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

清华大学 Tsinghua University
南開大學 Nankai University
ByteDance 字节跳动

Outline

Background

Design

Evaluation

Conclusion

Outline

Background

Design

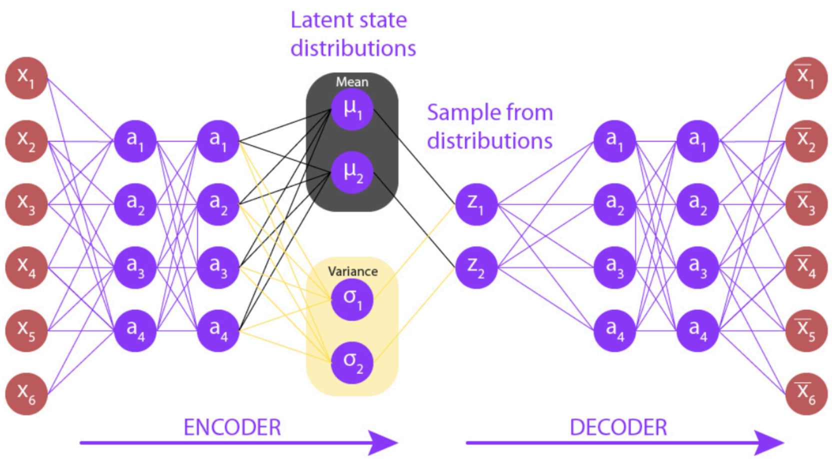
Evaluation

Conclusion

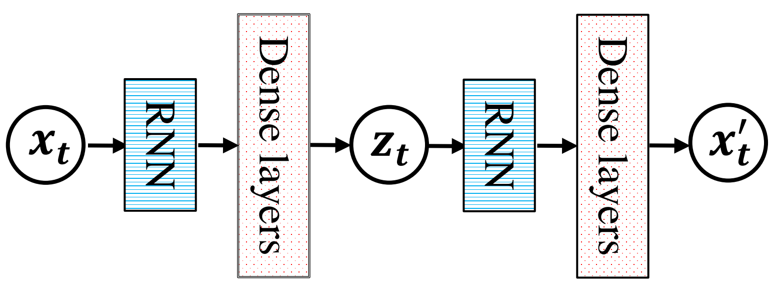
DL Algorithms in the Infra Operation

- 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

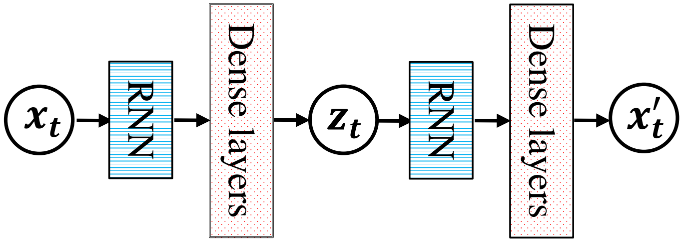
Coarse-grained model -> clustering -> fine-grained models

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

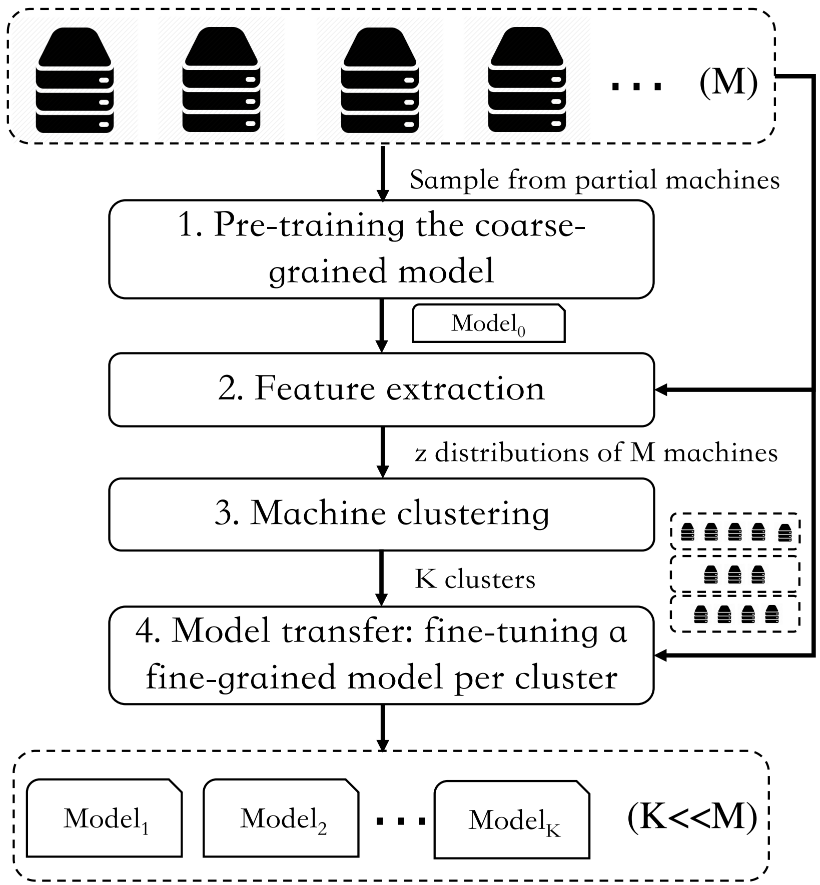
Background

Design

Evaluation

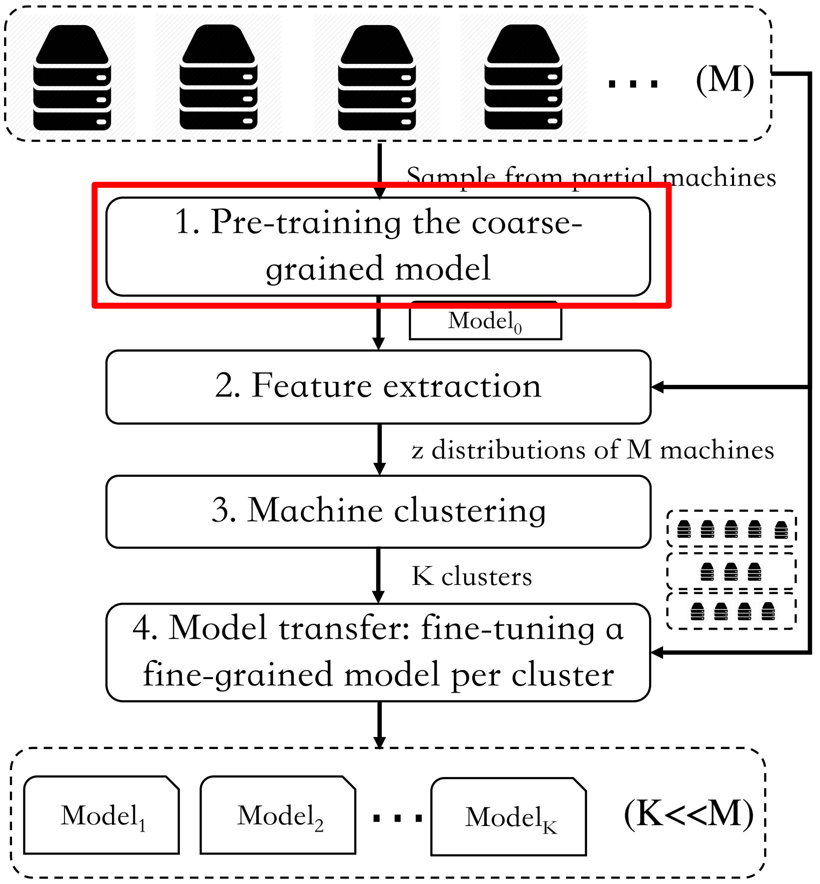
Conclusion

Framework of model training



Framework of model training

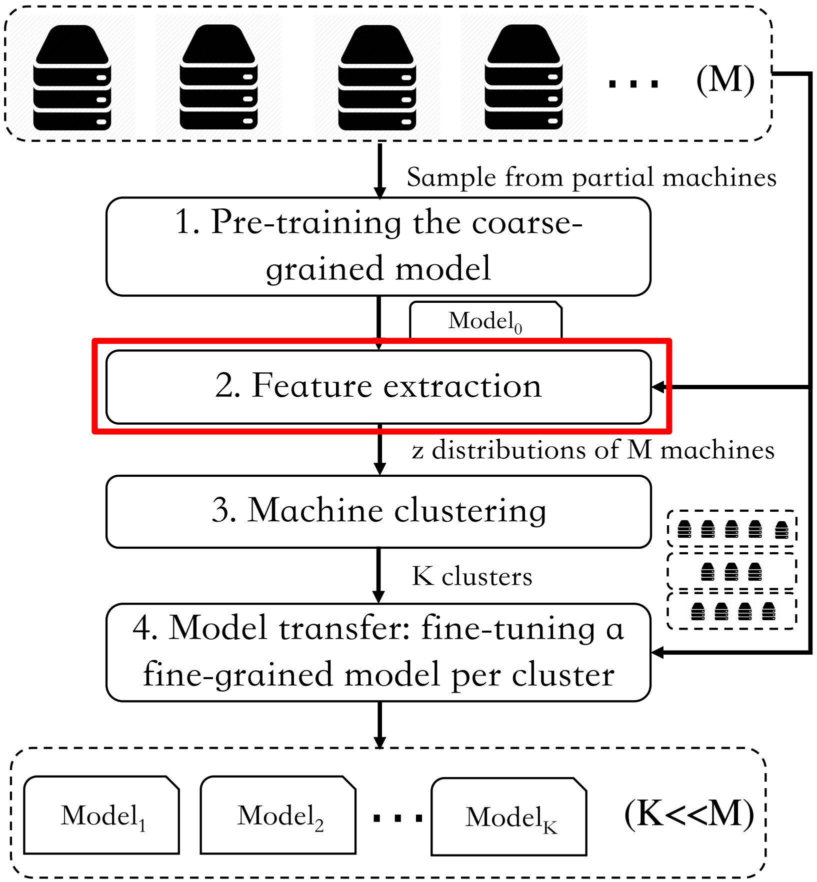
Framework of model training



Framework of model training

- Sampling strategy:
  - Machine sampling
  - Time sampling

Framework of model training



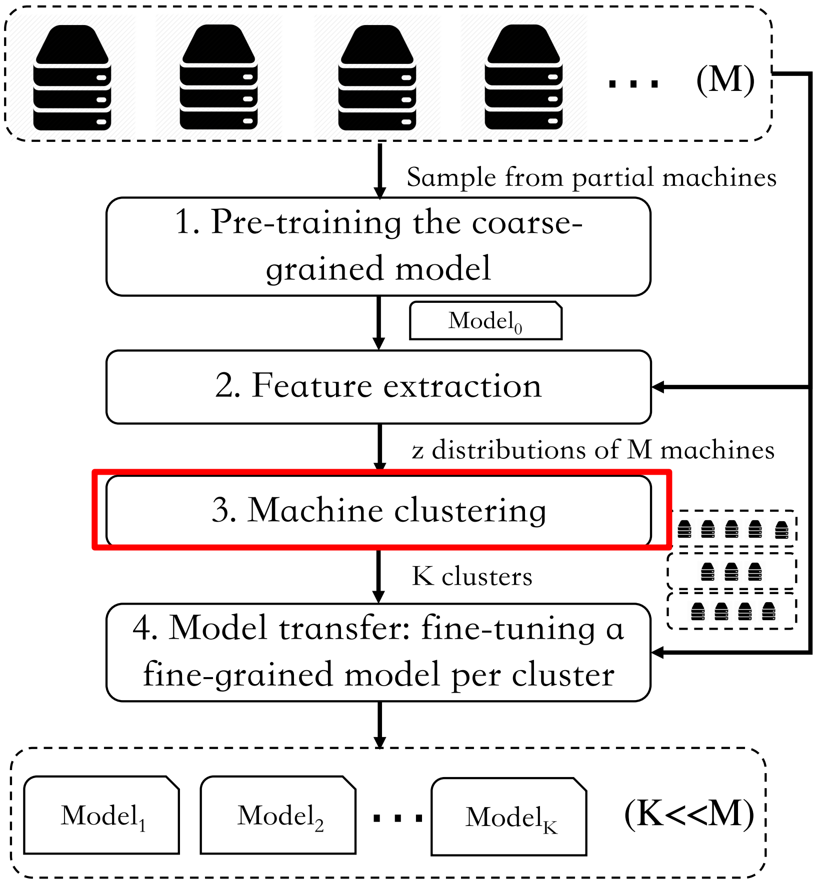
Framework of model training

x_t sequence

z_t sequence

z_t distribution

Framework of model training



Framework of model training

z_t distribution

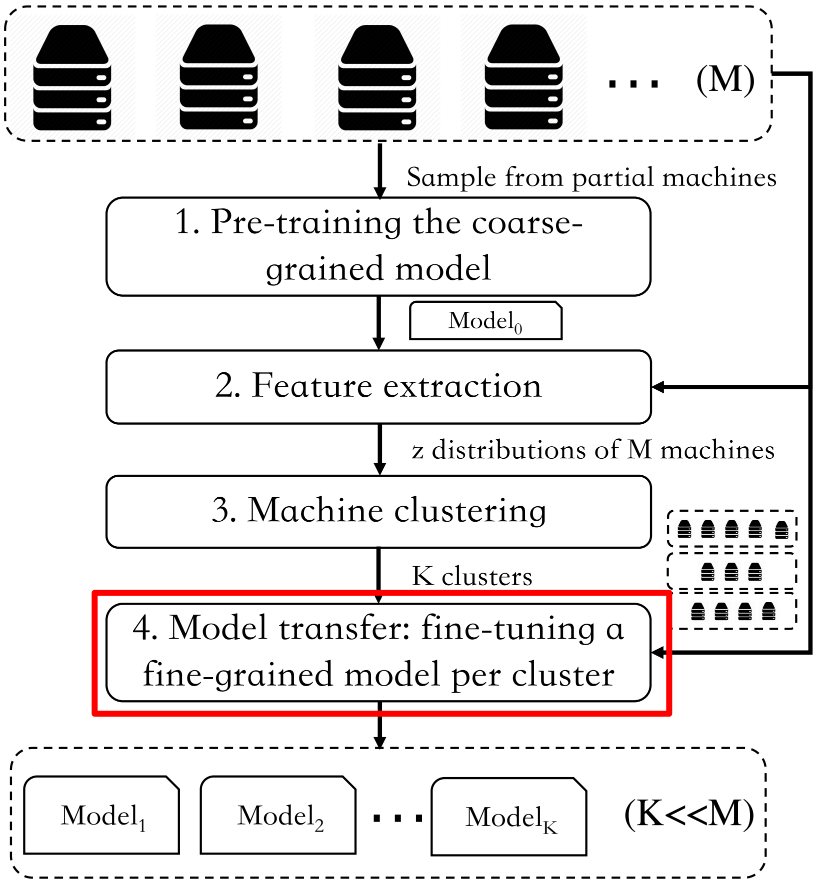
Wasserstein distance

distance matrix

HAC algorithm

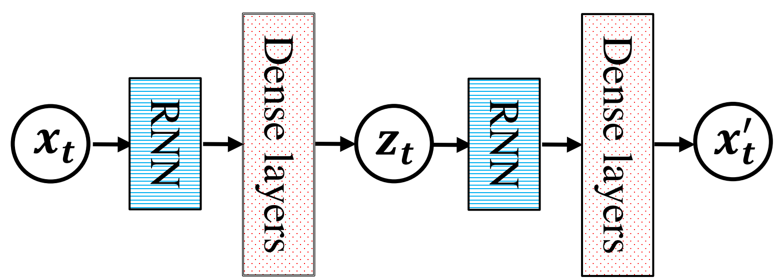
clustering results

Framework of model training

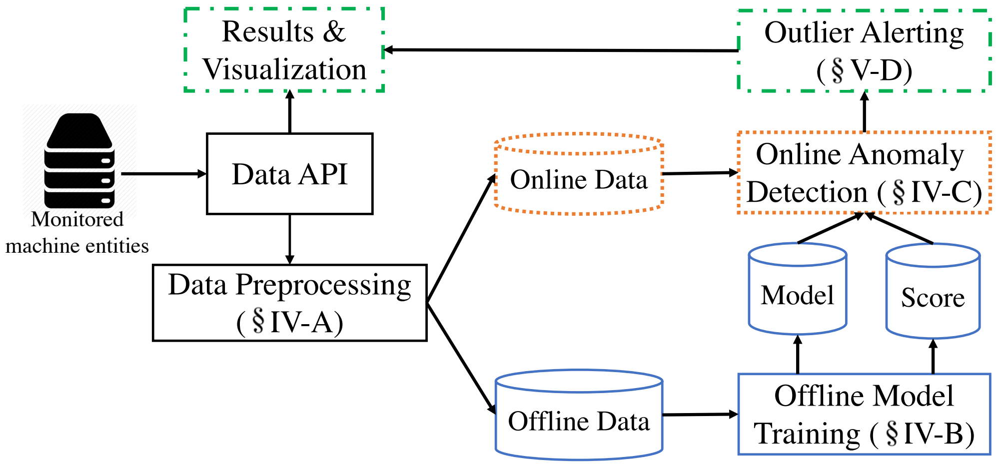


Framework of model training

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



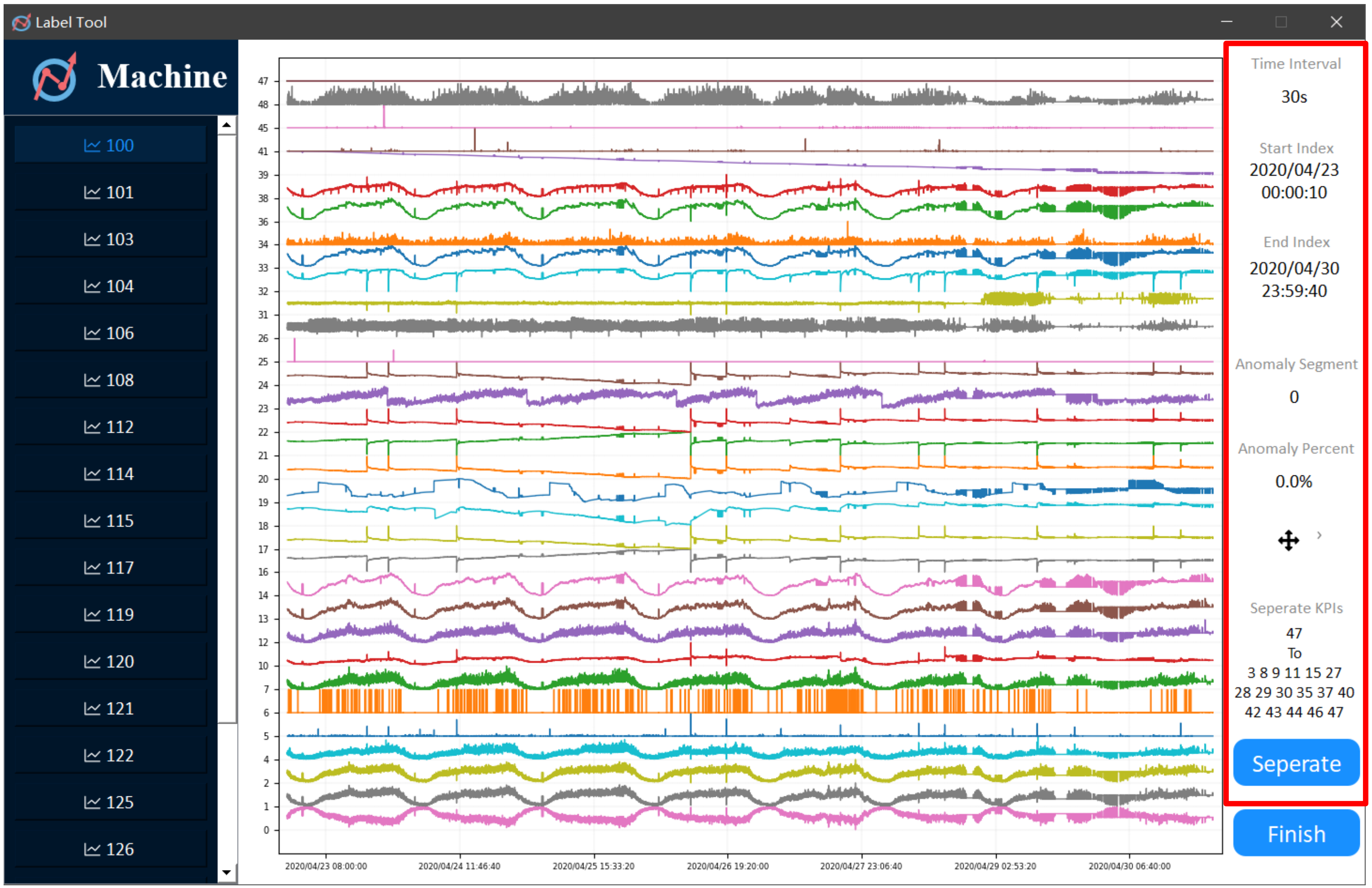
System architecture



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

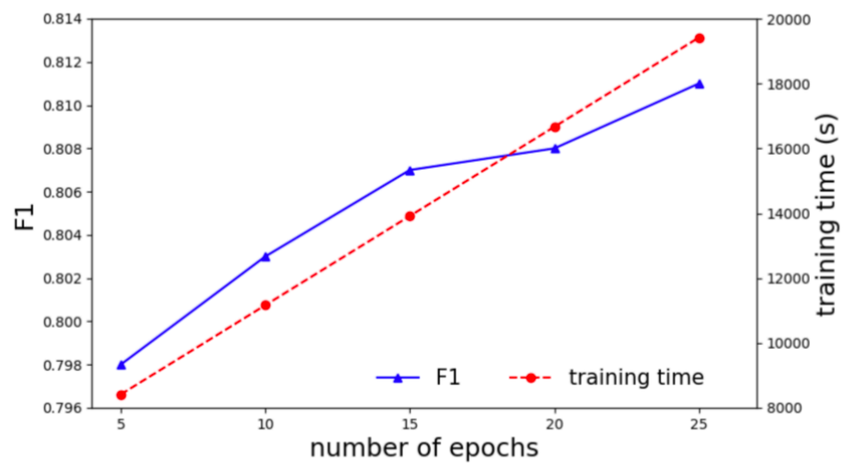
Overall performance

- **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



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

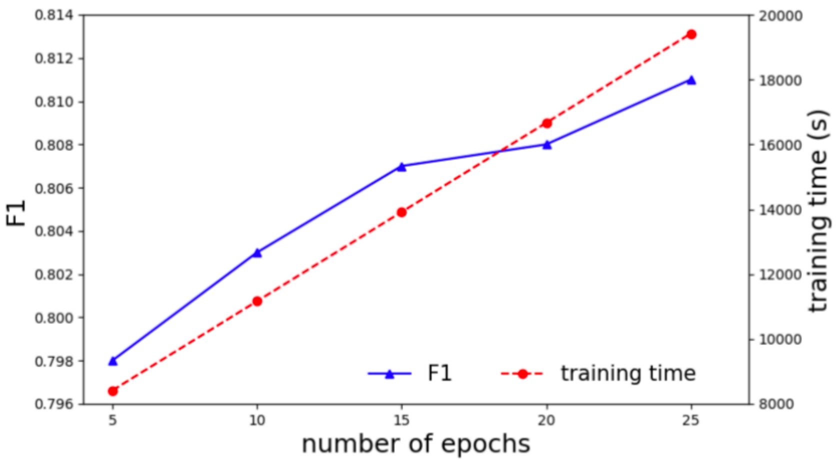
Overall performance

- **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

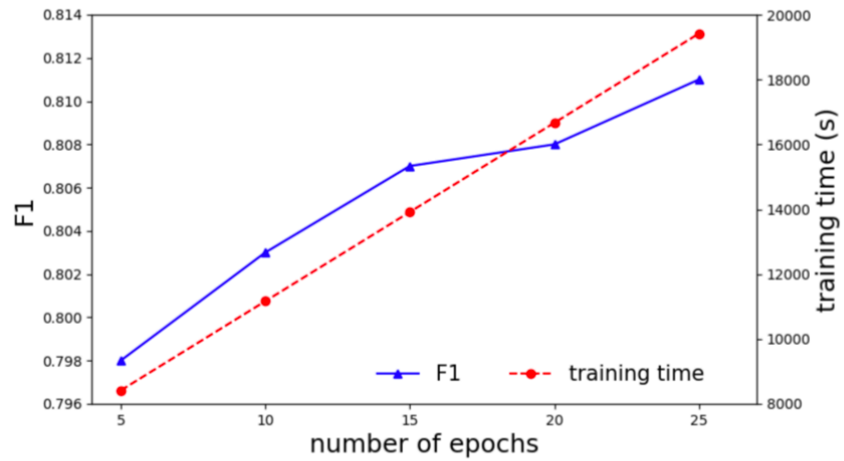
Overall performance

- **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

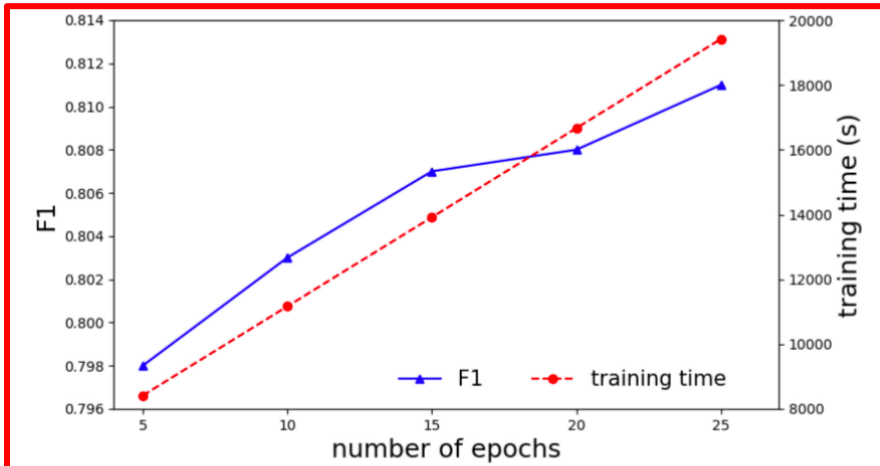
Overall performance

- **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.

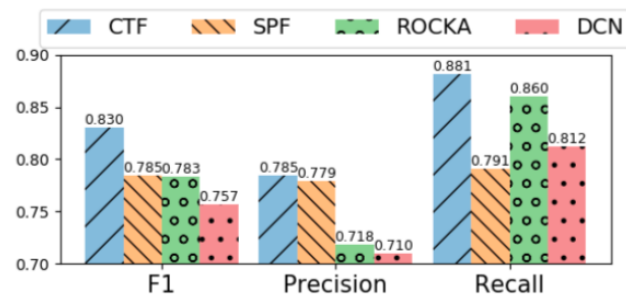
Comparison with model variations

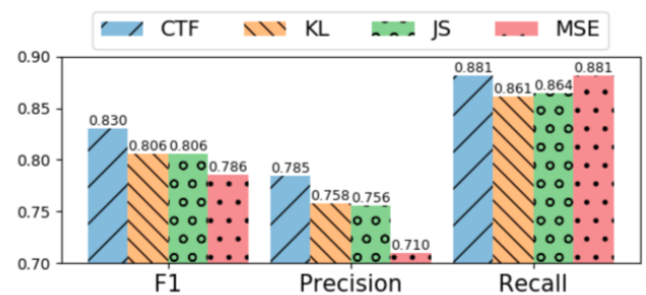


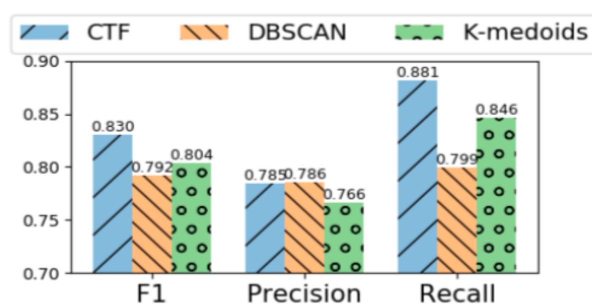
F1 and training time under different numbers of epochs for CTF w/o transfer

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

Thank you!

Q & A

sunm19@mails.tsinghua.edu.cn

INFOCOM 2021