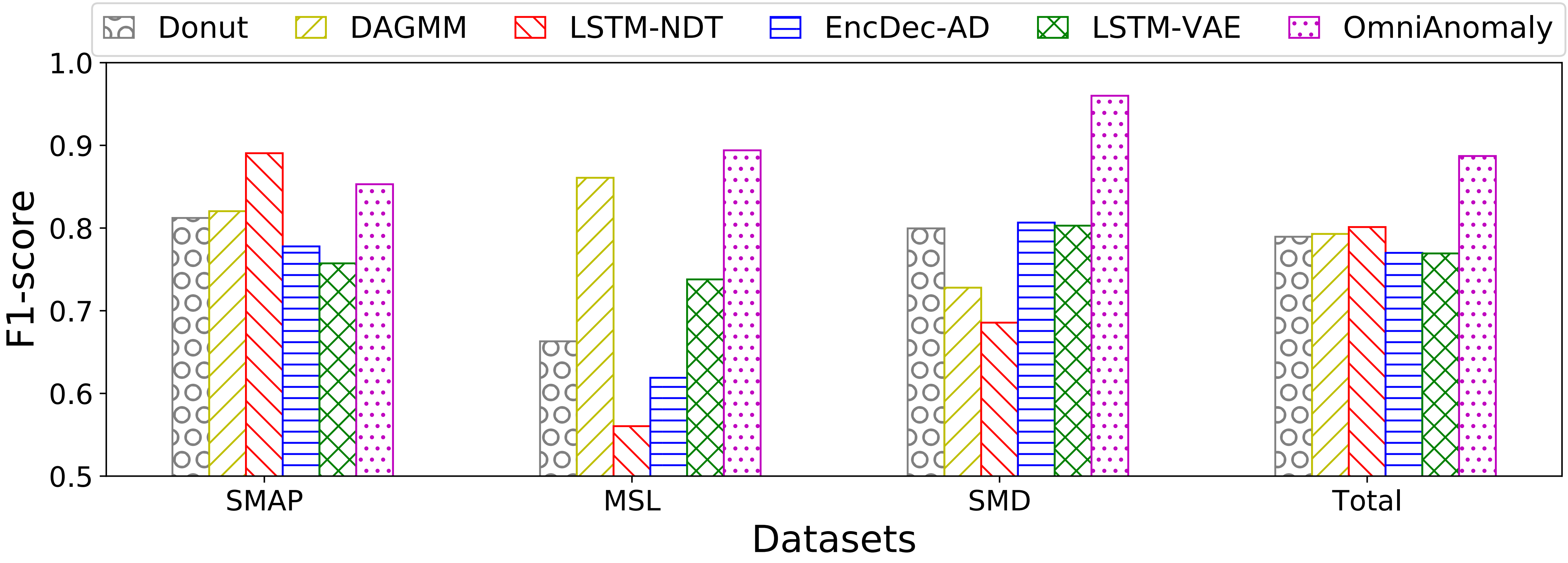
Evaluation

Conclusion

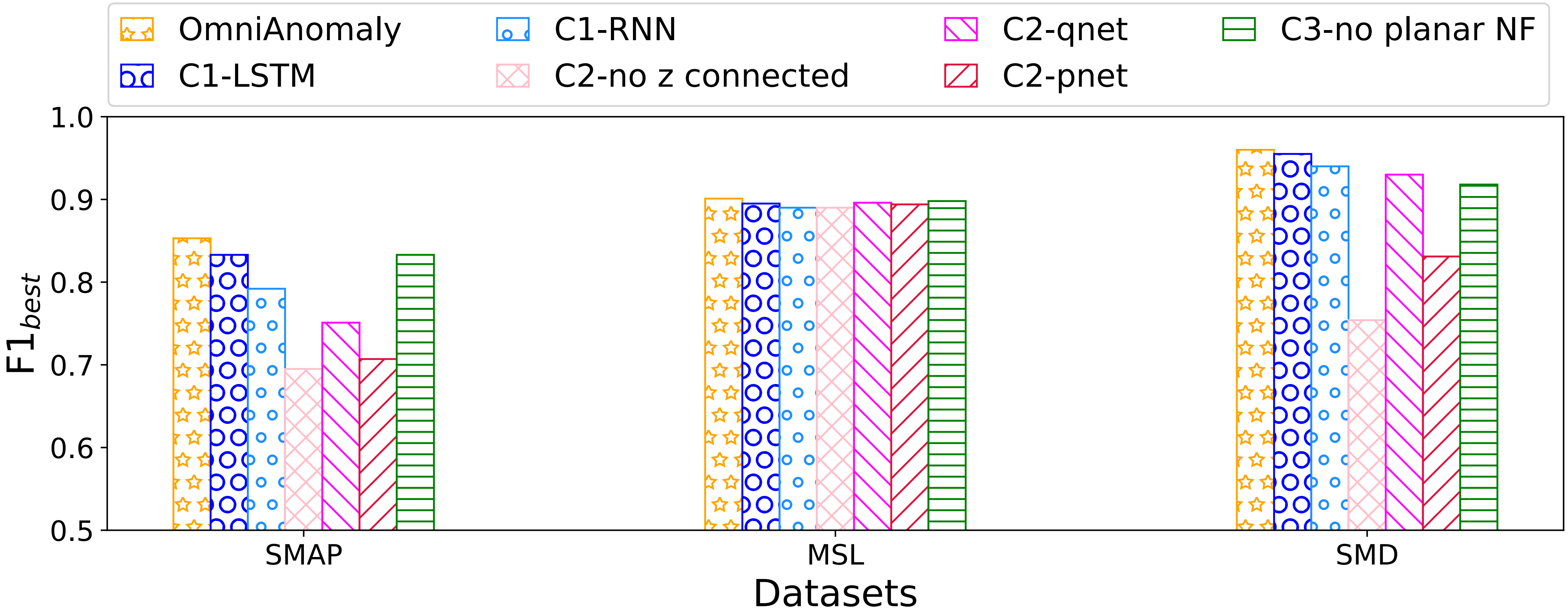
Datasets

DataSet name	Number of entities	Number of dimensions	Training set size	Testing set size	Anomaly ratio(%)
SMAP	55	25	135183	427617	13.13
MSL	27	55	58317	73729	10.72
SMD	28	38	708405	708420	4.16

F1-best of OmniAnomaly and baselines



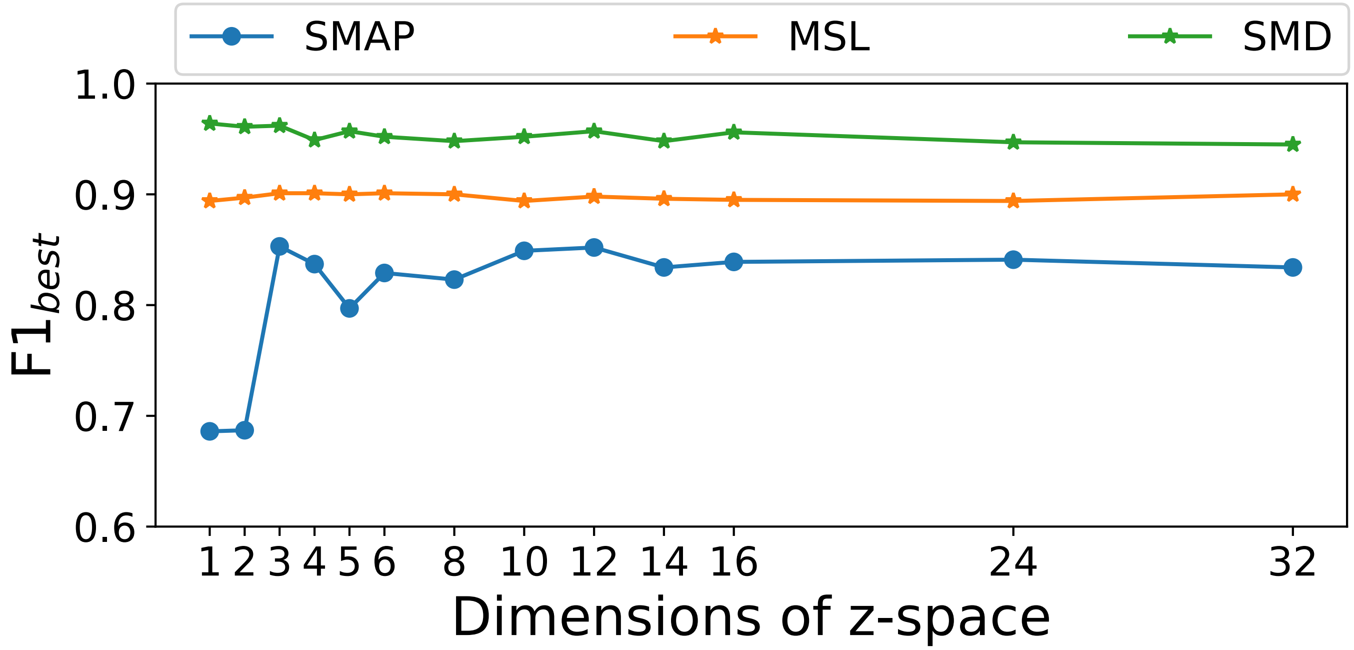
F1-best of OmniAnomaly and variants



F1 obtained through POT vs. F1-best

Evaluation metrics for OmniAnomaly	SMAP	MSL	SMD
F1 obtained through POT	0.8434	0.8989	0.8857
F1-best	0.8535	0.9014	0.9620

F1-best of OmniAnomaly with different z dimension



Outline

- Background
- Algorithm
- Evaluation
- Conclusion

OmniAnomaly

- The first multivariate time series anomaly detection method that deal with explicit temporal dependence among stochastic variables
- The first anomaly interpretation approach for stochastic based multivariate time series anomaly detection algorithms
- Achieve an overall F1-score of 0.86 in three real world datasets.
- The interpretation accuracy is up to 0.89.

Lessons for time series data learning

- A combination of stochastic deep Bayesian model and deterministic RNN model is necessary
- The connection of stochastic variables is necessary and effective
- It is necessary to assume non-Gaussian distributions in z-space

Lessons for for multivariate time series anomaly detection

- Reconstruction-based models are more robust than prediction-based models

- It is critical to obtain robust latent representations which can accurately capture the normal patterns of time series

- Reconstruction-based stochastic approaches offer an opportunity to interpret the anomalies

Thanks

su-y16@mails.tsinghua.edu.cn

清華大學
Tsinghua University

CTF: Anomaly Detection in High-Dimensional Time Series with Coarse-to-Fine Model Transfer

Ming Sun, Ya Su, Shenglin Zhang, Yuanpu Cao, Yuqing Liu, Dan Pei,
Wenfei Wu, Yongsu Zhang, Xiaozhou Liu, Junliang Tang

INFOCOM 2021

清華大學 Tsinghua University

南開大學 Nankai University

ByteDance 字节跳动

Outline

Background

Design

Evaluation

Conclusion

Outline

Background

Design

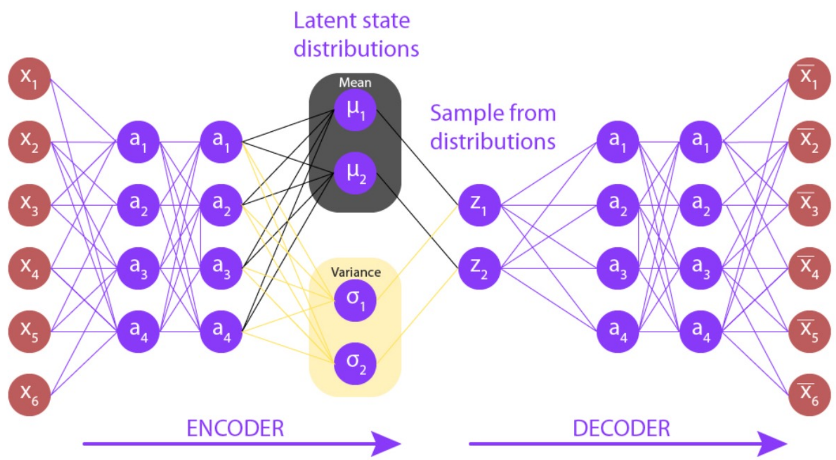
Evaluation

Conclusion

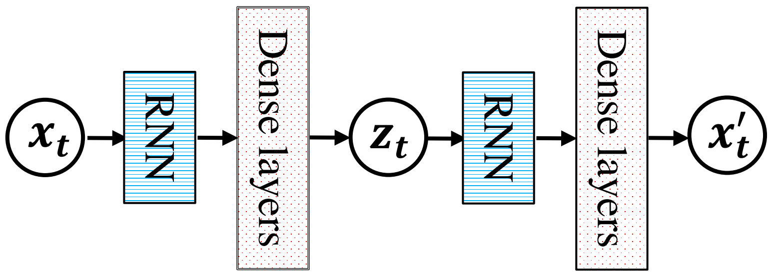
DL Algorithms in the Infra Operation

- Advantages
  - automation
  - robustness
  - Saving operator’s labor
- Example:
  - RNN-VAE for anomaly detection

RNN-VAE Based Algorithms



Variational Auto-Encoder (VAE)



Network architecture of RNN-VAE models at time t

$\boldsymbol{x}_t$ (49) -> $\boldsymbol{z}_t$ (3) -> $\boldsymbol{x}'_t$ (49)

KPI dimension reduced

Network Layers

- RNN: Shallow & general
- Dense layers: Deep & specific

Scalability is the problem for large scale

- High-Dimensional Data
  - Machines: in millions
  - KPI: in tens
  - Time: Frequent data query (2880 samples/day)

➢ One model per machine: time ✖
10X minutes * 1X million machines

➢ One model for all: accuracy ✖

Scalability is the problem for large scale

- High-Dimensional Data
  - Machines: in millions
  - KPI: in tens
  - Time: Frequent data query (2880 samples/day)

Goal: devise scalable deep learning (DL) algorithms for large-scale anomaly detection

Intuition and Challenges

- Intuition: Cluster Machines first, then run DL for each cluster

dependency

- Challenge 1: clustering model training
    - Clustering cannot run on high-dimensional data
    - DL cannot run on whole dataset without clustering
    - Solution: Synthetic framework

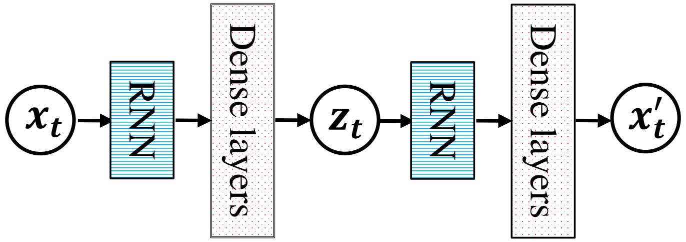
  Coarse-grained model -> clustering -> fine-grained models

Intuition and Challenges

- Intuition: Cluster Machines first, then run DL for each cluster
- Challenge 1: clustering $\xleftrightarrow{\text{dependency}}$ model training
  - Clustering cannot run on high-dimensional data
  - DL cannot run on whole dataset without clustering
  - Solution: Synthetic framework
- Challenge 2: High dimension of time domain
  - Hard to cluster even KPI is compressed
  - Solution: compress sequence to z-distribution

Intuition and Challenges

- Intuition: Cluster Machines first, then run DL for each cluster
- Challenge 1: clustering $\xleftrightarrow{\text{dependency}}$ model training
  - Clustering cannot run on high-dimensional data
  - DL cannot run on whole dataset without clustering
  - Solution: Synthetic framework
- Challenge 2: High dimension of time domain
  - Hard to cluster even KPI is compressed
  - Solution: compress sequence to z-distribution
- Challenge 3: Neural network training method
  - Solution: fine-tuning strategy
  - Freeze RNN and tune dense layers



Outline

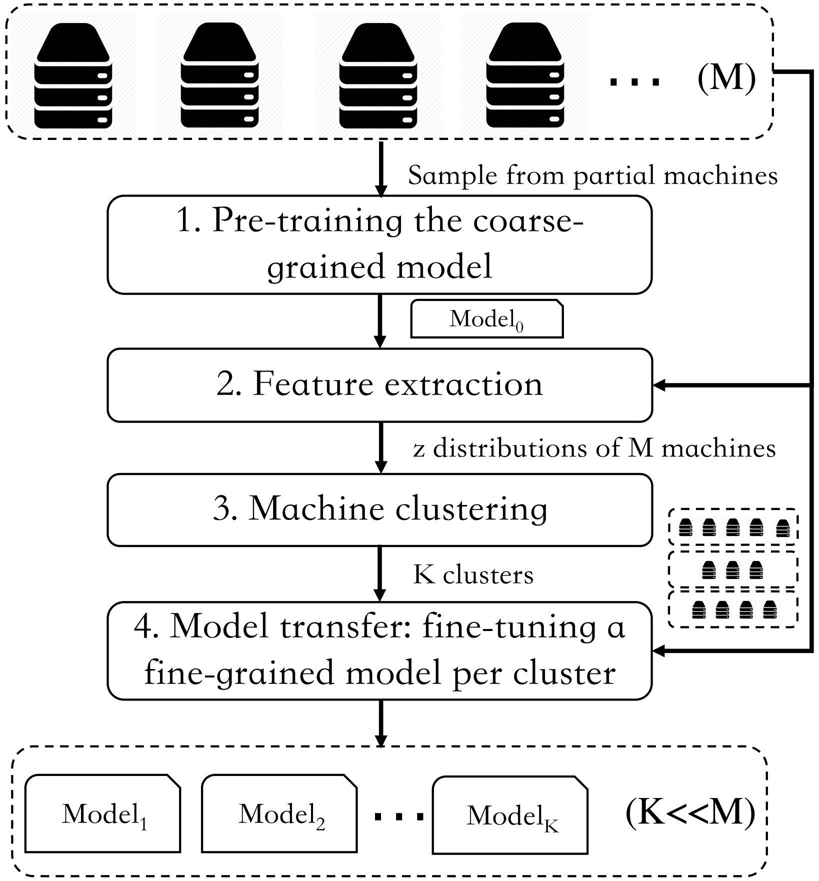
Background

Design

Evaluation

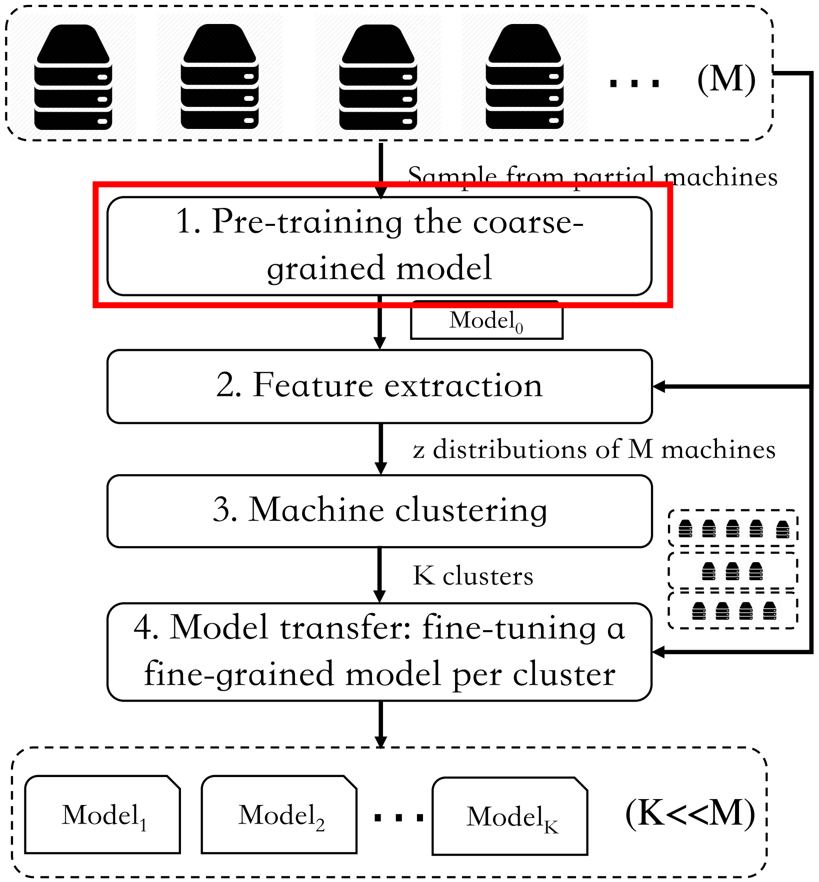
Conclusion

Framework of model training



Framework of model training

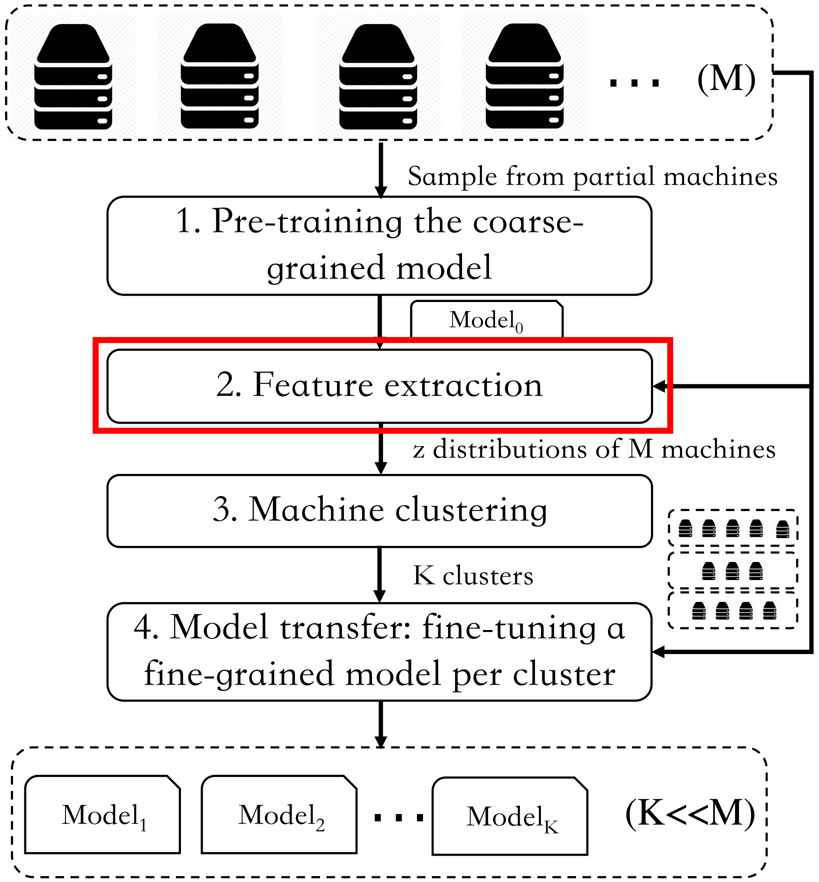
Framework of model training



Framework of model training

- Sampling strategy:
  - Machine sampling
  - Time sampling

Framework of model training



Framework of model training

$\boldsymbol{x_t}$ sequence

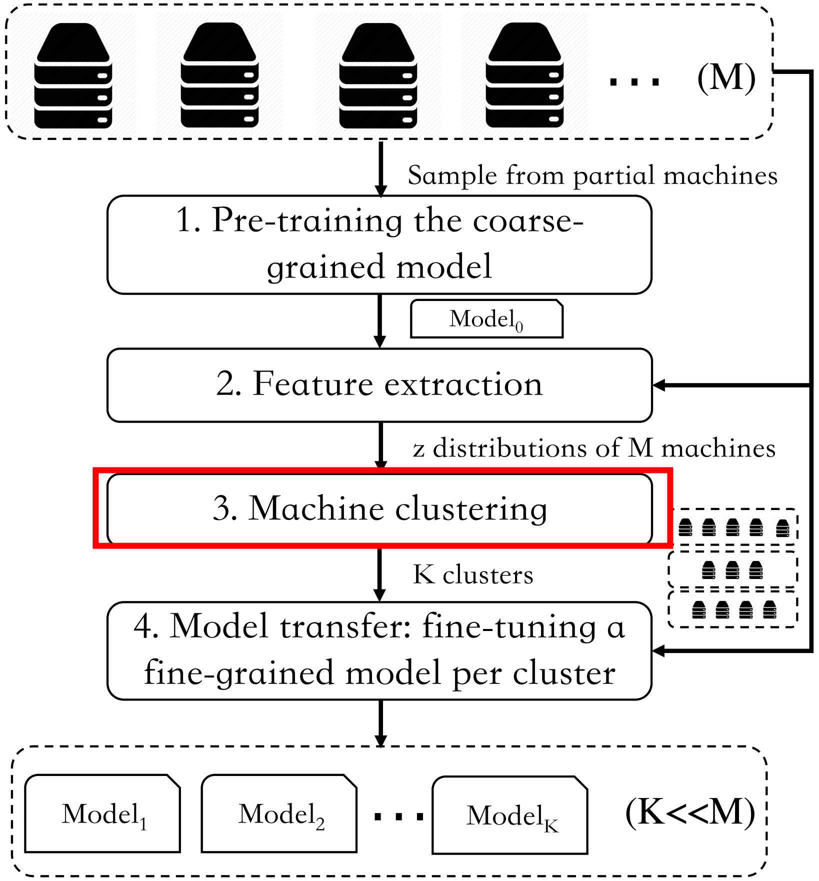
↓

$\boldsymbol{z_t}$ sequence

↓

$\boldsymbol{z_t}$ distribution

Framework of model training



Framework of model training

z_t distribution

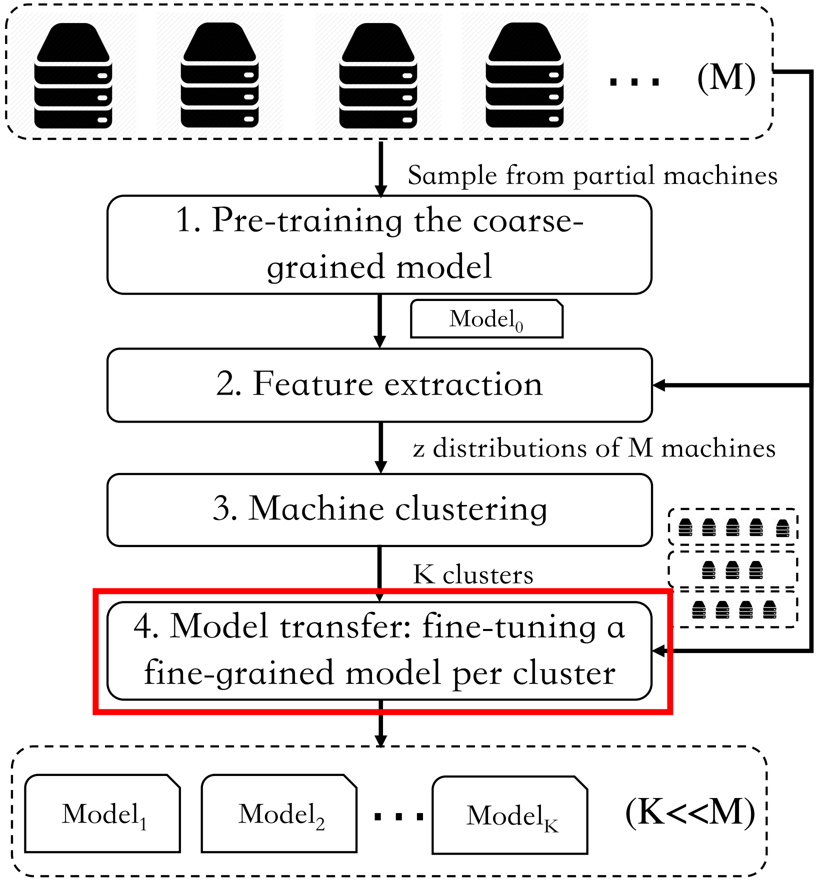
↓ Wasserstein distance

distance matrix

↓ HAC algorithm

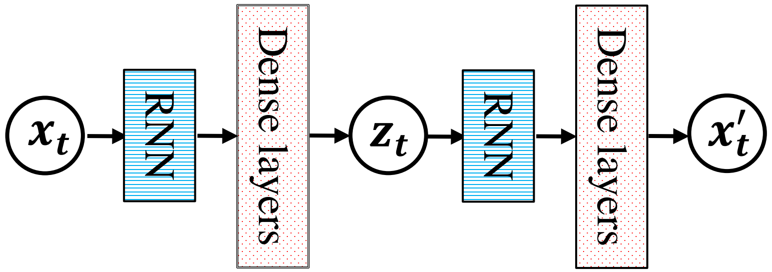
clustering results

Framework of model training

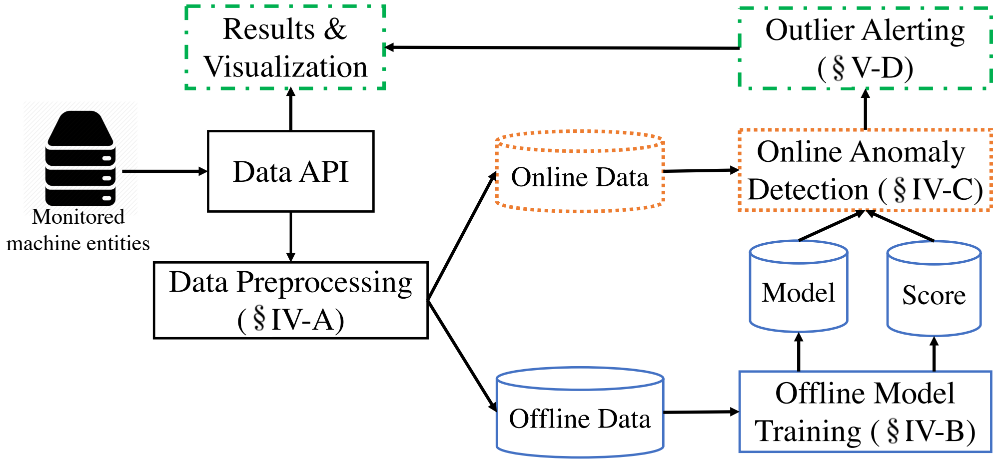


Framework of model training

- Fine-tuning strategy:
  - RNN: fixed
  - Dense layers: tuned



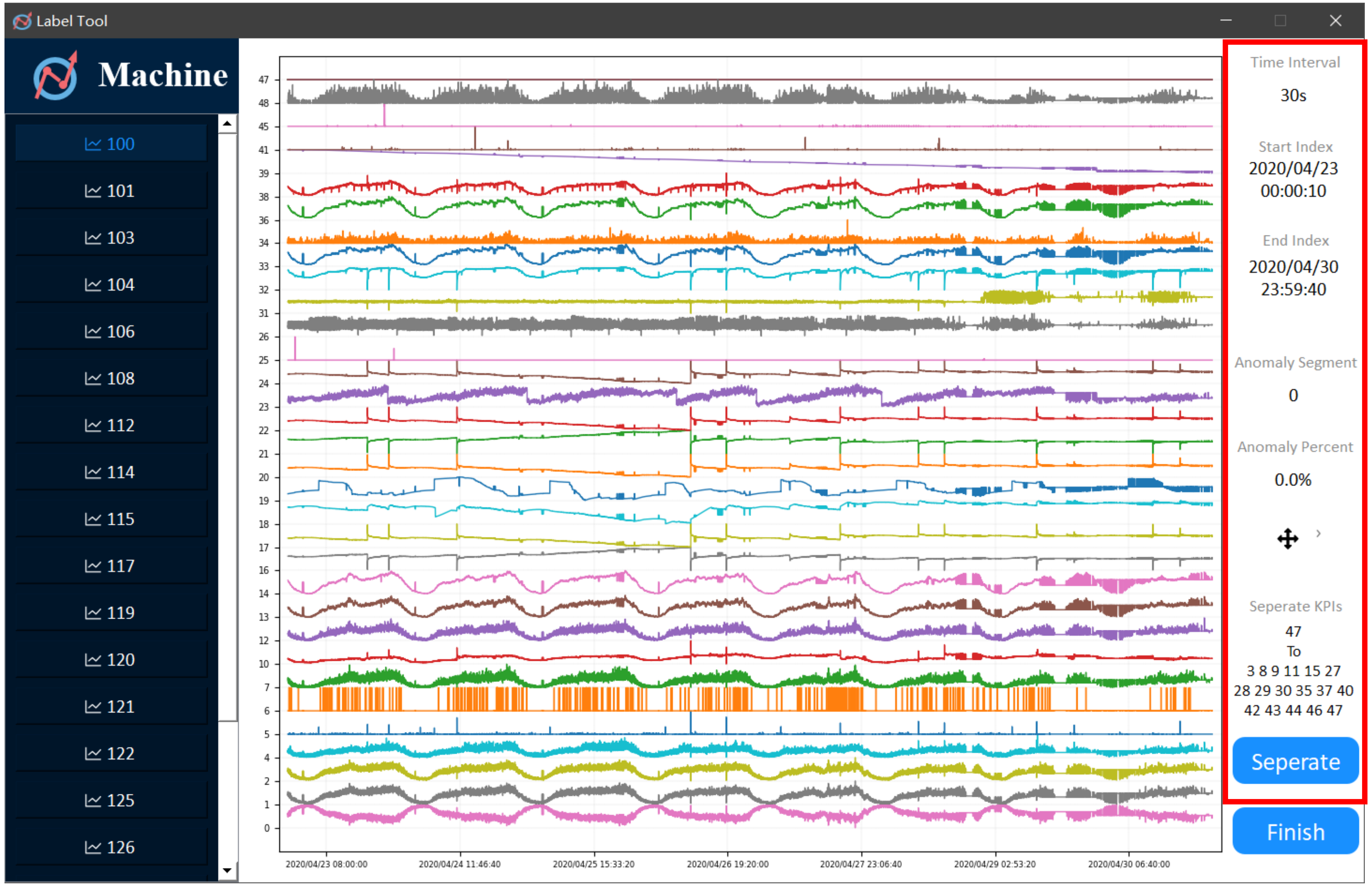
System architecture



System architecture

1. Data preprocessing
2. Offline model training
3. Online anomaly detection

Labeling tools



The interface of the labeling tool

Outline

Background

Design

Evaluation

Conclusion

Dataset & performance metrics

- **Dataset:**
  - # Machine entities: 533
  - Dimension of each machine entity: 49 KPIs x 37440 time points (frequency: 30s, 13 days)
  - Training = first 5 days, Testing = last 8 days
- **Metrics:**
  - F1, Precision, Recall: average of all machine entities.
  - Model training time

Overall performance

- **Scalability**
  - Pre-training: fixed (5493s)

M	533	10^3	10^4	10^5	10^5(6 servers)
Pre-training	5493	5493	5493	5493	5493
Feature extraction	166	311	3113	31130	5292
Clustering	3	6	232	576	576
Model transfer	2238	2238	4475	22375	4475
Total	7900	8048	13313	59574	15836
Average	14.822	8.048	1.331	0.596	0.158

The execution time of each step under different numbers of machine entities

Methods	F1	Precision	Recall
Without alerting	0.830	0.785	0.881
With alerting	0.892	0.907	0.877

F1, Precision, and Recall scores of CTF without and with alerting

Overall performance

- **Scalability**
  - Pre-training: fixed (5493s)
  - feature extraction: 0.3s / machine

M	533	10^3	10^4	10^5	10^5(6 servers)
Pre-training	5493	5493	5493	5493	5493
Feature extraction	166	311	3113	31130	5292
Clustering	3	6	232	576	576
Model transfer	2238	2238	4475	22375	4475
Total	7900	8048	13313	59574	15836
Average	14.822	8.048	1.331	0.596	0.158

The execution time of each step under different numbers of machine entities

Methods	F1	Precision	Recall
Without alerting	0.830	0.785	0.881
With alerting	0.892	0.907	0.877

F1, Precision, and Recall scores of CTF without and with alerting

Overall performance

- **Scalability**
  - Pre-training: fixed (5493s)
  - feature extraction: 0.3s / machine
  - Clustering: much smaller
  - Fine-tuning: 448s / model

M	533	10^3	10^4	10^5	10^5(6 servers)
Pre-training	5493	5493	5493	5493	5493
Feature extraction	166	311	3113	31130	5292
Clustering	3	6	232	576	576
Model transfer	2238	2238	4475	22375	4475
Total	7900	8048	13313	59574	15836
Average	14.822	8.048	1.331	0.596	0.158

The execution time of each step under different numbers of machine entities

Methods	F1	Precision	Recall
Without alerting	0.830	0.785	0.881
With alerting	0.892	0.907	0.877

F1, Precision, and Recall scores of CTF without and with alerting

Overall performance

- **Scalability**
  - Pre-training: fixed (5493s)
  - feature extraction: 0.3s machine
  - Clustering: much smaller
  - Fine-tuning: 448s / model
- **Effectiveness**
  - F1: 0.830->0.892

M	533	10^3	10^4	10^5	10^5(6 servers)
Pre-training	5493	5493	5493	5493	5493
Feature extraction	166	311	3113	31130	5292
Clustering	3	6	232	576	576
Model transfer	2238	2238	4475	22375	4475
Total	7900	8048	13313	59574	15836
Average	14.822	8.048	1.331	0.596	0.158

The execution time of each step under different numbers of machine entities

Methods	F1	Precision	Recall
Without alerting	0.830	0.785	0.881
With alerting	0.892	0.907	0.877

F1, Precision, and Recall scores of CTF without and with alerting

Overall performance

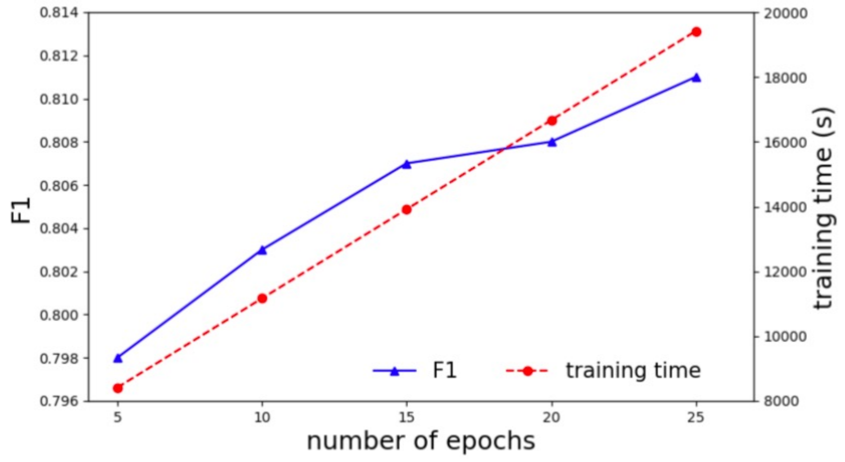
- **Validating the Synthetic Framework**
  - One model/machine
  - One model for all
  - CTF w/o transfer

Methods	F1	Precision	Recall	Training time
CTF	0.830	0.785	0.881	7900
One model/machine[a]	0.842	0.820	0.864	168150
One model for all	0.796	0.791	0.802	5493
CTF w/o transfer	0.798	0.758	0.843	8413

[a] We evaluate 10% machine entities in this method.

Comparison with model variations

2 hours
vs
2 days



F1 and training time under different numbers of epochs for CTF w/o transfer

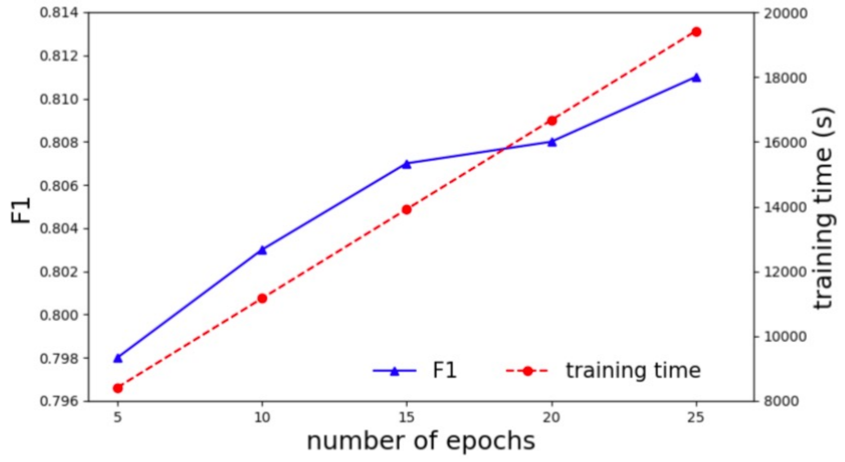
Overall performance

- **Validating the Synthetic Framework**
  - One model/machine
  - One model for all
  - CTF w/o transfer

Methods	F1	Precision	Recall	Training time
CTF	0.830	0.785	0.881	7900
One model/machine[a]	0.842	0.820	0.864	168150
One model for all	0.796	0.791	0.802	5493
CTF w/o transfer	0.798	0.758	0.843	8413

[a] We evaluate 10% machine entities in this method.

Comparison with model variations



F1 and training time under different numbers of epochs for CTF w/o transfer

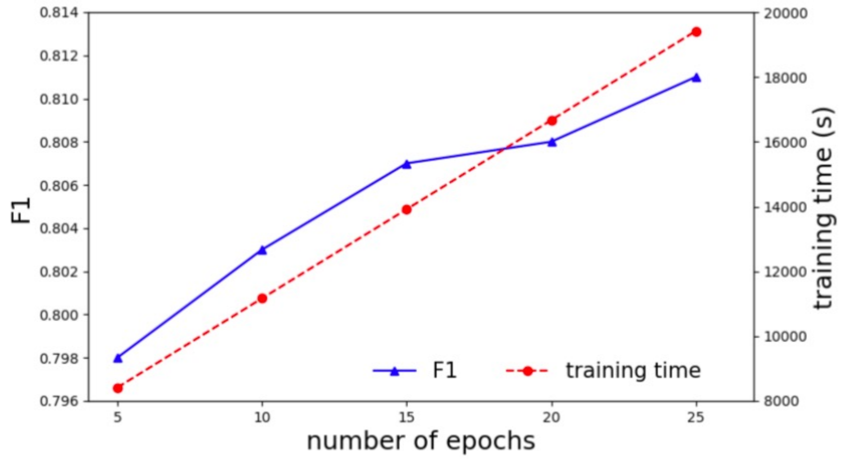
Overall performance

- **Validating the Synthetic Framework**
  - One model/machine
  - One model for all
  - CTF w/o transfer

Methods	F1	Precision	Recall	Training time
CTF	0.830	0.785	0.881	7900
One model/machine[a]	0.842	0.820	0.864	168150
One model for all	0.796	0.791	0.802	5493
CTF w/o transfer	0.798	0.758	0.843	8413

[a] We evaluate 10% machine entities in this method.

Comparison with model variations



F1 and training time under different numbers of epochs for CTF w/o transfer

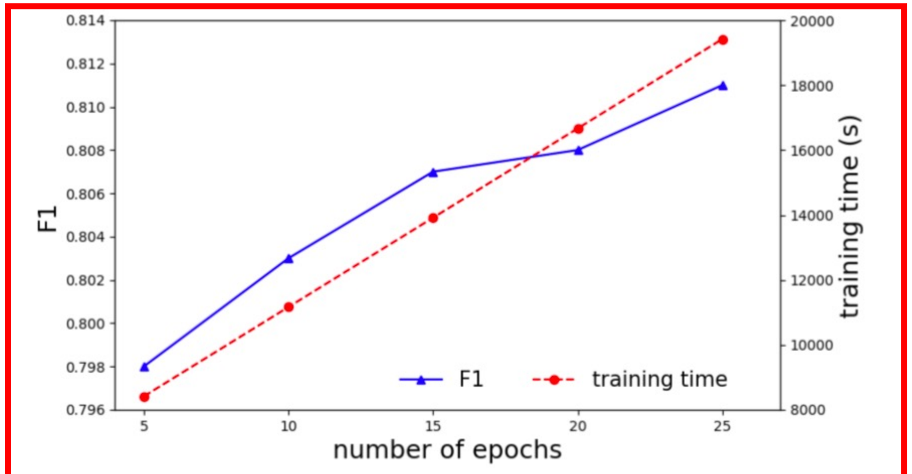
Overall performance

- **Validating the Synthetic Framework**
  - One model/machine
  - One model for all
  - CTF w/o transfer

Methods	F1	Precision	Recall	Training time
CTF	0.830	0.785	0.881	7900
One model/machine[a]	0.842	0.820	0.864	168150
One model for all	0.796	0.791	0.802	5493
CTF w/o transfer	0.798	0.758	0.843	8413

[a] We evaluate 10% machine entities in this method.

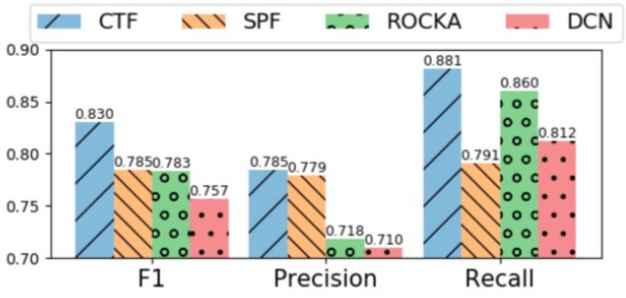
Comparison with model variations

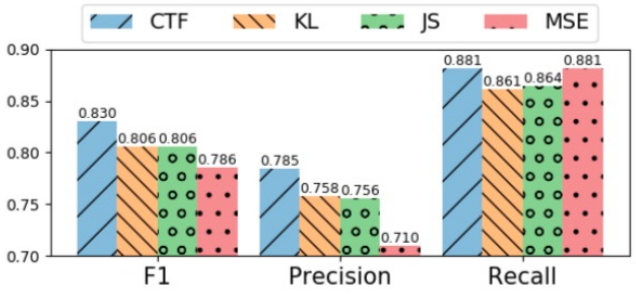


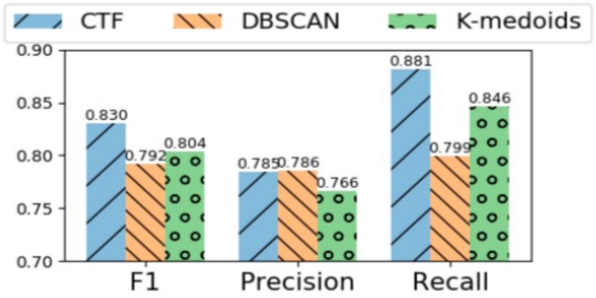
F1 and training time under different numbers of epochs for CTF w/o transfer

Validating Design Choices

- **Choice of Clustering Objects**
  - SPF, ROCKA, DCN
- **Choice of Distance Measures**
  - KL divergence, JS divergence, mean squared error
- **Choice of Clustering Algorithms**
  - DBSCAN, K-medoids







Outline

Background

Design

Evaluation

Conclusion

Conclusion

- CTF: synthetic framework, high-dimensional time series (machine, KPI, time)
- Techniques: $\boldsymbol{z_t}$ distribution clustering, model reuse, fine-tuning
- Evaluation: CTF scalability and effectiveness
- Labeling tool + labeled dataset

CTF can reduce the model training time from about two months ($O(M \cdot T_m)$) to 4.40 hours ($O(M \cdot T_f) + O(K \cdot T_m)$ $(M \gg K, T_m \gg T_f)$) for one hundred thousand machines. It achieves an F1-Score of 0.830, with only 0.012 performance loss.

Thank you!

Q & A

sunm19@mails.tsinghua.edu.cn

INFOCOM 2021