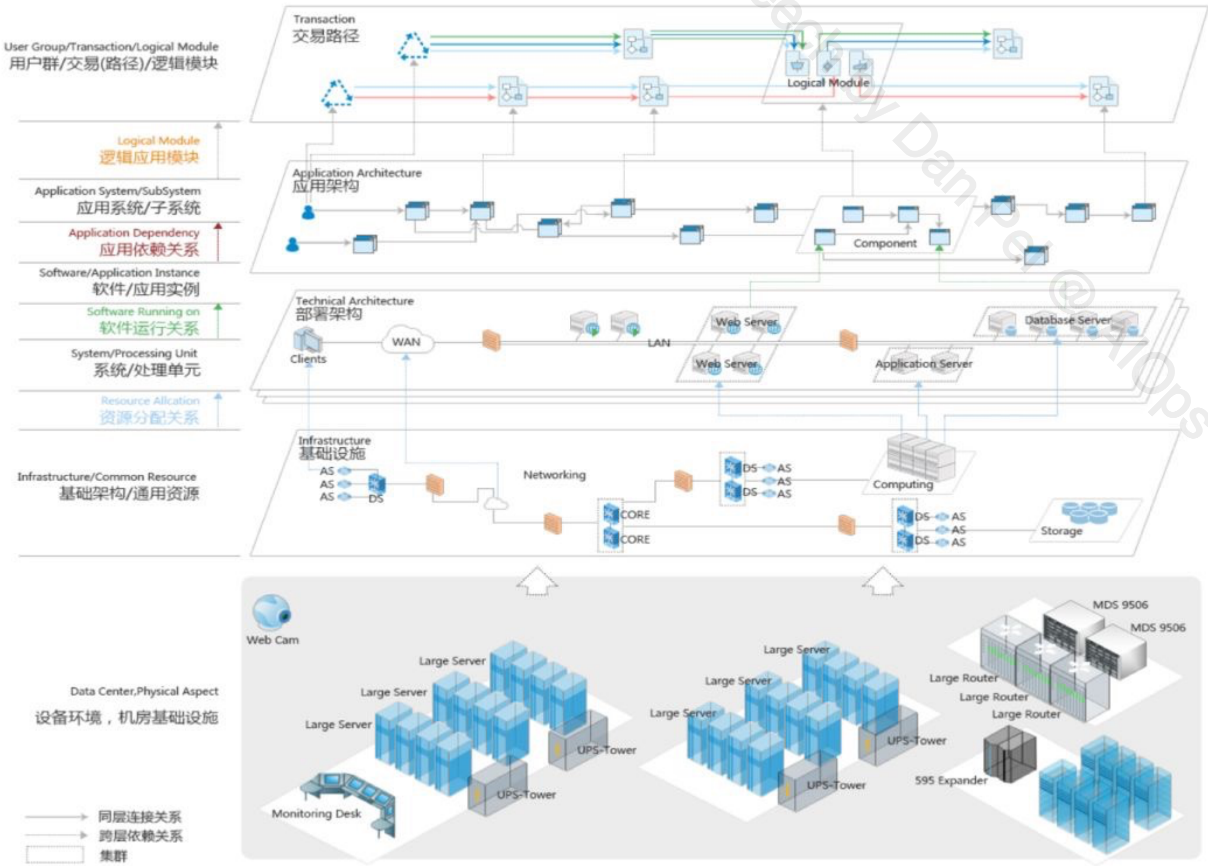
Towards Autonomous IT Operations through Artificial Intelligence

Dan Pei

Tsinghua University

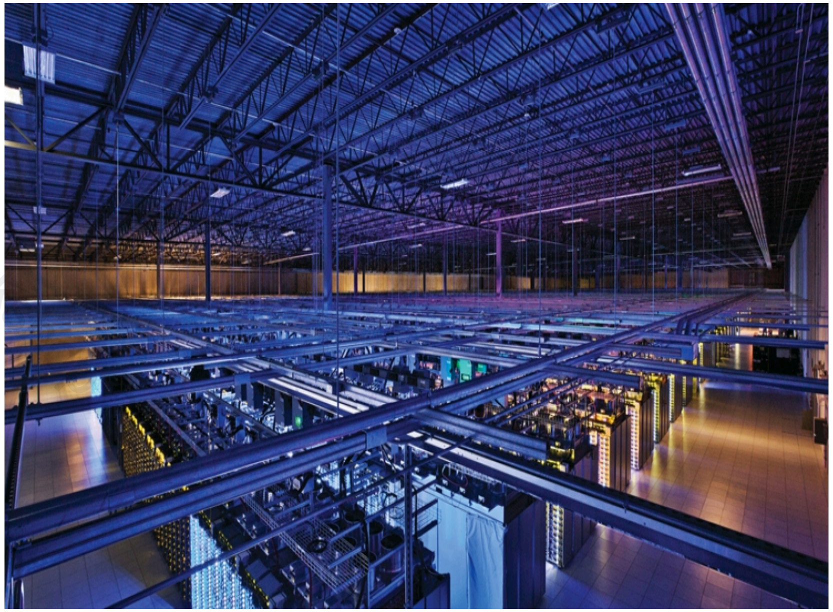
清華大學 | Tsinghua NetMan

IT Operations is one of the technology foundations of the increasingly digitalized world.



WeChat TikTok NETFLIX

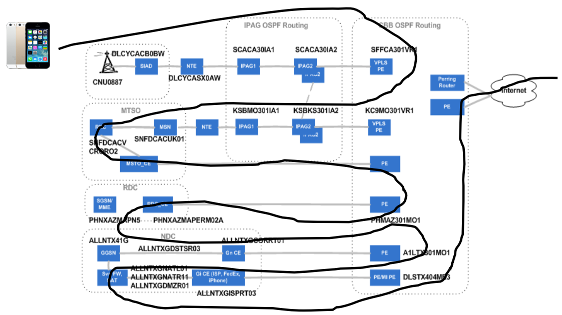
facebook YouTube



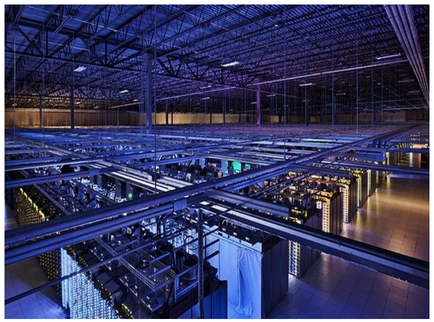
IT Operations

Responsible for ensuring the digitalized businesses and societies run reliably, efficiently and safely, despite the inevitable failures of the imperfect underlying hardware and software.

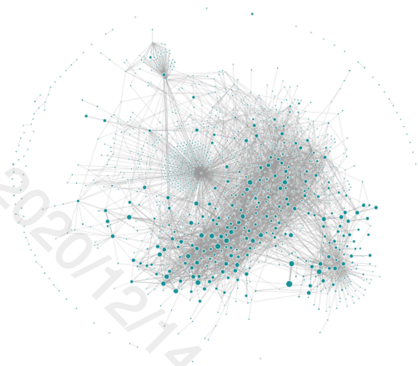
Large & complex
access network



Large & complex
data center



Large & complex
application software



Some IT Operations Companies

All collect IT Operations data and started to offer AIOps (AI for IT Operations) products

Valued at 105 Billion USD

Valued at 25 Billion USD

Valued at
11 Billion USD

Valued at
30 Billion USD

Valued at
2.7 Billion USD

“Internet needs an AI-based knowledge plane”
--- Dave Clark in his SIGCOMM 2003 paper.

A Knowledge Plane for the Internet

David D. Clark*, Craig Partridge◆, J. Christopher Ramming† and John T.

*M.I.T Lab for Computer Science
200 Technology Square
Cambridge, MA 02139
{ddc,jtw}@lcs.mit.edu

◆BBN Technologies
10 Moulton St
Cambridge, MA 02138
craig@bbn.com

†SRI
333 Rav
Menlo Pa
chrisramm

ABSTRACT

We propose a new objective for network research: to build a fundamentally different sort of network that can assemble itself given high level instructions, reassemble itself as requirements change, automatically discover when something goes wrong, and automatically fix a detected problem or explain why it cannot do so.

We further argue that to achieve this goal, it is not sufficient to improve incrementally on the techniques and algorithms we know today. Instead, we propose a new construct, the Knowledge Plane, a pervasive system within the network that builds and maintains high-level models of what the network is supposed to do, in order to provide services and advice to other elements of the network. The knowledge plane is novel in its reliance on the tools of AI and cognitive systems. We argue that cognitive techniques, rather than traditional algorithmic approaches, are best suited to meeting the uncertainties and complexity of our objective.

transparent network with rich end-sy
deeply embedded assumption of
administrative structure are critical stre
users when something fails, and high
much manual configuration, diagnosis a

Both user and operator frustrations aris
design principle of the Internet—the
with intelligence at the edges [1,2].
without knowing what that data is, or
combination of events is keeping dat
edge may recognize that there is a prob
that something is wrong, because the c
be happening. The edge understands
expected behavior is; the core only dea
network operator interacts with the core
as per-router configuration of routes a
for the operator to express, or the netw

From 1981 to 1989, he acted as **chief protocol architect** in the development of the Internet, and chaired Internet Architecture Board

Industry opinions on AI's role in IT operations

Huawei CEO Ren Zhengfei:

"AI is the most important tool for managing the networks.

一、巨大的存量网络是人工智能最好的舞台

为什么要聚焦GTS、把人工智能的能力在服务领域先做好呢？对于越来越庞大、越来越复杂的网络，人工智能是我们建设和管理网络的最重要的工具，人工智能也要聚焦在服务主航道上，这样发展人工智能就是发展主航道业务，我们要放到这个高度来看。如果人工智能支持GTS把服务做好，五年以后我们自已的问题解决了，我们的人工智能又是世界一流。

首先，是解决我们在全球巨大的网络存量的网络维护、故障诊断与处理的能力的提升。我们在全球网络存量有一万亿美元，而且每年上千亿的增加。容量越来越大，流量越来越快，技术越来越复杂，维护人员的水平要求越来越高，经验要求越来越丰富，越来越没有这样多的人才，人工智能，大有前途。

Jeff Dean Head of AI, Google:

"We can (use AI to) improve everywhere in a system that have tunable parameters or heuristics"

Anywhere We've Punted to a User-Tunable Performance Option!

Many programs have huge numbers of tunable command-line flags, usually not changed from their defaults

```
--eventmanager_threads=16
--bigtable_scheduler_batch_size=8
--mapreduce_merge_memory=134217728
--lexicon_cache_size=1048576
--storage_server_rpc_freelist_size=128
...
```

Anywhere We're Using Heuristics To Make a Decision!

Compilers: instruction scheduling, register allocation, loop nest parallelization strategies, ...

Networking: TCP window size decisions, backoff for retransmits, data compression, ...

Operating systems: process scheduling, buffer cache insertion/replacement, file system prefetching, ...

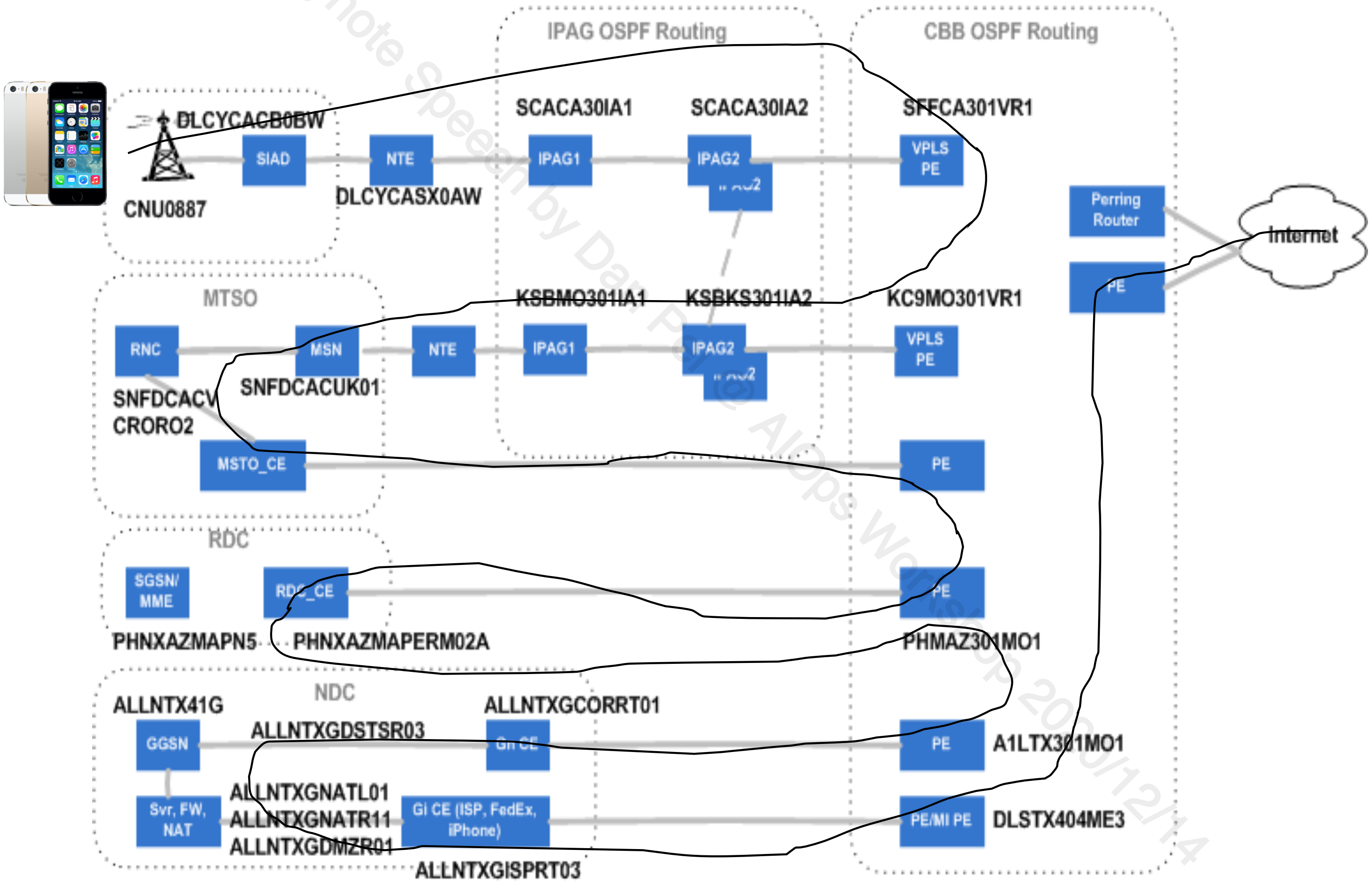
Job scheduling systems: which tasks/VMs to co-locate on same machine, which tasks to pre-empt, ...

ASIC design: physical circuit layout, test case selection, ...

Outline

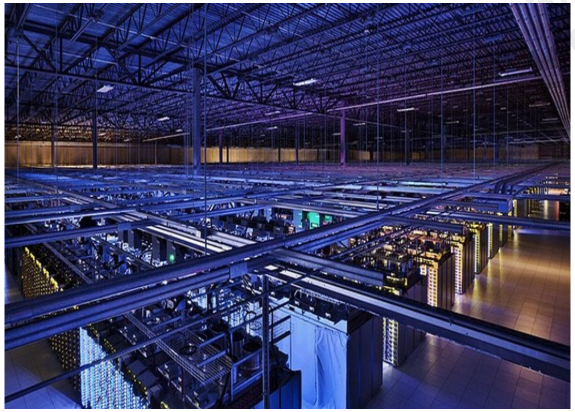
- IT Operations (Ops) background
- *Is machine learning necessary for Ops?*
- Case Study Overview
  - Unsupervised Anomaly Detection in Ops
  - Alert Analysis in Ops
- Lessons Learned

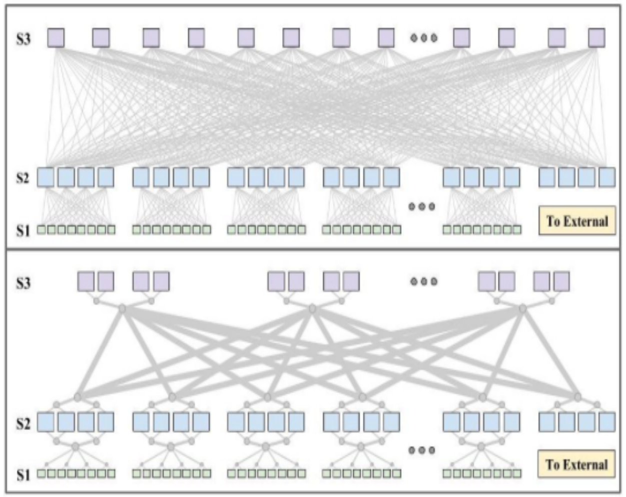
Complex Edge Networks

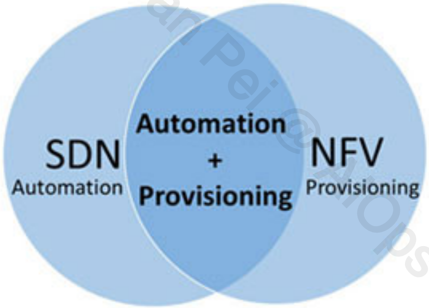


Complex and Evolving Data Center Hardwares

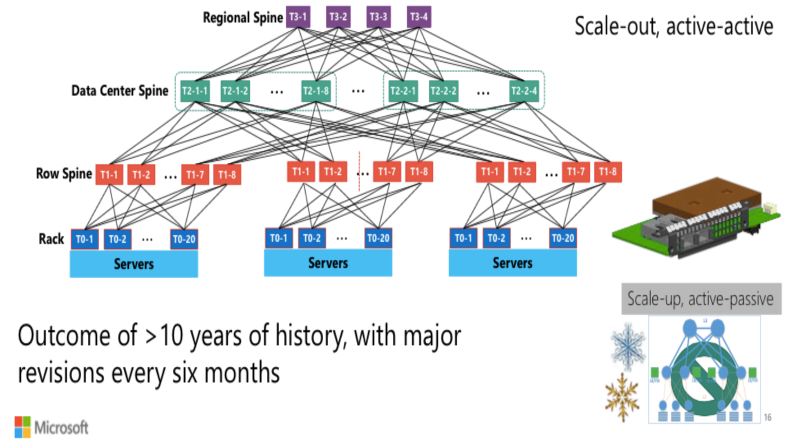
10s of thousands of servers

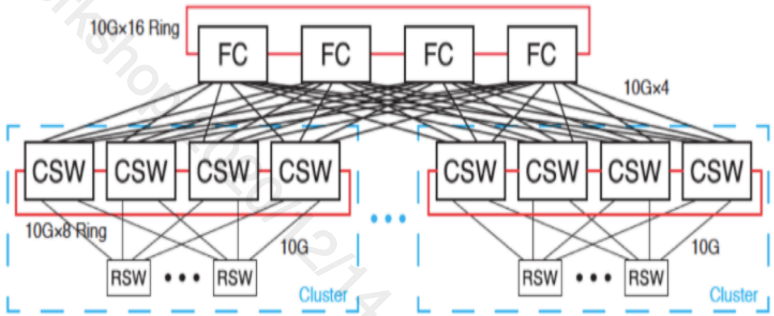






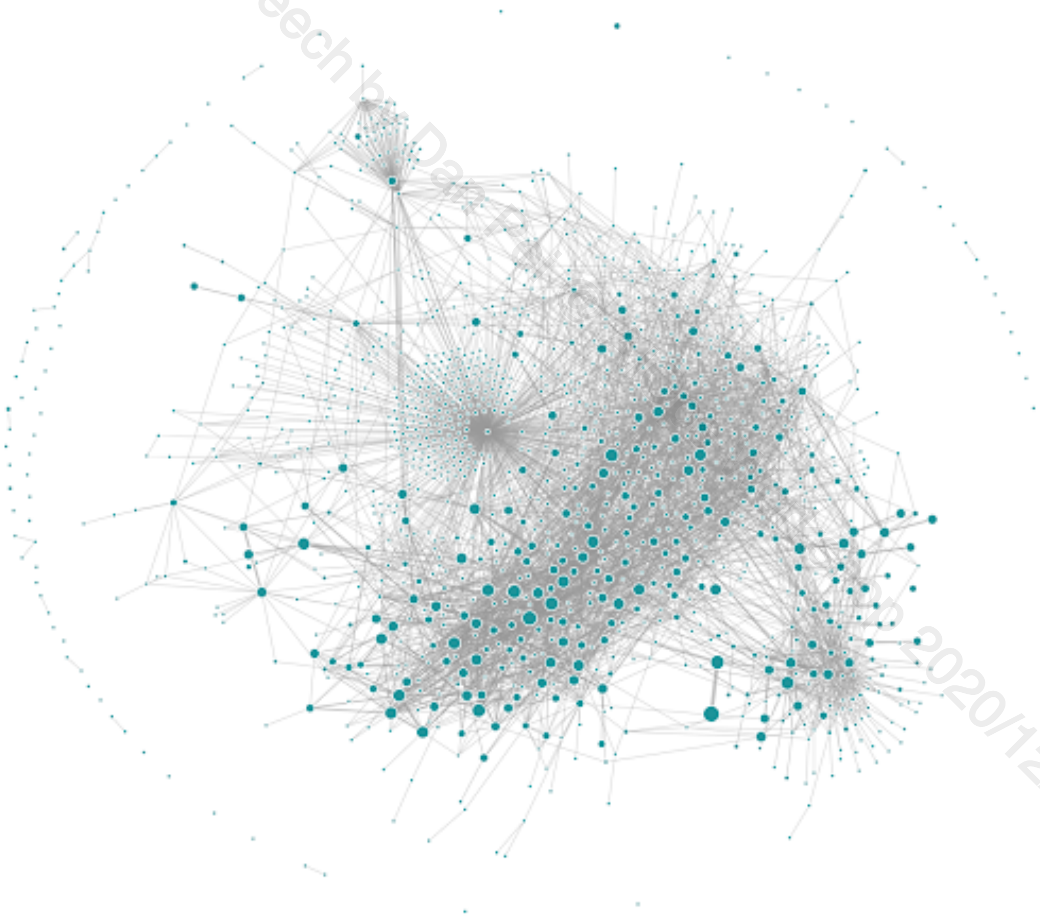
Frequent topology changes





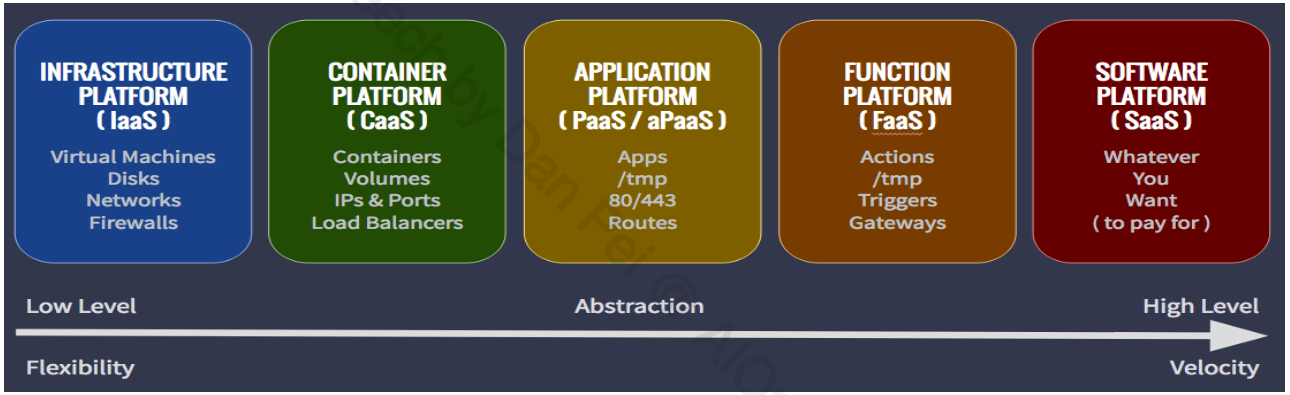
Complex Software Module Dependences

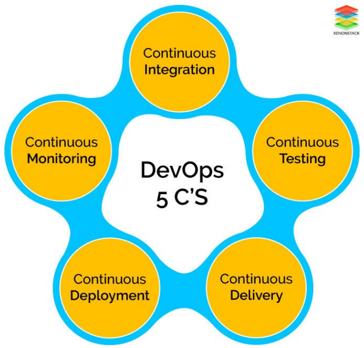
Application dependency at Uber in 2018

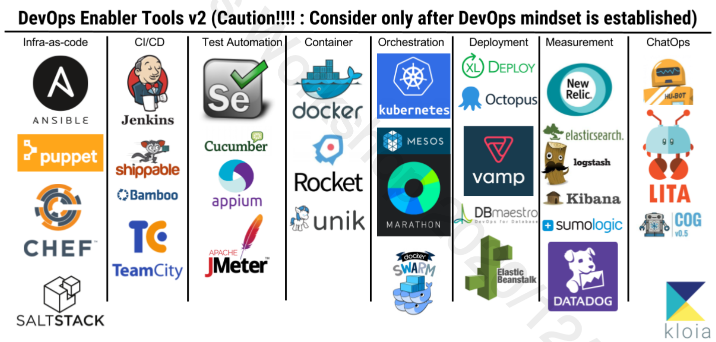


Evolving Techniques Enable Frequent Software Changes, one major cause of failures

10s of thousands software/config changes per day in a large company



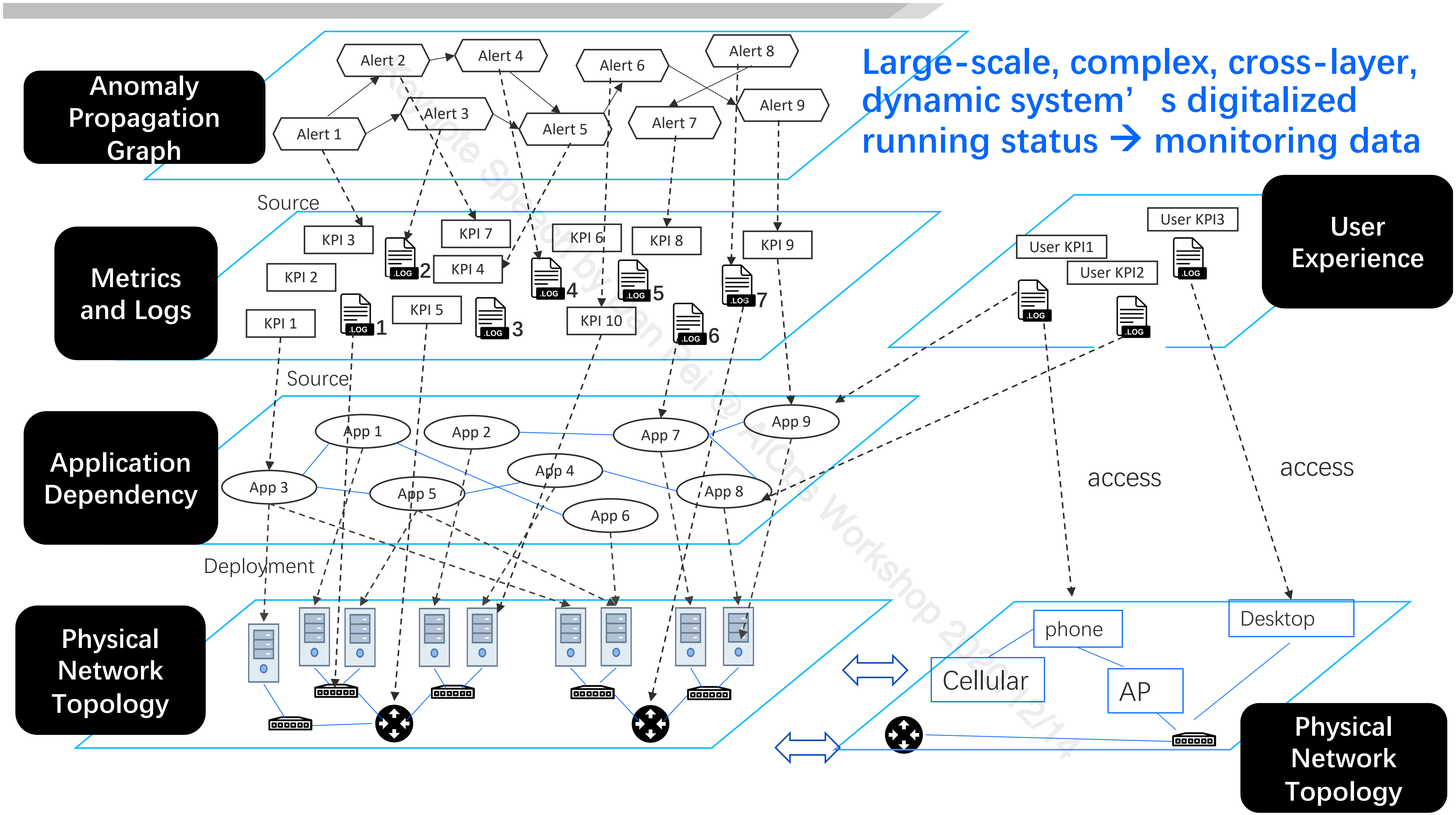




DevOps

Continuous Integration/Continuous Delivery

Large-scale, complex, cross-layer, dynamic system’ s digitalized running status → monitoring data



TeraBytes of Ops data per day overwhelm Ops engineers

Each offers some clues, but due to complexity and volume, each is hard to manually analyze, let alone collectively analyze all data sources.

Software module **Invocation Traces**

Application Performance Monitoring

Metrics

Probing

Alerts

Logs

Configs

Free texts
(**tickets**, change, manual)

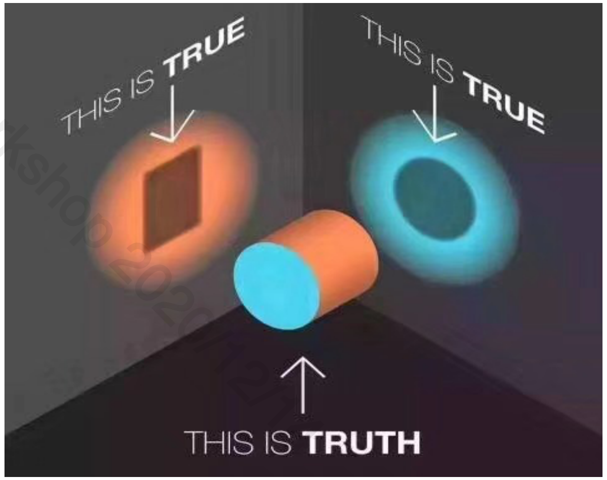
Traffic dump

Social Media

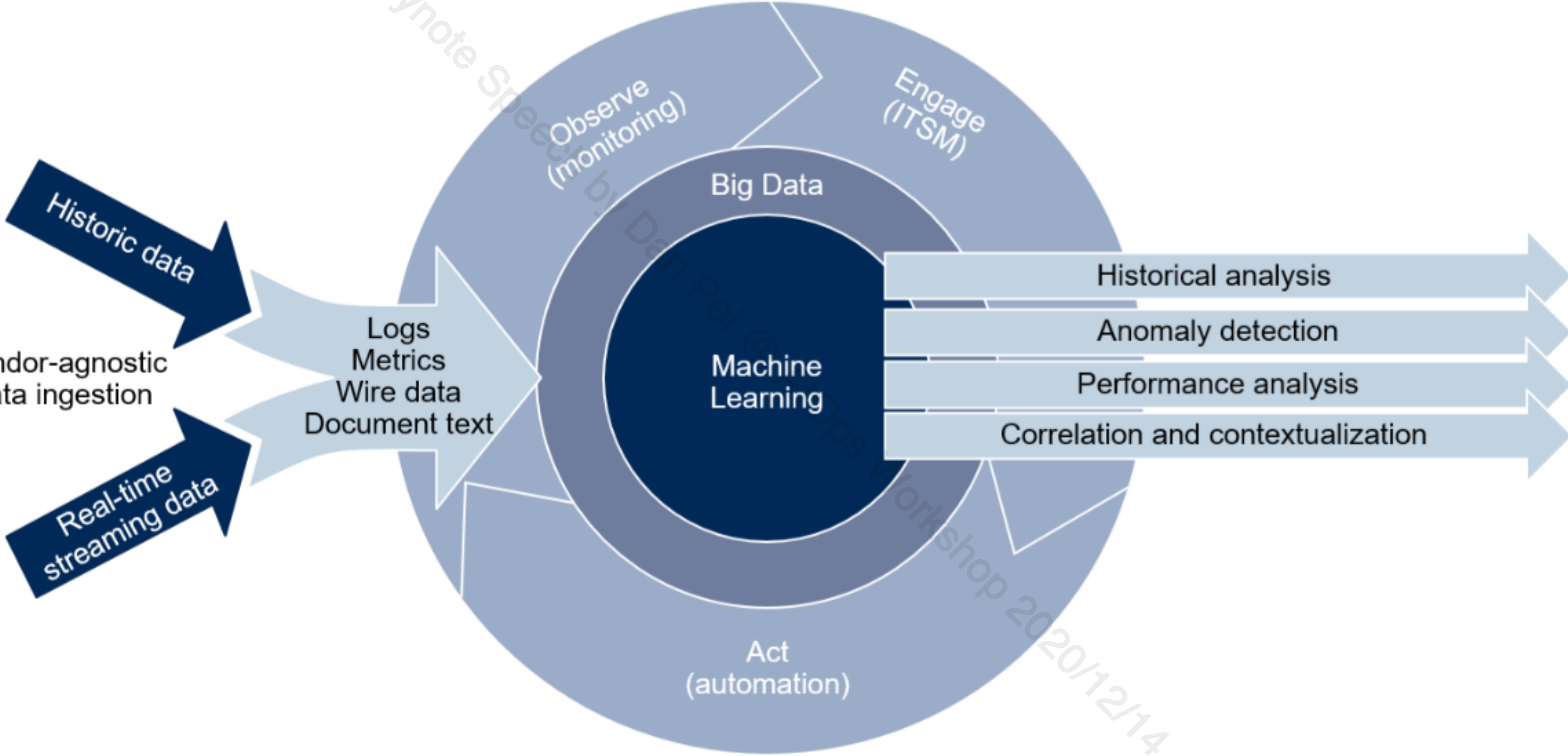
We have no choice but relying on Artificial Intelligence to extract useful signals out of the Big Ops Data which have every low signal-to-noise ratio.

- Volume
- Velocity
- Variety
- Value

We have no choice but relying on Artificial Intelligence to incorporate (expert or mined) knowledge (topology, call graph, causal relationship) to correlate signals.

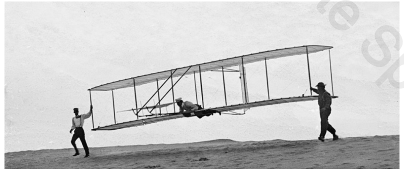


AIOps Platform Enabling Continuous ITOM



ID: 340492

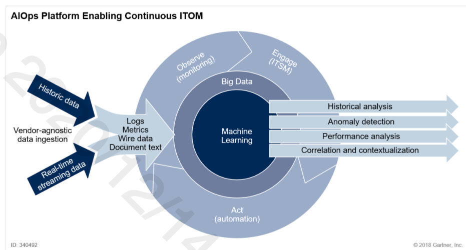
Towards Autonomous IT Operations



Manual and few data

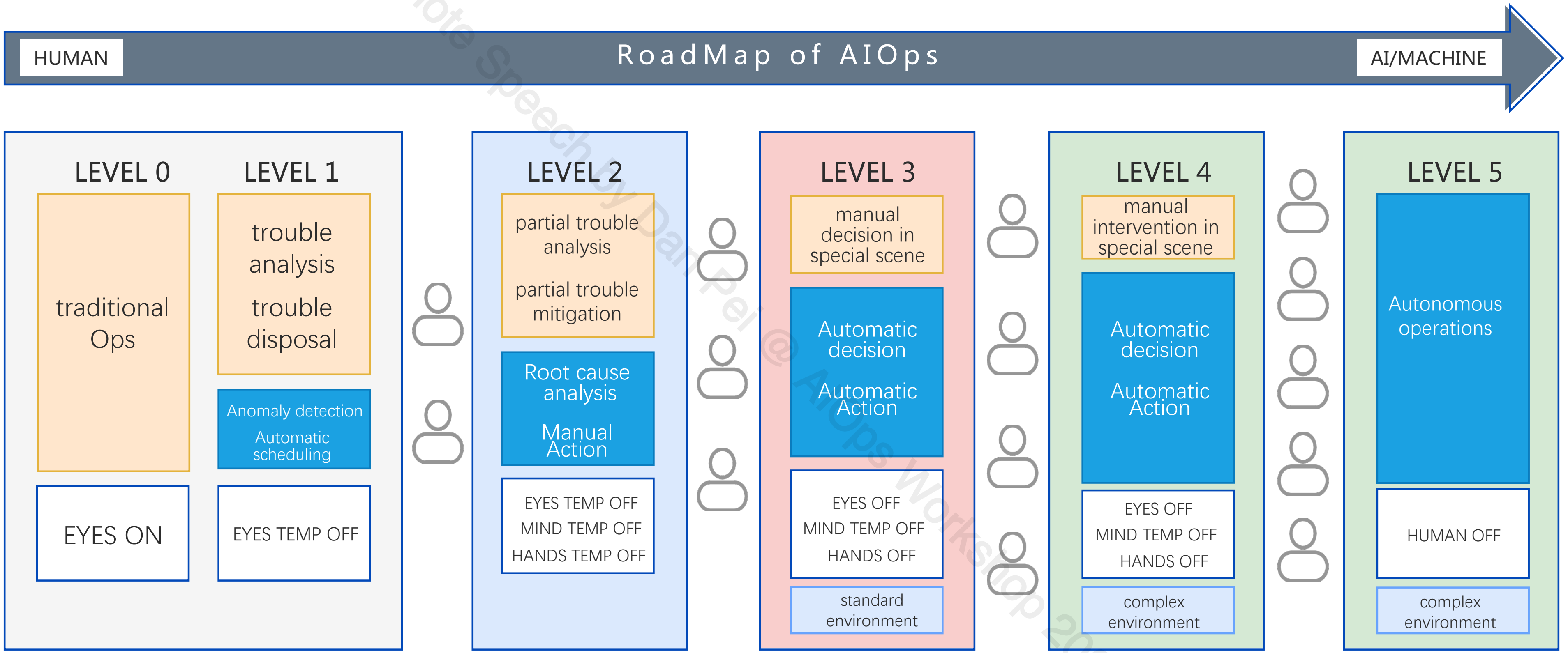
Lots of data but
manual decision

Autonomous



Spaceship Avalon: 5000 passengers and 258 crew members in
hibernation. Flying towards Planet Homestead II, 120-year trip.

Levels of AIOps



Levels of Autonomous IT Operations

- Cores Per Op (CPO) under specific SLA (e.g. 99.5% availability):
  The average number of x86 CPU cores managed by an Op (40hours/week)

Level=⌊ Log (CPO/100) ⌋	Cores Per Op (CPO)	Typical Enterprises
Level 0	O(100)	Finance
Level 1	O(1K)	Medium Internet companies running on public clouds
Level 2	O(10K)	Large Internet companies
Level 3	O(100K)	
Level 4	O(1M)	
Level 5	O(10M)	

Autonomous IT Operations:
use Artificial Intelligence to automatically deal with all causes of changes to IT systems

Software & hardware failures	Automatic Healing
Software changes	Autonomous software deployment
Traffic load changes	Automatic Elastic Resource Allocation
Malicious attacks	Autonomous Defense

Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Case Study
  - Unsupervised Anomaly Detection in Ops
    - *Time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019, KDD2019, INFOCOM2021)*
    - Log anomaly detection (IWQoS 2017, IJCAI 2019, IPCCC2020a, IPCCC2020b, ISSRE2020)
    - Trace anomaly detection (ISSRE 2020)
    - Zero-day attack detection (INFOCOM2020a)
  - Alert Analysis in Ops
    - INFOCOM2020b, ICSE SEIP 2020, FSE 2020
- Lessons Learned

All case studies are from joint work with Industry Collaborators

阿里巴巴 Alibaba.com

Bai du 百度

Tencent 腾讯

ByteDance 字节跳动

HUAWEI

滴滴出行
滴滴一下美好出行

中国建设银行
China Construction Bank

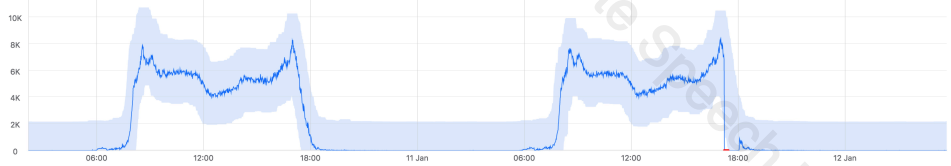
Bank
中国光大银行
CHINA EVERBRIGHT BANK

WeBank
微众银行

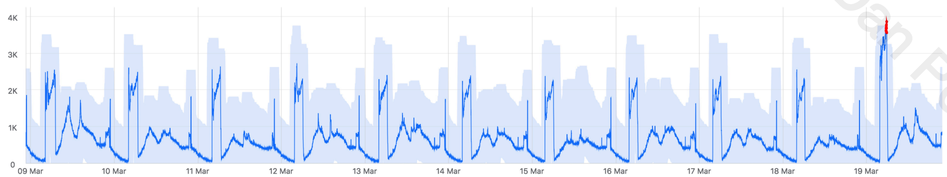
Diverse Metrics and Their Diverse Anomalies

Time series algorithms are needed to parse and make sense of metrics data

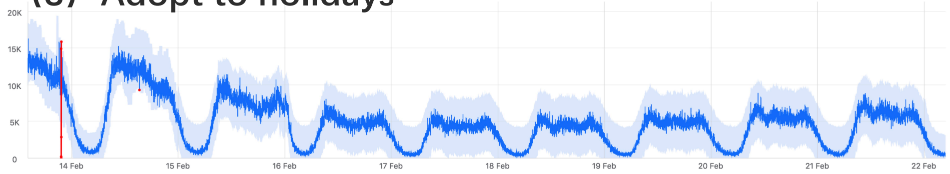
(1) Seasonal metrics



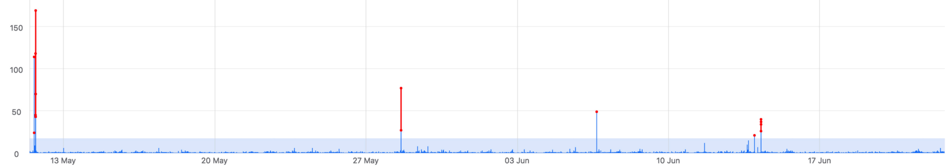
(2) Periodicity shift



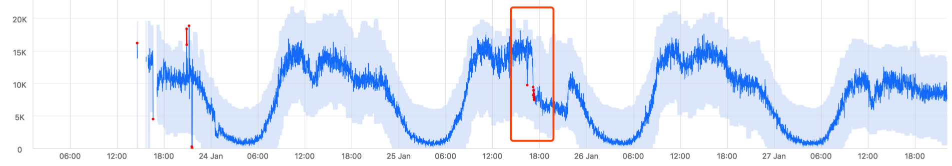
(3) Adopt to holidays



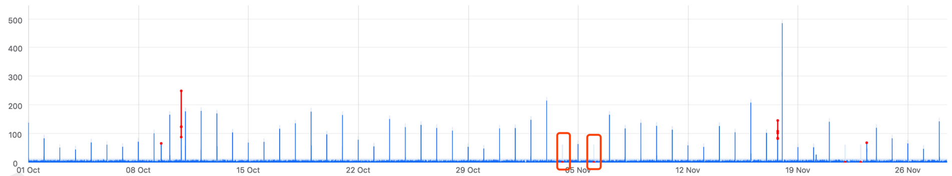
(4) Identify variable metrics and obtain extreme threshold



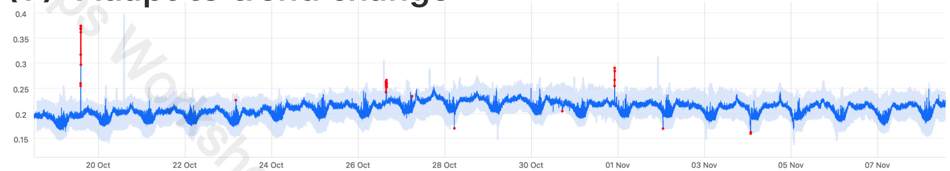
(5) Detect too rapid a change



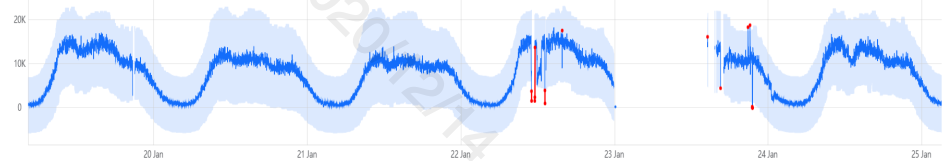
(6) Detect the lack of seasonality.



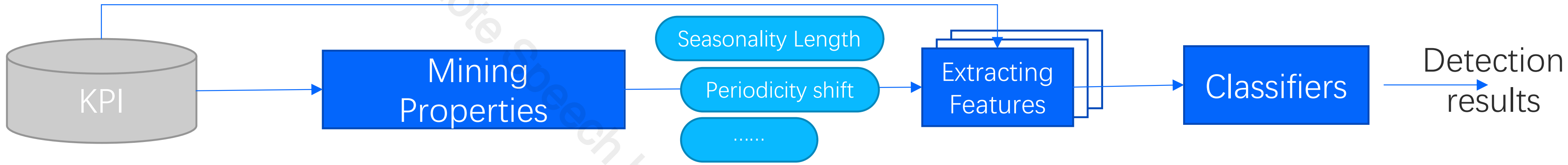
(7) Adapt to trend change

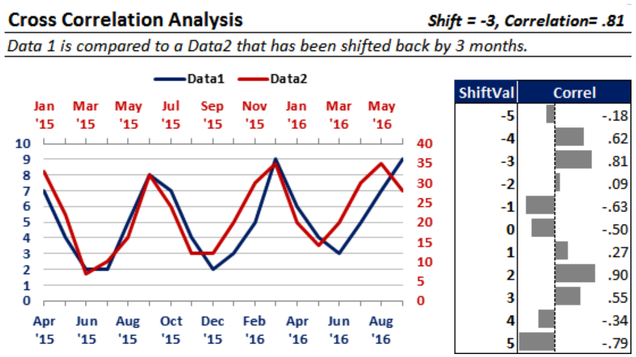


(8) Robust against data loss or interruption

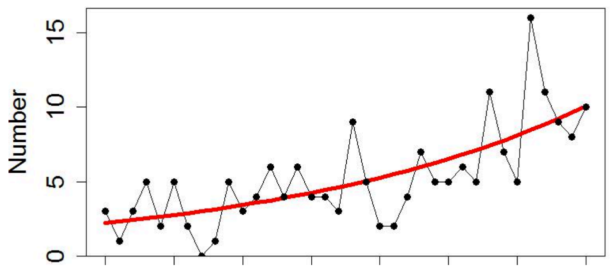


Architecture

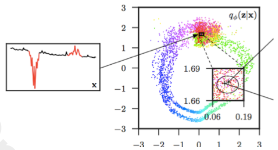




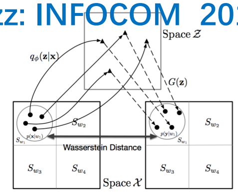
ShiftVal	Correl
-5	-.18
-4	.62
-3	.81
-2	.09
-1	-.63
0	-.50
1	.27
2	.90
3	.55
4	-.34
5	-.79



Donut: WWW2018



Buzz: INFOCOM 2019



Label-Less: INFOCOM 2019

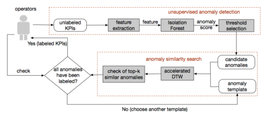
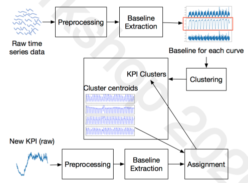
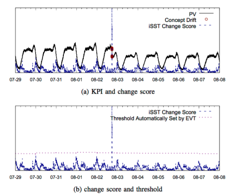


Figure 2: The overall framework of Label-Less.

ROCKA: IWQOS 2018



StepWise: ISSRE 2018 Best Paper



Donut: supervised->unsupervised: smooth KPIs

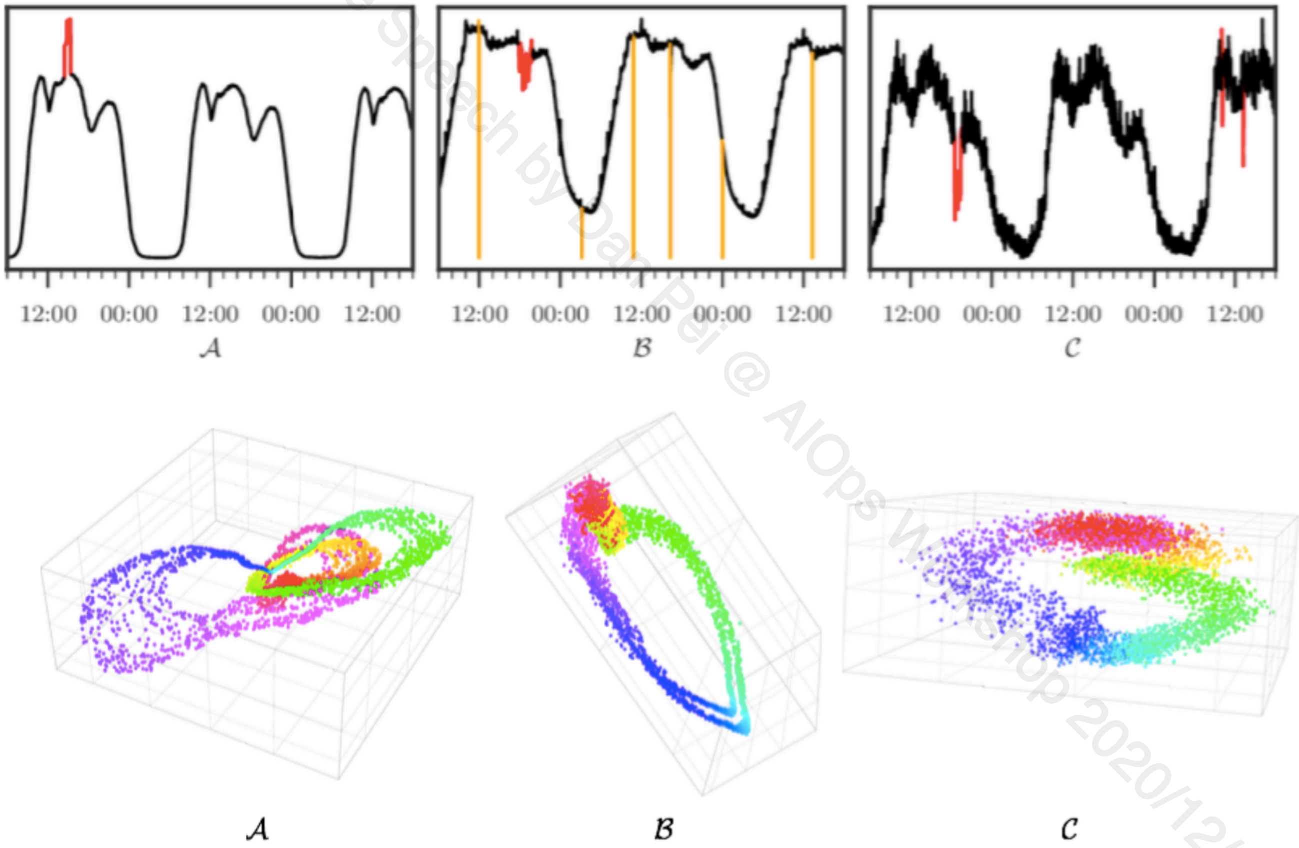
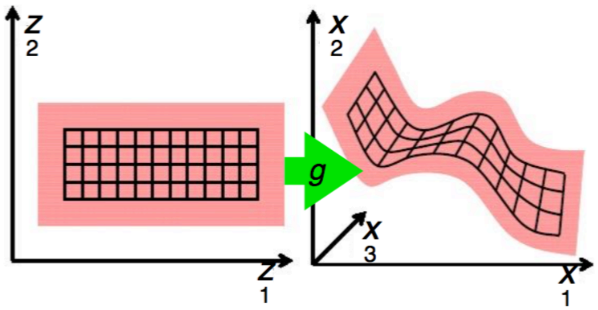
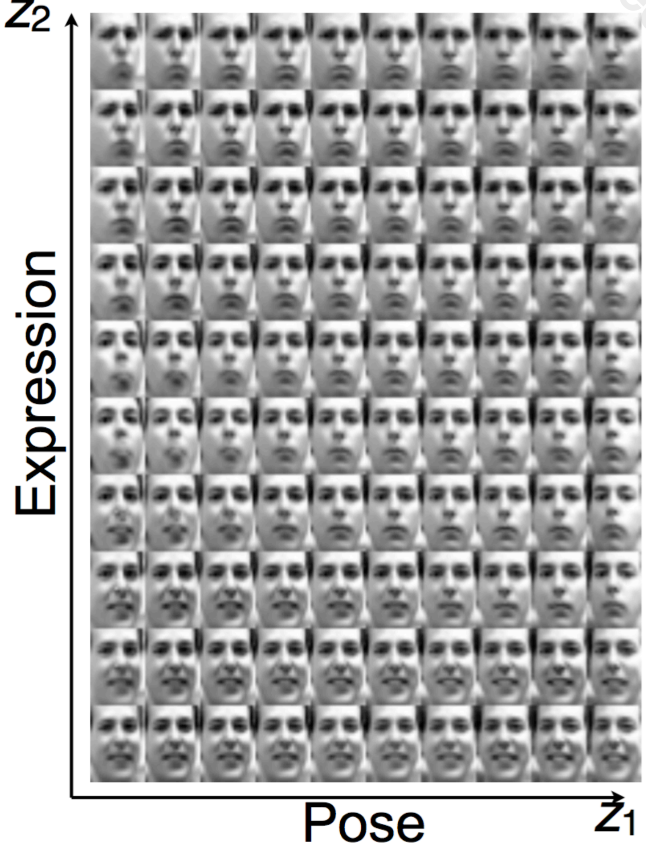


Figure 12: 3-d latent space of all three datasets.

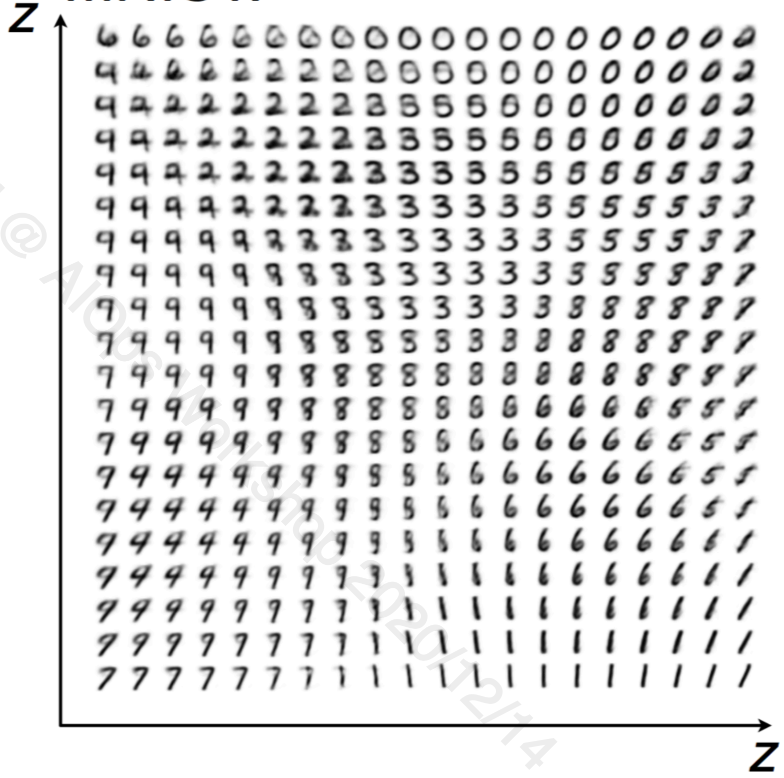
Latent Variable Models



Frey Faces:



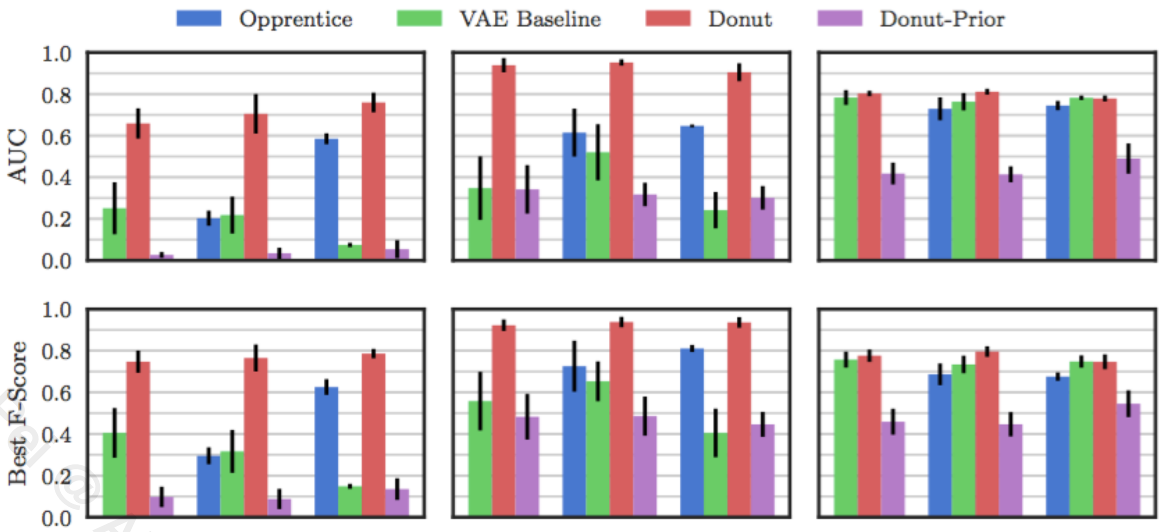
MNIST:

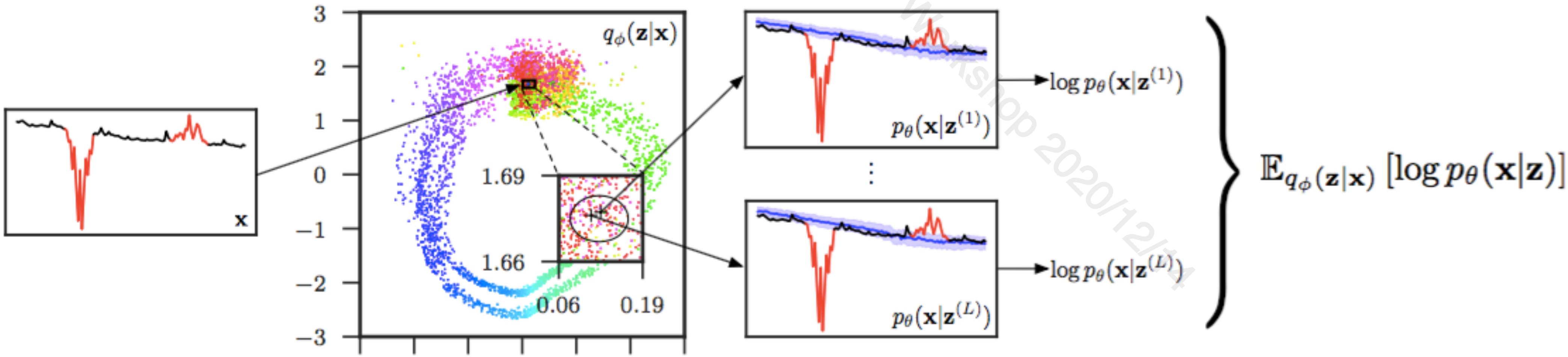


Unsupervised KPI Anomaly Detection Through Variational Auto-Encoder

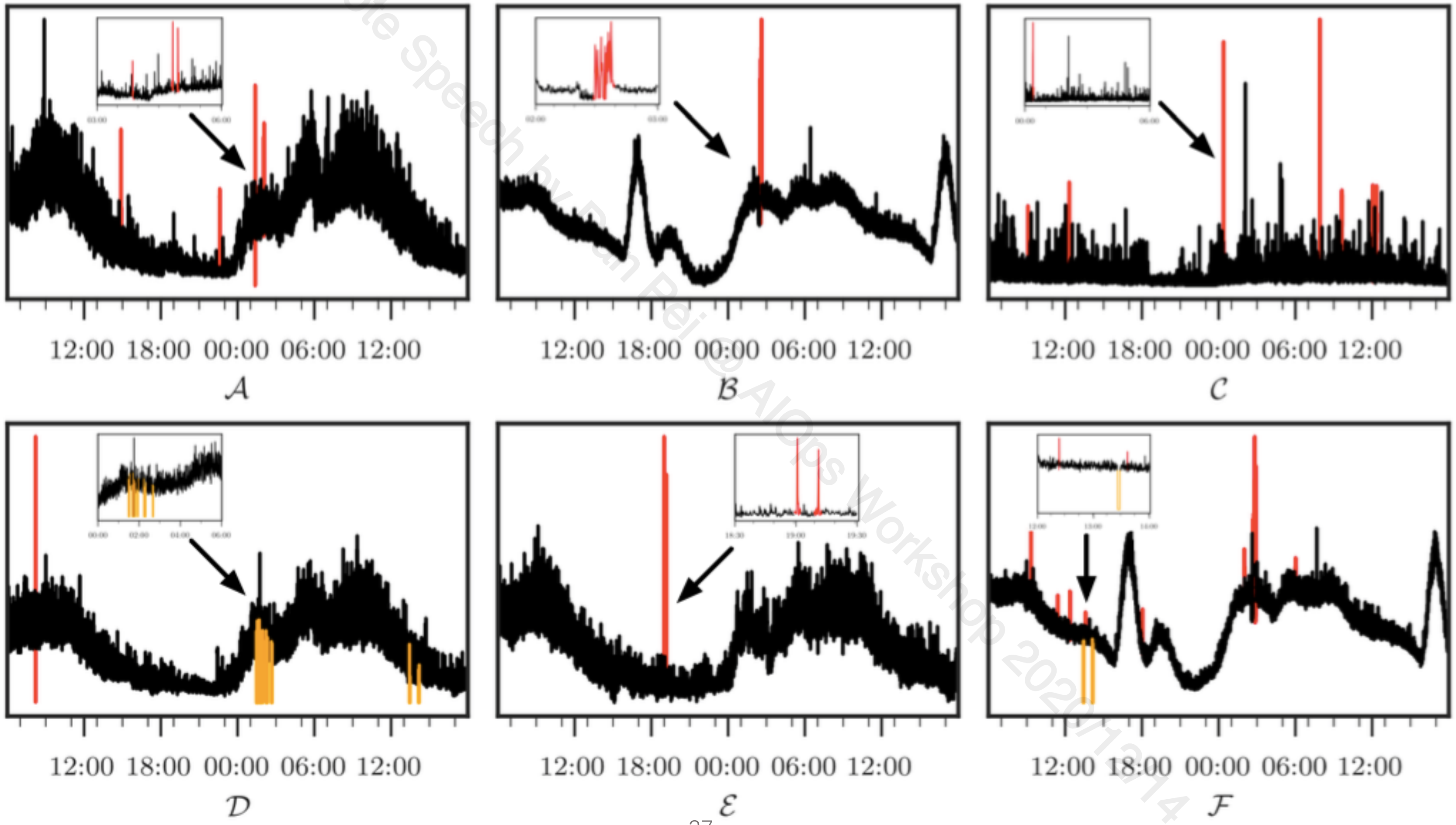
WWW2018

Accuracy of 0.8~0.9, even better than supervised approach.

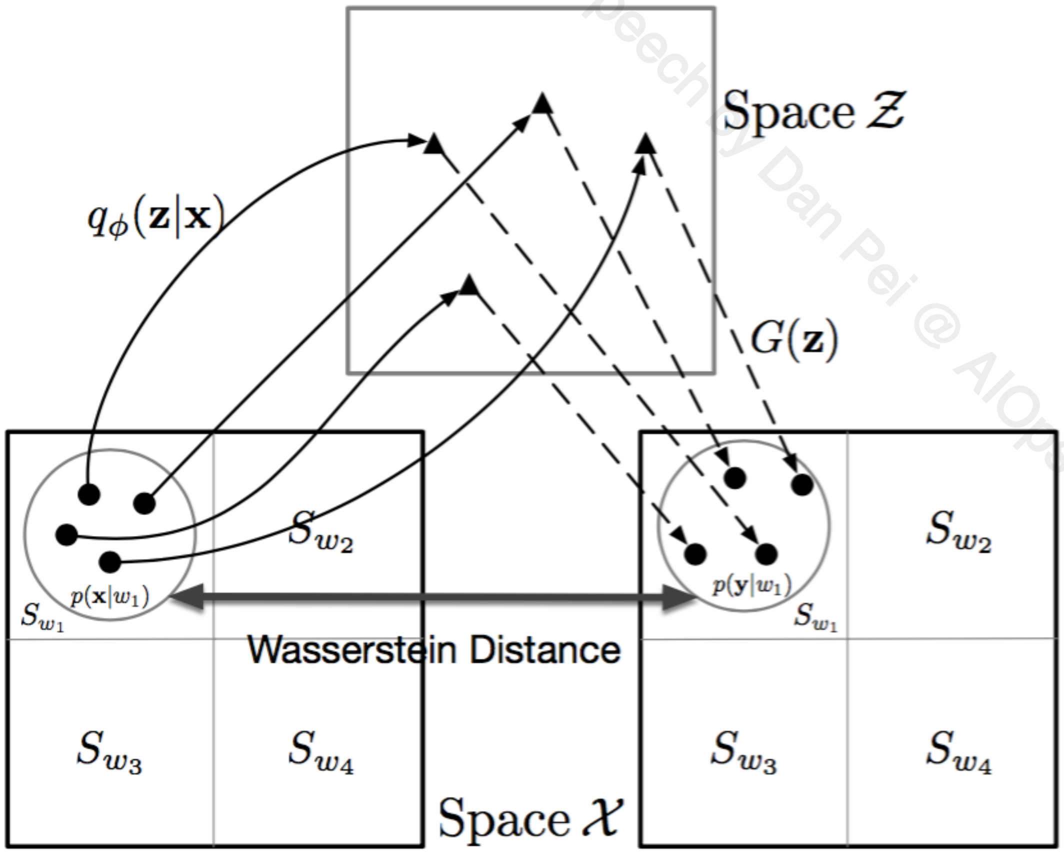




Buzz: Apply Adversarial Training for non-Gaussian noise



Unsupervised Anomaly Detection for Intricate KPIs via Adversarial Training of VAE



INFOCOM 2019

We use two major ideas in Buzz:

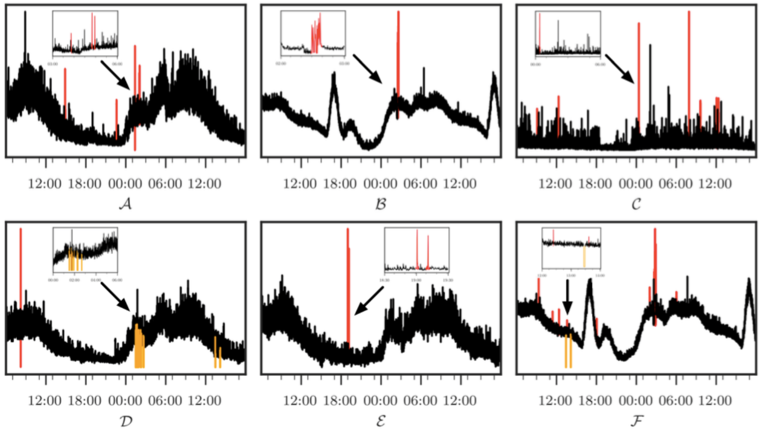
- Wasserstein distance: the distance between the two probability distributions

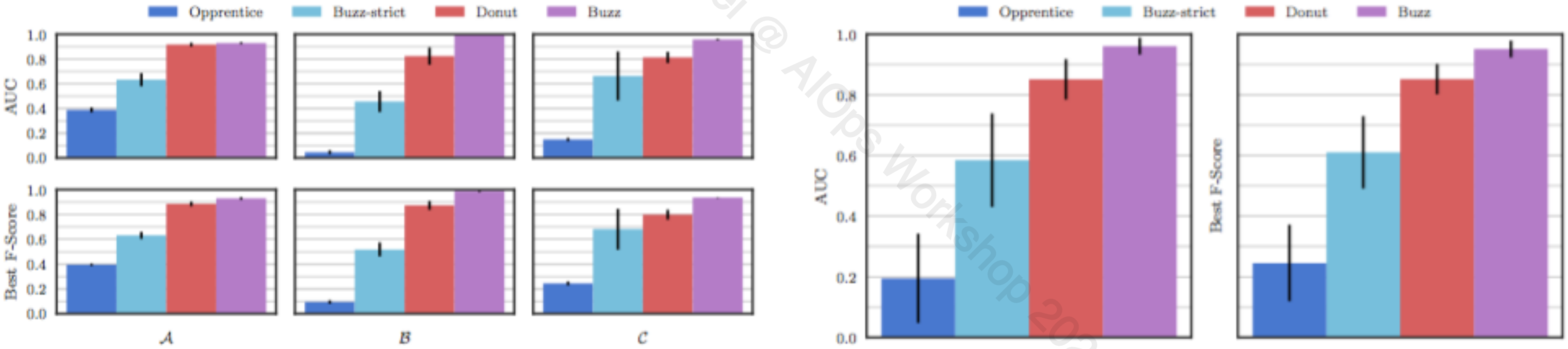
- Partitioning from measure theory.

a powerful and commonly used analysis method for distribution in measure theory.

Experiment Results

Best F-Score outperforms Donut by up to 0.15

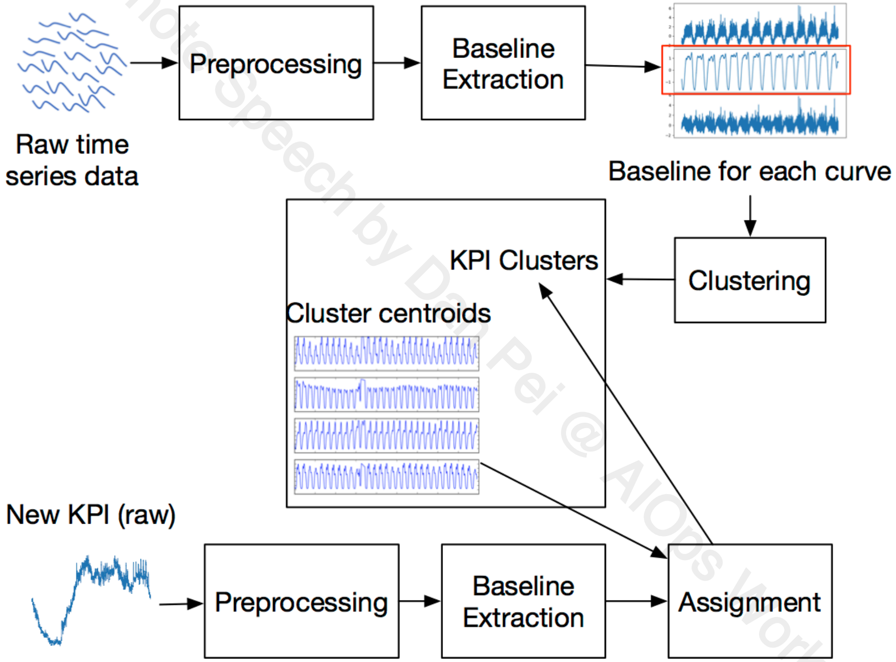




(a) Dateset A, B, C

(b) Average of 11 KPIs

Clustering + Transfer Learning to reduce training overhead



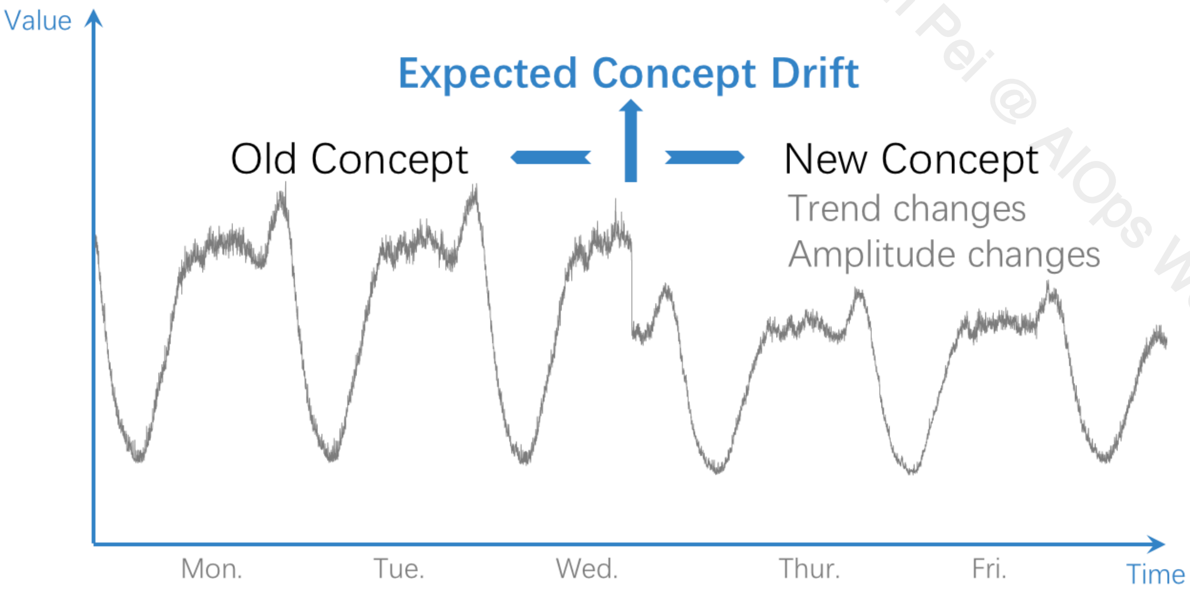
	Original DONUT [WWW2018]	ROCKA+DONUT+KPI-specific threshold
Avg. F-score	0.89	0.88
Total training time (s)	51621	5145

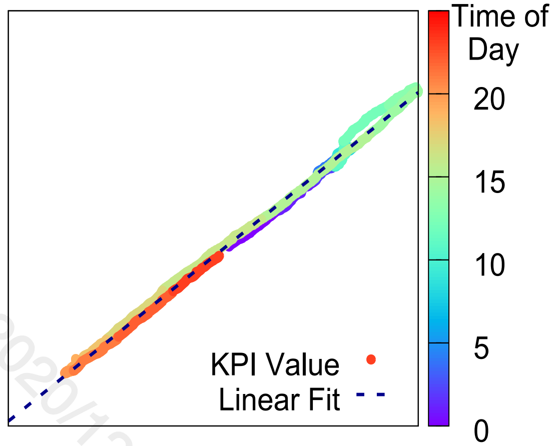
Adapt to Concept Drift

ISSRE 2018 Best Paper

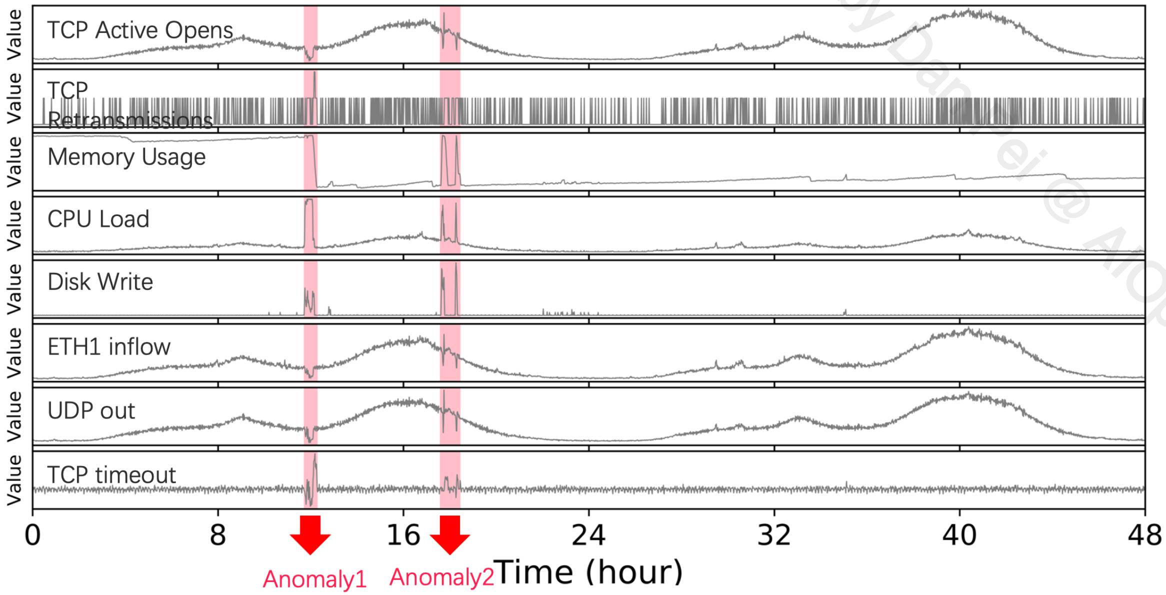
concept drift adaption improve anomaly detection F-score by 203%（**0.225 to 0.681**）

Observation: Old and New Concept Can Be Linearly Fitted

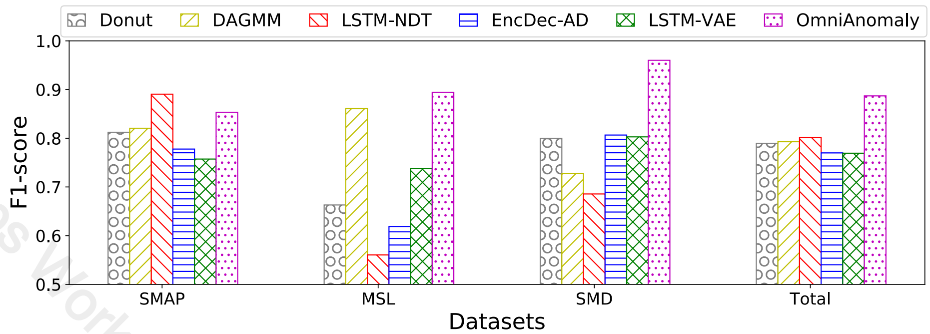




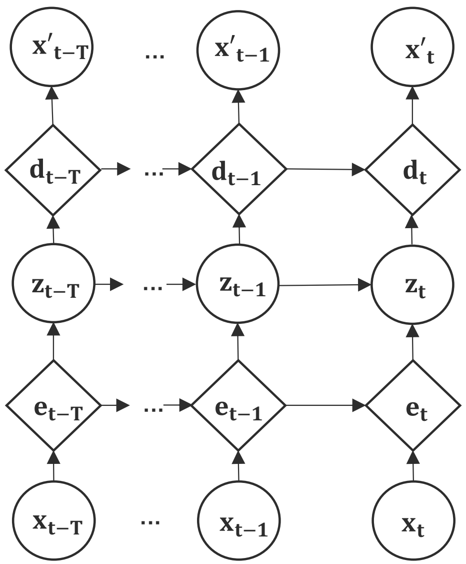
Multivariate Time Series Anomaly Detection with OmniAnomaly (KDD 2019)



F1-best of OmniAnomaly and baselines



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

A good z_t can represent x_t well **regardless of whether** x_t is anomalous or not.

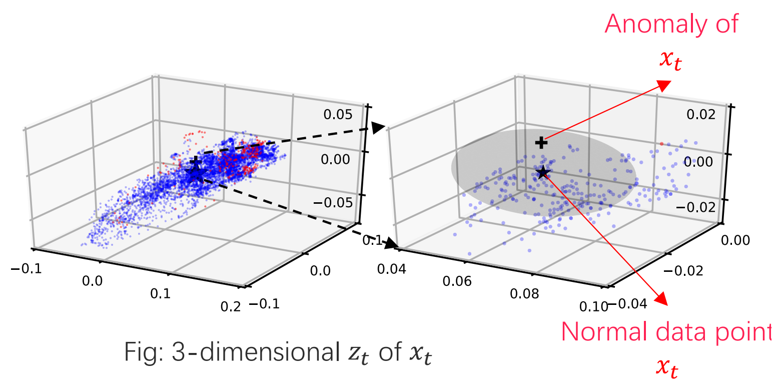


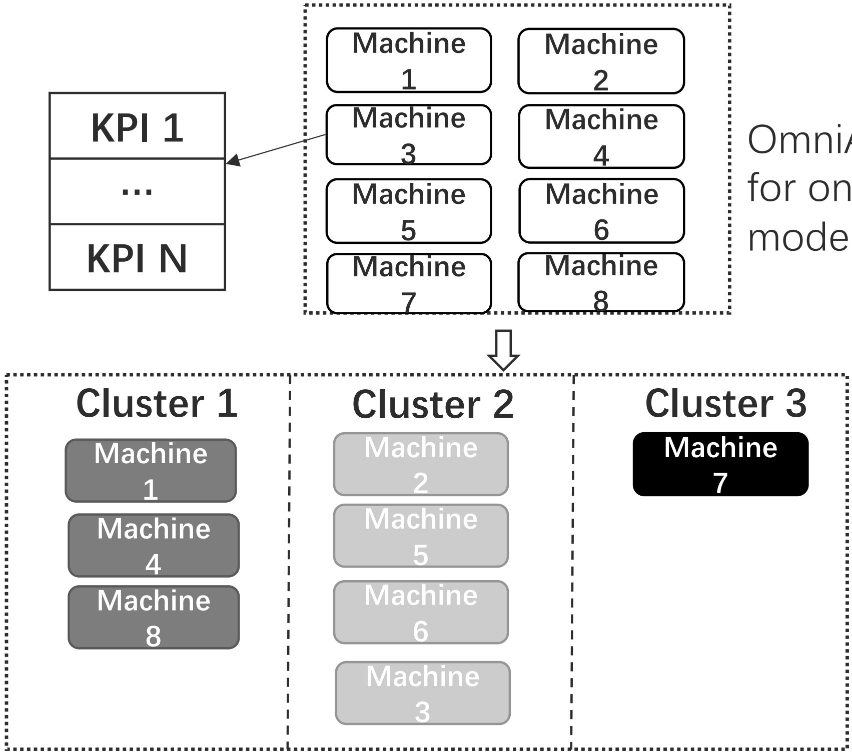
Fig: 3-dimensional z_t of x_t

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

Transfer Learning in Latent Space for MTSAD

training one OmniAnomaly model for each machine costs much time (e.g., 900s for each machine).

Clustering and fine-tuning could greatly reduce the training time with a limited accuracy loss.



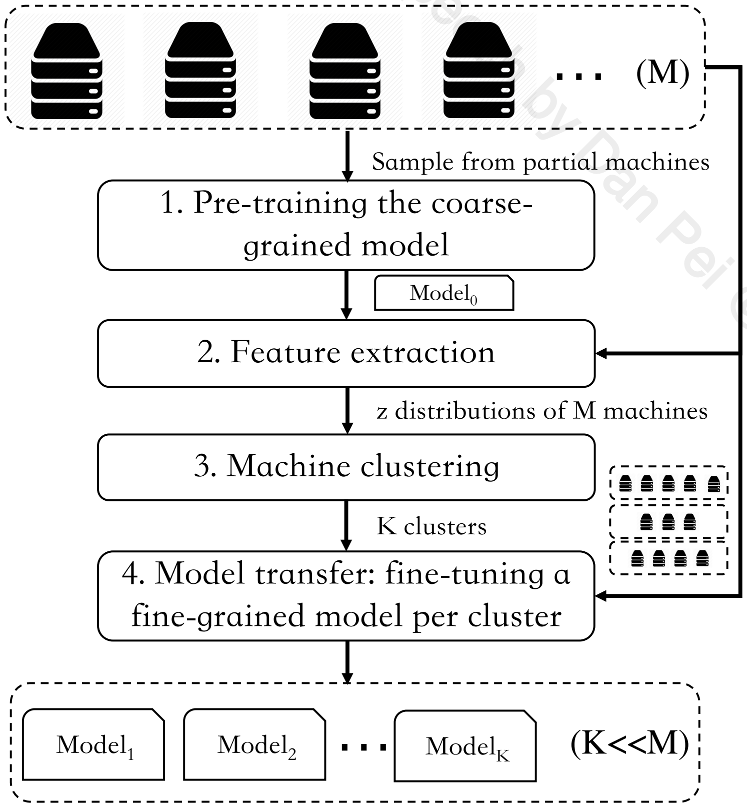
OmniAnomaly: one model for one machine (i.e., 8 models)

one model per cluster (i.e., 3 models)

1. Challenes:
2. The high dimensionality of multivariate time series with noises and anomalies.

- It's challenging to cluster on x or make dimensionality reduction.
- Noises and anomalies may mislead the measurement of distances.

Framework of model training



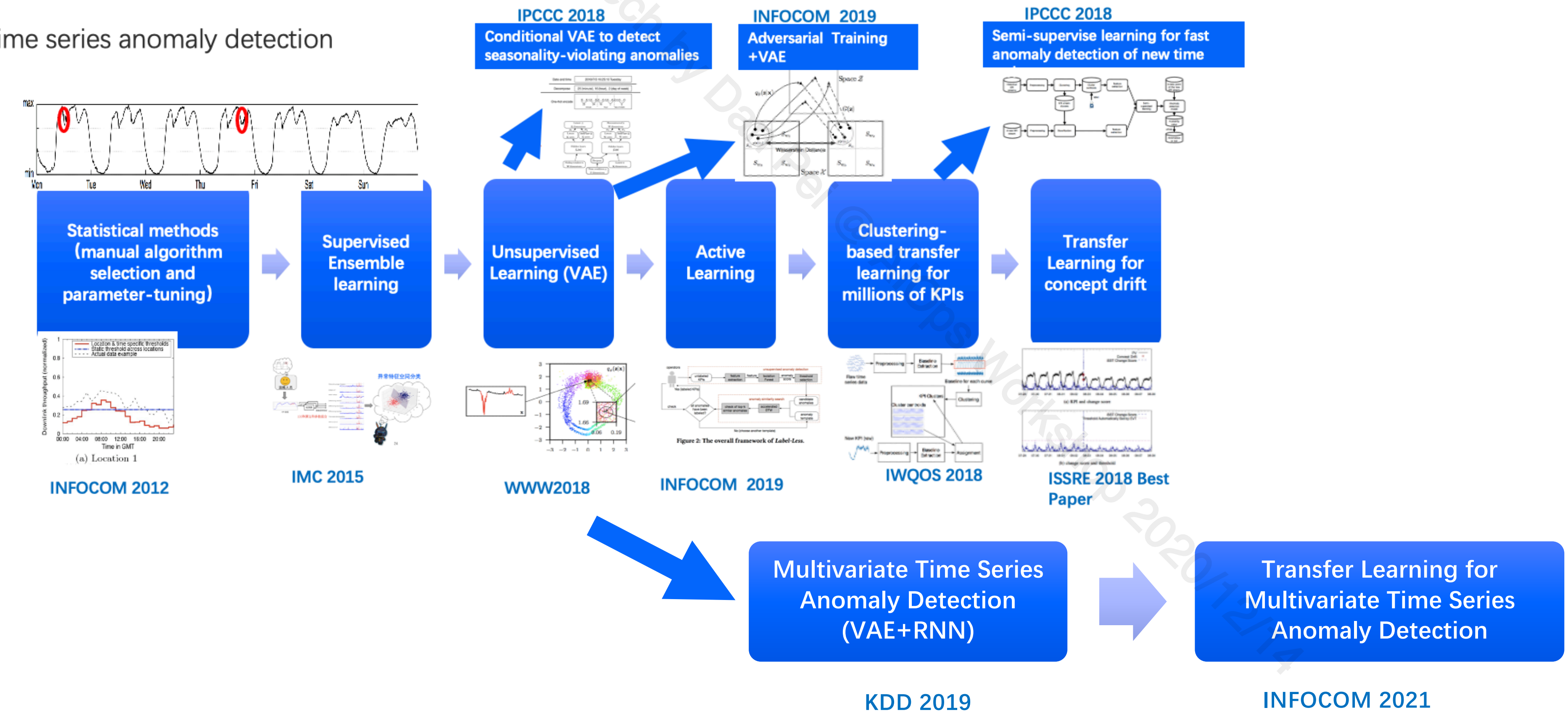
Framework of model training

1. **Sampling strategies in pre-training:**
    - Machine entity sample
    - Time period sample
2. **Feature extraction:**
    - z sample
3. **Clustering on z distribution:**
    - Distance: Wasserstein distance
    - Clustering: Hierarchical agglomerative clustering (HAC) algorithm
4. **Fine-tuning fine-grained models:**
    - Sampling strategies like 1

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.

Time Series Anomaly Detection

Time series anomaly detection



Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Case Study
  - Unsupervised Anomaly Detection in Ops
    - Time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019, KDD2019, INFOCOM2021)
    - *Log anomaly detection (IWQoS 2017, IJCAI 2019, IPCCC2020a, IPCCC2020b, ISSRE2020)*
    - Trace anomaly detection (ISSRE 2020)
    - Zero-day attack detection (INFOCOM2020a)
  - Alert Analysis in Ops
    - INFOCOM2020b, ICSE SEIP 2020, FSE 2020
- Lessons Learned

Hundreds of types of logs in a typical enterprise

NLP techniques are needed to parse and make sense of the log data

Application logs

System logs

- UNIX
- Linux
- Windows
- JVM
- …

Environment Logs

- Power
- A/C
- …

Middleware Logs

- Message Queue
- Tuxedo
- Weblogic
- Tomcat
- Apache
- …

Network Logs

- Switch
- Router
- Load Balancer
- …

Security Device Logs

- Firewall
- IDS
- IPS
- WAF
- …

DB logs

- Oracle
- DB2
- Informix
- SQLServer
- MySQL
- …

```
2018-10-10 20:53:51,194 [JAgentSocketServer.cpp:121] WARN  agent 9995 - Listening Port : 20510↓
2018-10-10 20:53:51,194 [RequestHandlerService.cpp:189] WARN  agent 9995 - RequestHandlerService::handle_input(ACE_HANDLE=38)↓
2018-10-10 20:53:51,195 [ResponseCOUNT.cpp:159] INFO  agent 9995 - IO: Command (1) INITIALISE_PROCESS ↓
2018-10-10 20:53:51,195 [ResponseCOUNT.cpp:302] INFO  agent 9995 - ResponseCOUNT: rc=0↓
2018-10-10 20:53:51,199 [ResponseCOUNT.cpp:159] INFO  agent 9995 - IO: Command (2) INITIALISE_ROOT ↓
2018-10-10 20:53:51,199 [ResponseCOUNT.cpp:302] INFO  agent 9995 - ResponseCOUNT: rc=0↓
2018-10-10 20:53:51,204 [ResponseCOUNT.cpp:159] INFO  agent 9995 - IO: Command (3) INITIALISE_THREAD ↓
```

```
 INFO [WebContainer : 15] - queryForList:IDA_TEMPLATE.LISTDATA_MOST_CLICK↓
 INFO [WebContainer : 8] - queryForList:IDA_NOTICE.LISTDATA_BY_USER↓
com.teradata.ida.auth.dto.SysUserVO@2c3d3e1d↓
[8/10/18 8:29:31:581 CST] 00000032 SystemOut     O  INFO [WebContainer : 1] - queryForList:IDA_TEMPLATE_AUTH.findTemplateByRoleId↓
DEBUG [WebContainer : 7] - 2018-08-10 08:29:32 DEBUG |CsParamSetAction|showAtomsBygid|Start||start=0|limit=25|page=1|fromIndex=0|toInd
 INFO [WebContainer : 7] - queryForList:SEG_BIZ_ATOM_DEF.findAtomByRoleAndShowArea↓
```

```
EXPLANATION:↓
Channel program 'CS_EDI_S' ended abnormally.↓
ACTION:↓
Look at previous error messages for channel program 'CS_EDI_S' in the error↓
files to determine the cause of the failure.↓
----- amqrmrsa.c : 487 ---------------------------------------------------------
08/07/2018 10:14:54 AM - Process(29670.329016) User(mqm) Program(amqrmppa)↓
AMQ9513: Maximum number of channels reached.↓
```

Syslog Messages Under the Type "SIF"

1. Interface ae3, changed state to down
2. Vlan-interface vlan22, changed state to down
3. Interface ae3, changed state to up
4. Vlan-interface vlan22, changed state to up
5. Interface ae1, changed state to down
6. Vlan-interface vlan20, changed state to down
7. Interface ae1, changed state to up
8. Vlan-interface vlan20, changed state to up

Syslog Messages Under the Type “SIF” Before A Failure

1. Interface *, changed state to down
2. Vlan-interface *, changed state to down
3. Interface *, changed state to up
4. Vlan-interface *, changed state to up

A template is a combination of words with high frequency

Common practice for syslog pre-processing:
Extracting templates from syslog messages
Matching syslog messages to templates

Challenges of Log Analysis

Semantic information could be lost if only log template index is used.

Log2Vec

Services can generate new log templates

LogParse

Existing log-based methods cannot detect anomalies accurately.

LogAnomaly

Too few log data for new services

LogTransfer

Semantic-aware log representation

Challenges:

1. Out-of-vocabulary (OOV) words
   - The vocabulary is growing continuously because the service can be upgraded to add new features and fix bugs
2. Domain-specific semantic information
   - Logs contain logs of domain-specific words

Historical logs:

L_1. Interface ae3, changed state to down
L_2. Interface ae3, changed state to up
L_3. Interface ae1, changed status to down
L_4. Interface ae1, changed status to up

Real-time logs:

L_5. Vlan-interface vlan22, changed state to down
L_6. Vlan-interface vlan22, changed state to up

Out-of-vocabulary	Vlan-interface
Relation triples	(Interface, changed, state)
Antonym pairs	(down, up)
Synonym pairs	(state, status)

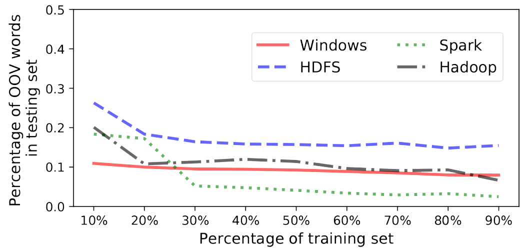
Examples of logs and domain-specific information

Semantic-aware log representation

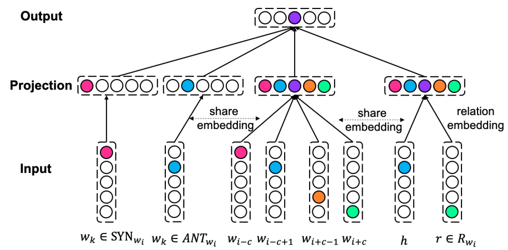
1. Highlighting the challenge of OOV words

2. A Log-specific word embedding method

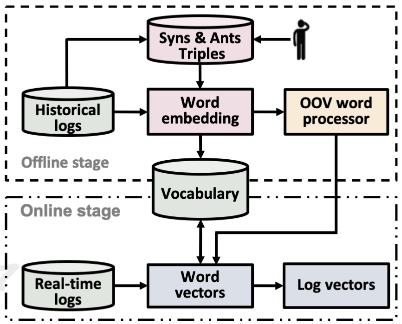
3. Semantic-aware representation framework for online log analysis



Measurements of OOV words



Log-specific word embedding



Framework of Log2Vec

Adaptiveness of Log Parsing

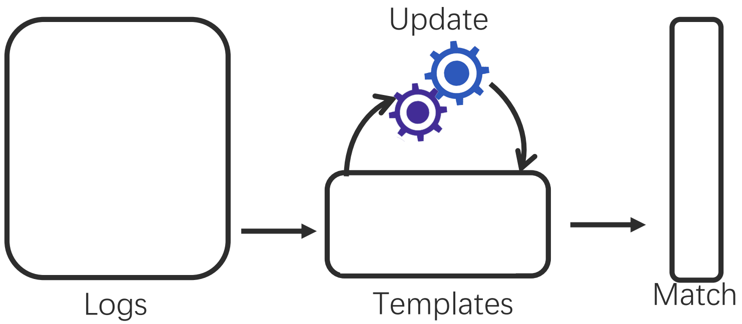
- Goal: match any types of online logs

Historical logs:
L_1. Interface ae3, changed state to down
L_2. Vlan-interface vl22, changed state to down
L_3. Interface ae3, changed state to up
L_4. Interface ae1, changed state to down
Real-time logs:
L_5. Interface ae1, changed state to up
L_6. Vlan-interface vl22, changed state to up

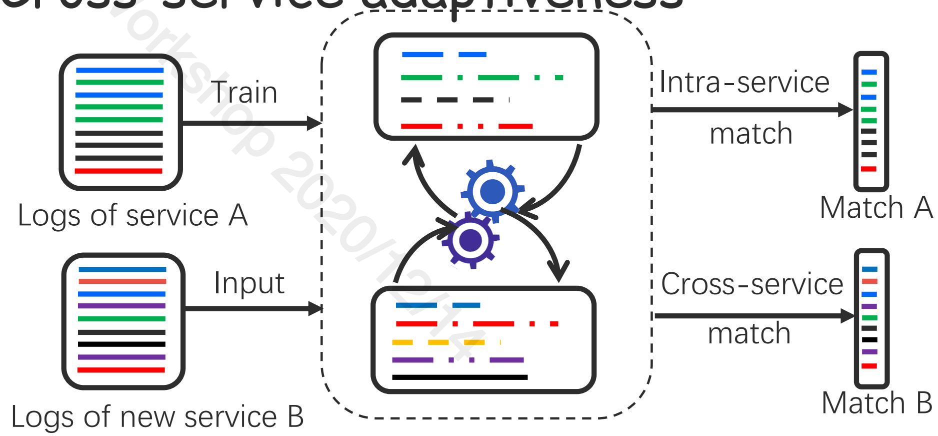
⇨

Templates:
T_1. Interface *, changed state to down
T_2. Vlan-interface *, changed state to down
T_3. Interface *, changed state to up
Template update:
T_4. ? ? ?
Template match:
L_1->T_1 ,ae3 L_2->T_2 ,vl22 L_3->T_3 , ae3
L_4->T_1,ae1 L_5->T_3 ,ae1 L_6-> ? ? ?

- Intra-service adaptiveness



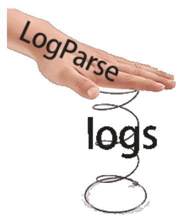
- Cross-service adaptiveness

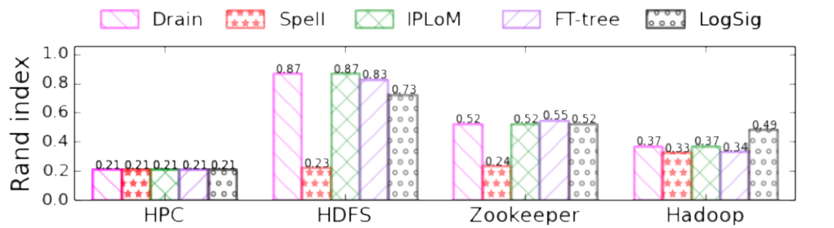


Adaptive Log Parsing Framework

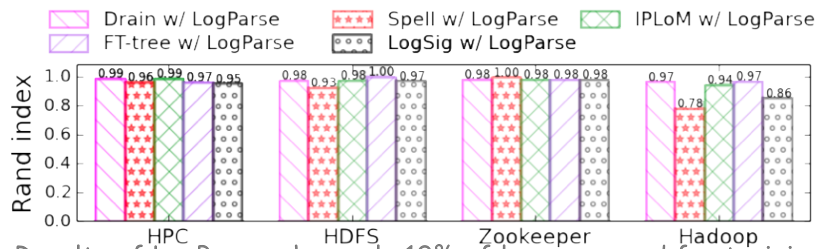
1. LogParse, an adaptive log parsing method
   - Intra-service adaptiveness
   - Cross-service adaptiveness

2. Improve log applications that requires a corresponding template for any given log
   - E.g., log compression





Results of baselines when only 10% of logs are used for training



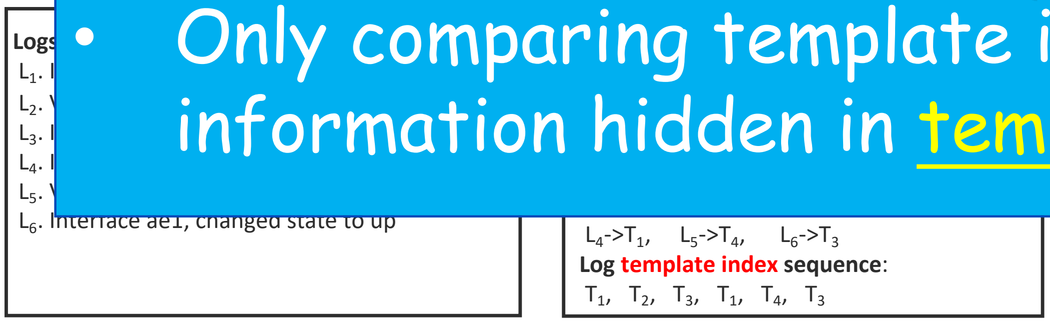
Results of LogParse when only 10% of logs are used for training

Training data (service A)	Testing data (service B)			
	HPC	HDFS	Zookeeper	Hadoop
HPC	-	0.983	0.999	0.923
HDFS	0.982	-	0.993	0.974
ZooKeeper	0.993	1.0	-	0.937
Hadoop	0.983	0.999	0.999	-

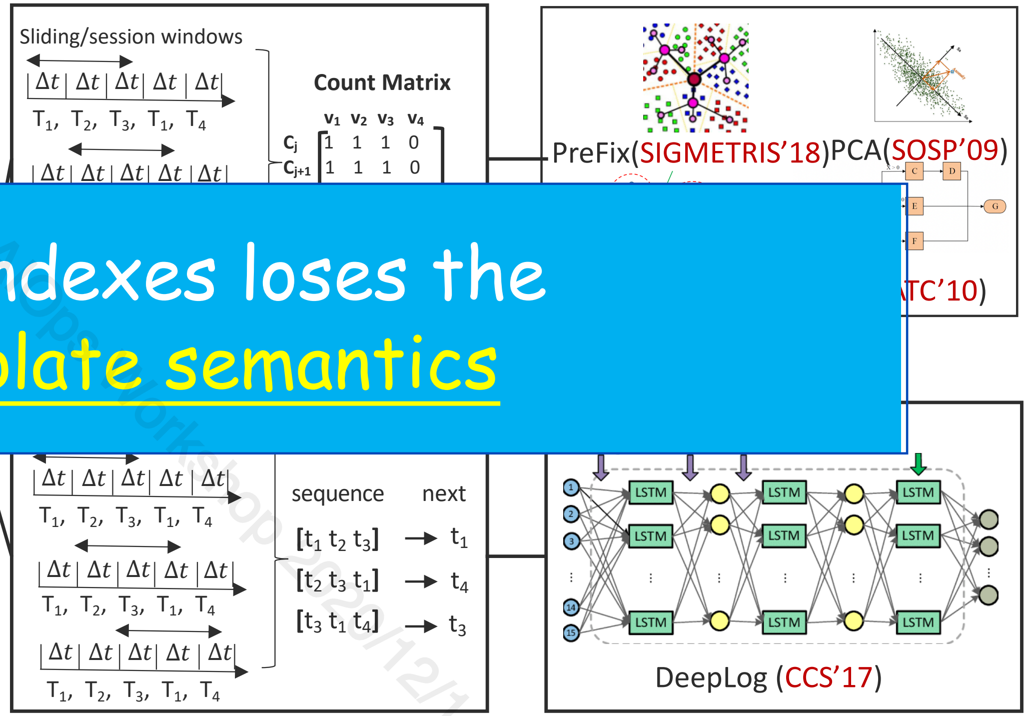
Evaluation on cross-service adaptive

Log-based anomaly detection

- Existing log anomaly detection:
  - Quantitative pattern based methods
  - Sequential pattern based methods



Quantitative anomalies detection methods

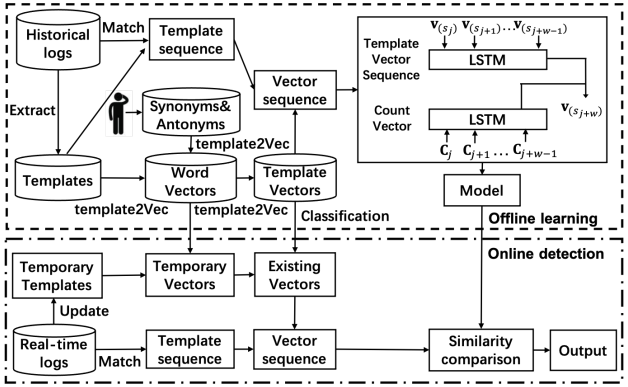


Sequential anomalies detection methods

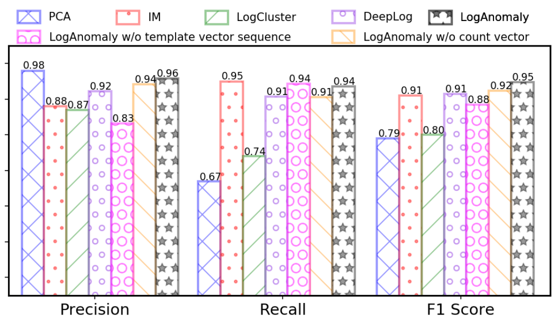
- Only comparing template indexes loses the information hidden in template semantics

LogAnomaly

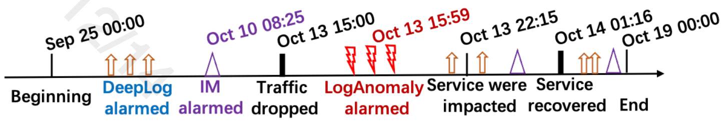
1. LogAnomaly, an accurate anomaly detection framework
2. Template Approximation
   - merging templates of new types automatically
3. Best results on public datasets and real-world switch logs



Design of LogAnomaly



Results on public datasets



Case study on real-world switch logs

LogTranser

Can we transfer anomalous patterns from one software system to another one?

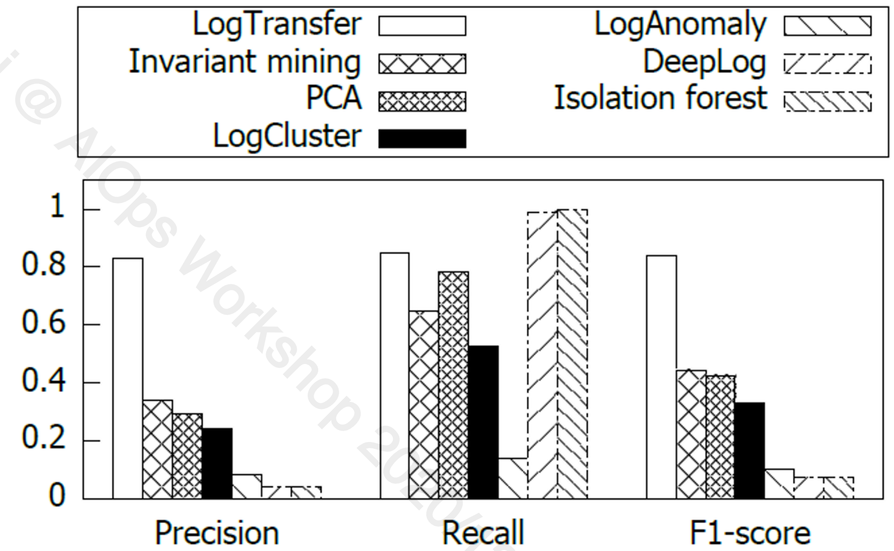
Challenges: syntax differences, noises

Service Type A

[SIF pica_sif]Interface te-1/1/11, changed state to down
[SIF pica_sif]Interface te-1/1/11, changed state to up
[OSPF]Neighbour(rid:, addr:) on vlan20, changed state from Init to ExStart
[OSPF]Neighbour(rid:, addr:) on vlan20, changed state from ExStart to Exchange
[OSPF]Neighbour(rid:, addr:) on vlan20, changed state from Exchange to Loading
[OSPF]Neighbour(rid:, addr:) on vlan20, changed state from Loading to Full
[OSPF]Neighbour(rid:, addr:) on vlan20, changed state from Full to Down
[SIF]Vlan-interface vlan20, changed state to down
[SIF]Vlan-interface vlan20, changed state to up

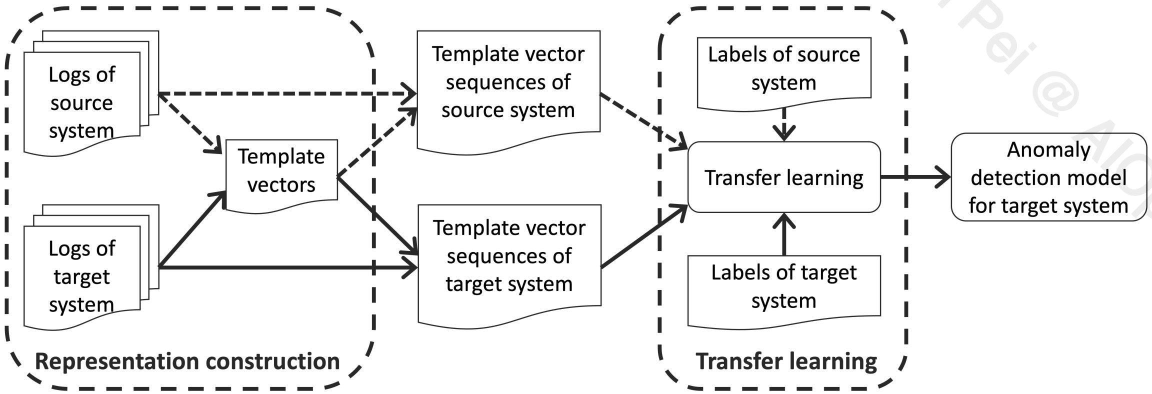
%%10IFNET/3/LINK_UPDOWN(I): GigabitEthernet1/0/10 link status is DOWN.
%%10IFNET/3/LINK_UPDOWN(I): GigabitEthernet1/0/10 link status is UP.
%%10OSPF/3/OSPF_NBR_CHG(I): OSPF 1 Neighbor (Vlan-interface20) from Loading to Full.
%%10OSPF/3/OSPF_NBR_CHG(I): OSPF 1 Neighbor (Vlan-interface20) from Full to ExStart.
%%10OSPF/3/OSPF_NBR_CHG(I): OSPF 1 Neighbor (Vlan-interface20) from Full to Down.
%%10OSPF/3/OSPF_NBR_CHG(I): OSPF 1 Neighbor (Vlan-interface20) from Full to Init.
%%10IFNET/3/LINK_UPDOWN(I): Vlan-interface20 link status is DOWN.
%%10IFNET/3/LINK_UPDOWN(I): Vlan-interface20 link status is UP.

Service Type B

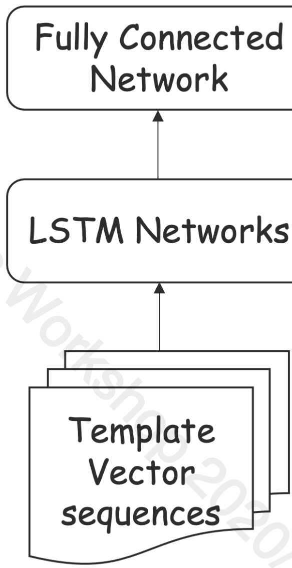


Switch log A -> Switch log B accuracy comparison

- Transfer learning



Fully Connected Network for anomaly detection (Shared)



LSTM networks to extract the pattern of log sequences (fine-tuning in target system)

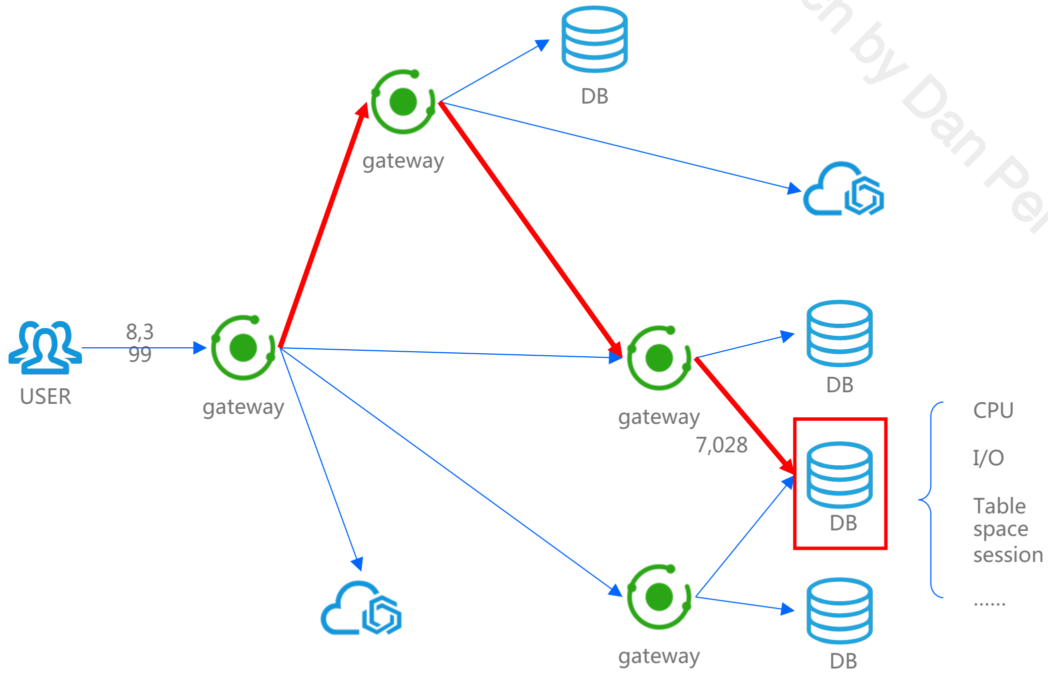
Separately-learned template vector sequences with syntactic and semantic info.

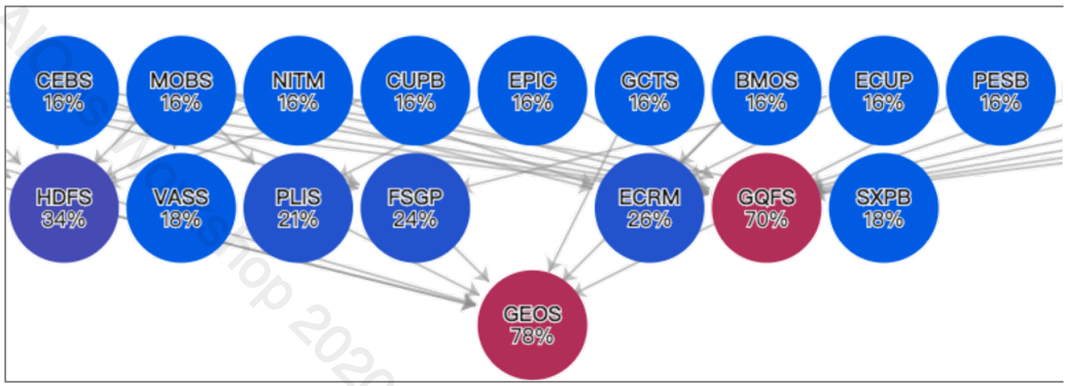
Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Case Study
  - Unsupervised Anomaly Detection in Ops
    - *Time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019, KDD2019, INFOCOM2021)*
    - Log anomaly detection (IWQoS 2017, IJCAI 2019, IPCCC2020a, IPCCC2020b, ISSRE2020)
    - *Trace anomaly detection (ISSRE 2020)*
    - Zero-day attack detection (INFOCOM2020a)
  - Alert Analysis in Ops
    - INFOCOM2020b, ICSE SEIP 2020, FSE 2020
- Lessons Learned

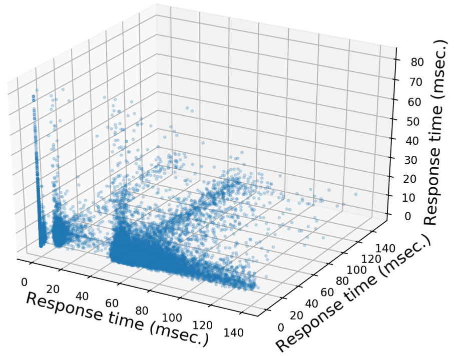
Software Module Invocation Traces

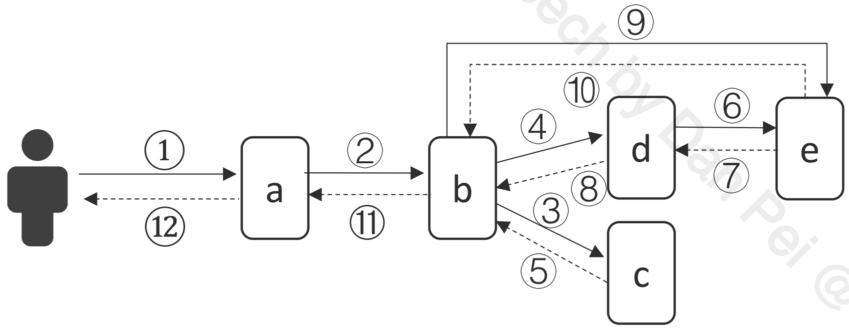
- Invocation trace: 10s~100s of module-to-module invocations for a unique transaction
  - One module failure can manifest itself cross-invocation and cross-transaction





This mandates that response times and call paths must be unified





For a microservice, its response time is determined by both itself and its call path

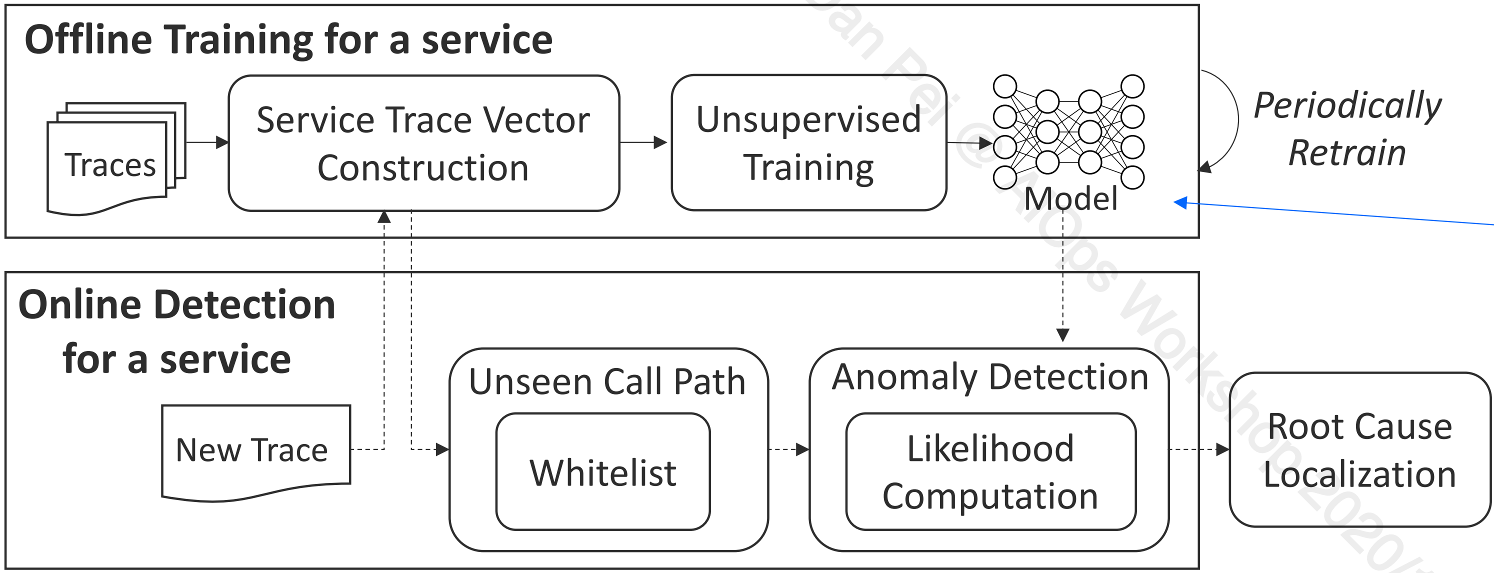
Microservice s	Call path of microservice s (s, call path)	Response time of (s, call path) (msec)
a	(a, (start→a))	222
b	(b, (start→a, a→b))	209
c	(c, (start→a, a→b, b→c))	4
d	(d, (start→a, a→b, b→c, b→d))	44
e	**(e, (start→a, a→b, b→c, b→d, d→e))**	**28**
e	**(e, (start→a, a→b, b→c, b→d, d→e, b→e))**	**67**

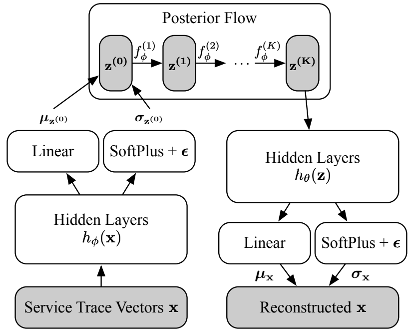
Microservice *e* is invoked twice, with different response time

Design of TraceAnomaly

TABLE III: Online evaluation results of different approaches on four large online services which contain hundreds of microservices, whose statistics are shown in Table I.

	Service-1		Service-2		Service-3		Service-4		Overall (Union of 4 services)	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Hard-coded Rule	0.910	0.800	0.920	0.792	0.911	0.812	0.930	0.800	0.910	0.804
WFG-based [5]	0.020	0.500	0.012	0.323	0.050	0.410	0.032	0.300	0.031	0.386
DeepLog* [8]	0.270	0.680	0.241	0.560	0.320	0.643	0.302	0.601	0.290	0.628
CPD-based [7]	0.52	0.063	0.43	0.090	0.57	0.110	0.64	0.072	0.531	0.081
CFG-based [6]	0.170	0.610	0.250	0.570	0.102	0.503	0.180	0.630	0.164	0.562
TraceAnomaly	**0.980**	**1.000**	**0.982**	**1.000**	**0.981**	**1.000**	**0.973**	**1.000**	**0.981**	**1.000**





Service trace vector construction

- Unify response time and call paths of traces in an interpretable way
  - Encode the response time and call paths of a trace in a service into a STV (Service Trace Vector)

Traces

Microservice s	Call path of microservice s (s, call path)	STV
a	(a, (start→a))	rt
b	(b, (start→a, a→b))	rt
c	(c, (start→a, a→b, b→c))	rt
d	(d, (start→a, a→b, b→c, b→d))	rt
e	(e, (start→a, a→b, b→c, b→d, d→e))	rt
e	(e, (start→a, a→b, b→c, b→d, d→e, b→e))	rt
…	…	…

The dimension ID of the STV corresponds to the call path of microservice *s*

The value of the dimension corresponds to the response time of microservice *s*

Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Case Study
  - Unsupervised Anomaly Detection in Ops
    - *Time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019, KDD2019, INFOCOM2021)*
    - Log anomaly detection (IWQoS 2017, IJCAI 2019, IPCCC2020a, IPCCC2020b, ISSRE2020)
    - Trace anomaly detection (ISSRE 2020)
    - *Zero-day attack detection (INFOCOM2020a)*
  - Alert Analysis in Ops
    - INFOCOM2020b, ICSE SEIP 2020, FSE 2020
- Lessons Learned

Detecting Zero-day Attacks

- WAF detects those **known** attacks effectively.
  - filter out **known** attacks
- **ZeroWall** detects **unknown** attacks **ignored by WAF rules**.
  - report **new attack patterns** to operators and security engineers to **update** **WAF rules**.

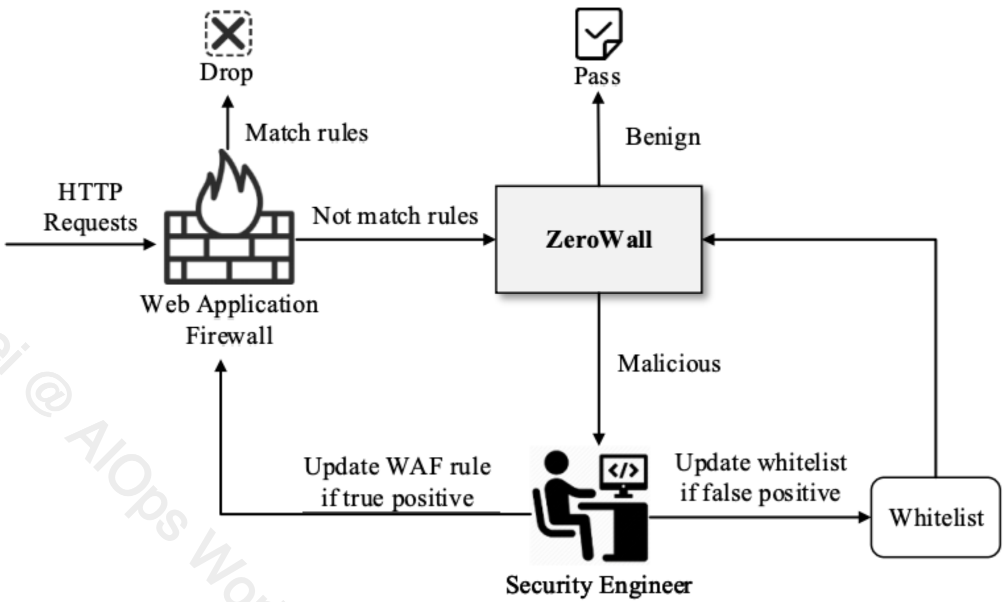
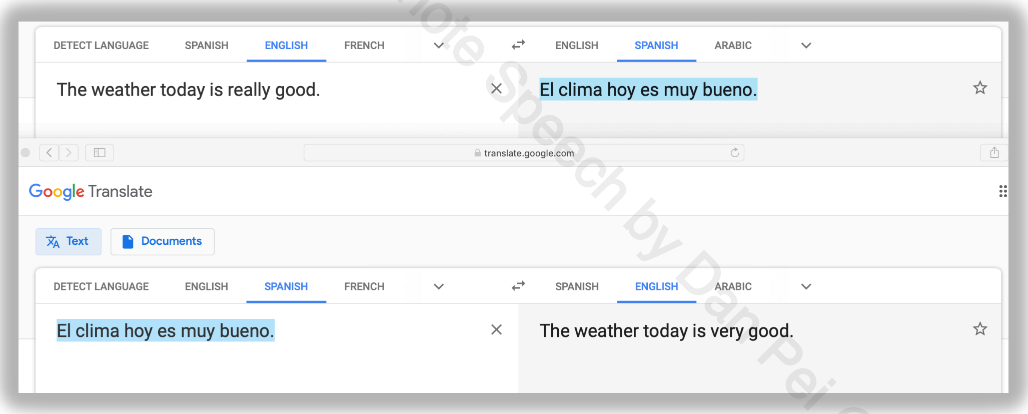


Figure 1: The workflow of *ZeroWall*.

Self-Translate Machine

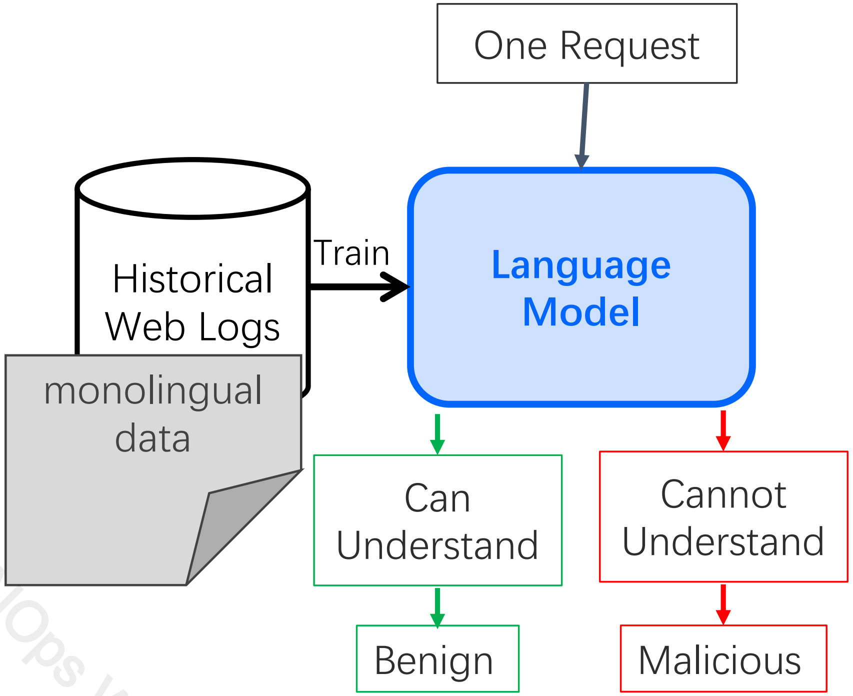


Self-translation works **well** for **normal** sentences

Output **deviates** significantly from the input, when the input is a sentence **not previously seen** in the training dataset of the self-translation models.

Idea

- HTTP request is a **string following HTTP**, and we can consider an HTTP request as one **sentence** in the ***HTTP request language***.
- **Most** requests are **benign**, and **malicious** requests are **rare**.
- Thus, we train a kind of **language model** based on historical logs, to **learn this language** from **benign requests**.



Deployed in the wild

Over **1.4** billion requests

Captured **28** different types of zero-day attacks (**10K** of zero-day attack requests)

Low overhead

Summary: Unsupervised Anomaly Detection in Ops

- Common Idea: somehow capture the "normal" patterns in the historical data, then any new points that "deviate" from the normal patterns are considered "anomalous".

- Domain specific feature engineering (time series, log, trace, etc.)

- Sometimes have to assume non-Gaussian distributions in x-space or z-space
  - GAN
  - Flows in Z-space
- Temporal dependency can be captured in x-space or z-space

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

- Clustering + transfer learning in x-space or z-space help reduce training overhead with little accuracy loss.

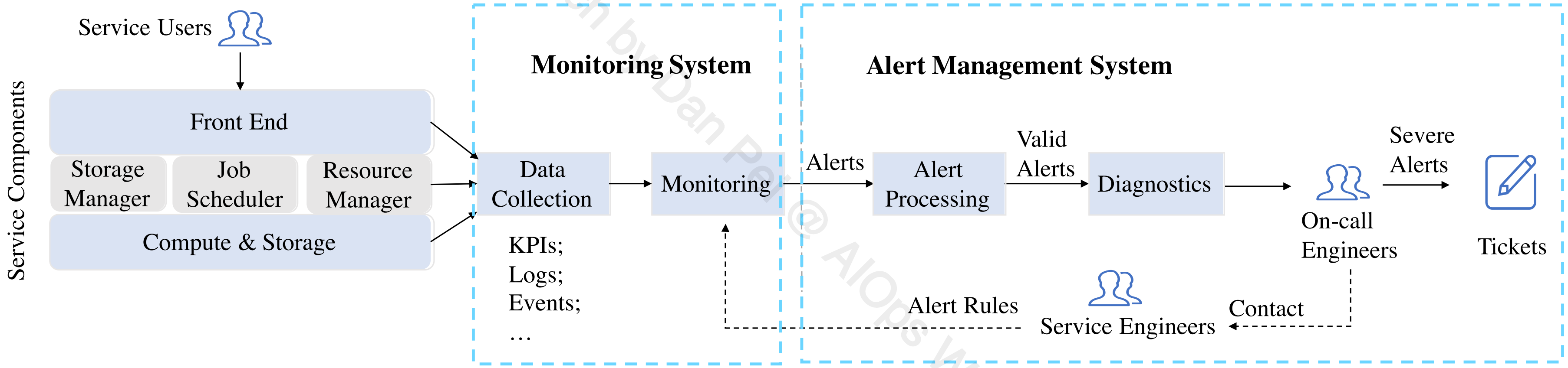
- Various distance metrics: e.g. Wasserstein distance

- Periodic re-training + whitelisting (active learning) for small changes

- Transfer learning for concept change.

Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Case Study
  - Unsupervised Anomaly Detection in Ops
    - Time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019, KDD2019, INFOCOM2021)
    - Log anomaly detection (IWQoS 2017, IJCAI 2019, IPCCC2020a, IPCCC2020b, ISSRE2020)
    - Trace anomaly detection (ISSRE 2020)
    - Zero-day attack detection (INFOCOM2020a)
  - *Alert Analysis in Ops*
    - INFOCOM2020b, ICSE SEIP 2020, FSE 2020
- Lessons Learned



Monitoring data

Time	Severity	Type
2019-02-20 10:04:32	P2-error	Memory

AppName	Server	Close Time
E-BANK	IP(*.*.*.*)	2019-02-20 10:19:45

Content
Current memory utilization is 79% (Threshold is 60%).
Resolution Record
Contact the service engineers responsible for E-BANK and get a reply that there is no effect on business, then close the alert.

Alert rules

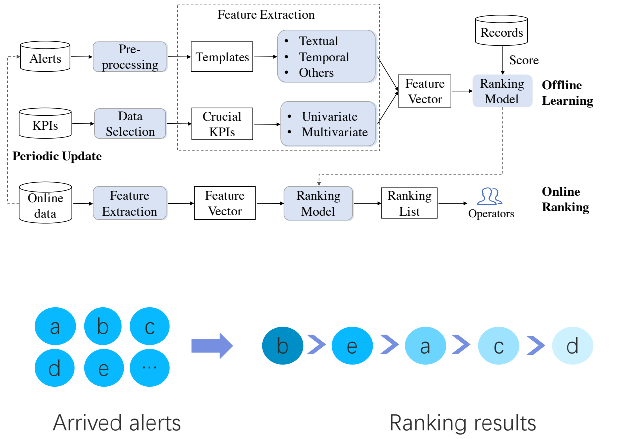
Summary

How to rank alert accurately and adaptatively, so as to ensure accurate and timely failure discovery

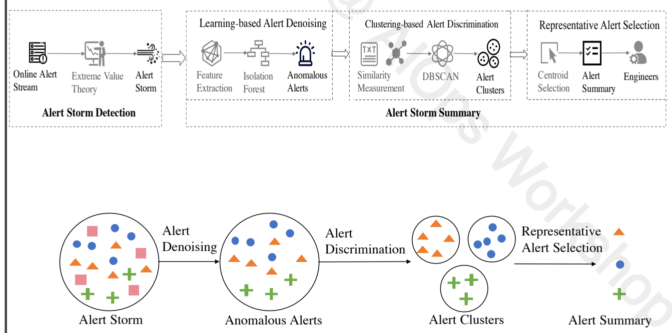
How to handle alert storm effectively, so as to assist failure diagnosis

How to predict incident with alerts, so as to take proactive actions to prevent incidents

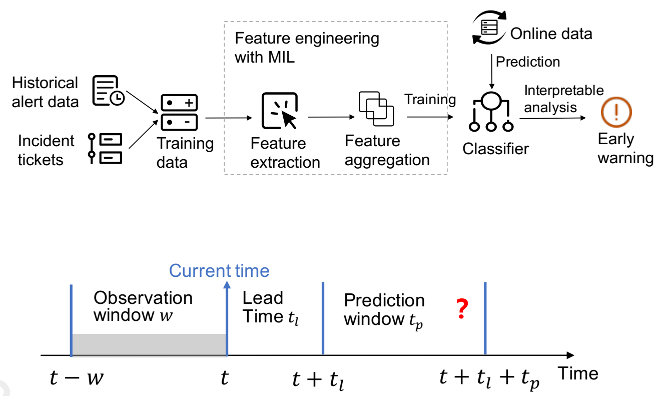
AlertRank



Alert Summary

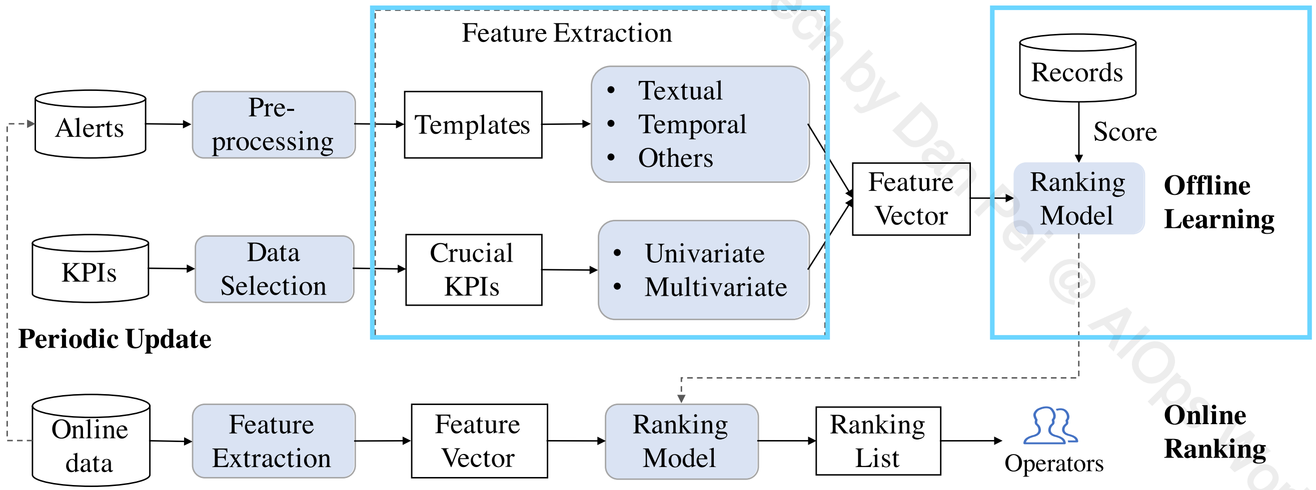


eWarn



Automatically and Adaptively Identifying Severe Alerts for Online Service Systems, INFOCOM 2020
Understanding and Handling Alert Storm for Online Service Systems, ICSE SEIP 2020
Real-Time Incident Prediction for Online Service Systems, ESEC/FSE 2020

Alert Rank



Datasets	A			B			C		
Methods	P	R	F1	P	R	F1	P	R	F1
AlertRank	**0.85**	**0.93**	**0.89**	**0.82**	**0.90**	**0.86**	**0.93**	**0.92**	**0.93**
Rule-based	0.43	0.68	0.53	0.47	0.70	0.56	0.41	0.74	0.53
Bug-KNN	0.72	0.76	0.74	0.79	0.62	0.70	0.80	0.53	0.64

Datasets	A			B			C		
Methods	P	R	F1	P	R	F1	P	R	F1
AlertRank	**0.85**	**0.93**	**0.89**	**0.82**	**0.90**	**0.86**	**0.93**	**0.92**	**0.93**
Alert Only	0.82	0.79	0.80	0.75	0.80	0.77	0.67	0.77	0.72
KPI Only	0.42	0.40	0.41	0.32	0.39	0.35	0.36	0.31	0.33

Core idea:
- Multi-feature fusion: alert features and KPI features
- Learning to rank problem

- Our model benefits from the ensemble features extracted from multiple data sources
- Alert features are more powerful than KPI features.

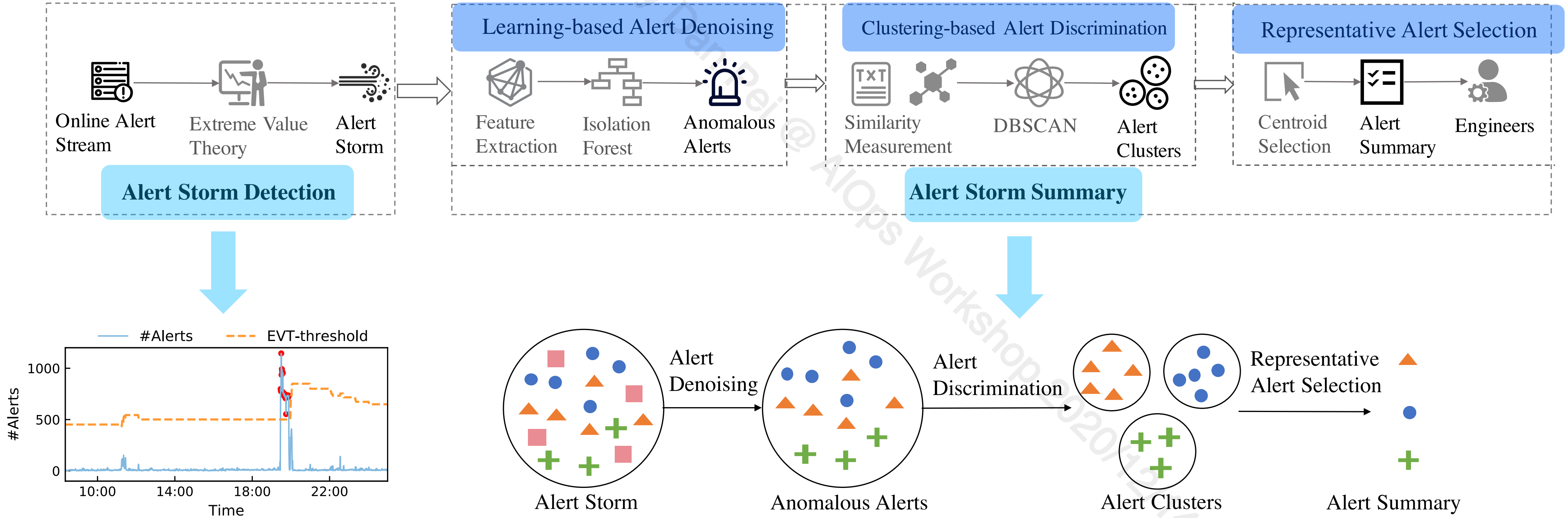
AlertSummary

- Anomaly detection problem
- Features: alert attributes
- Isolation forest

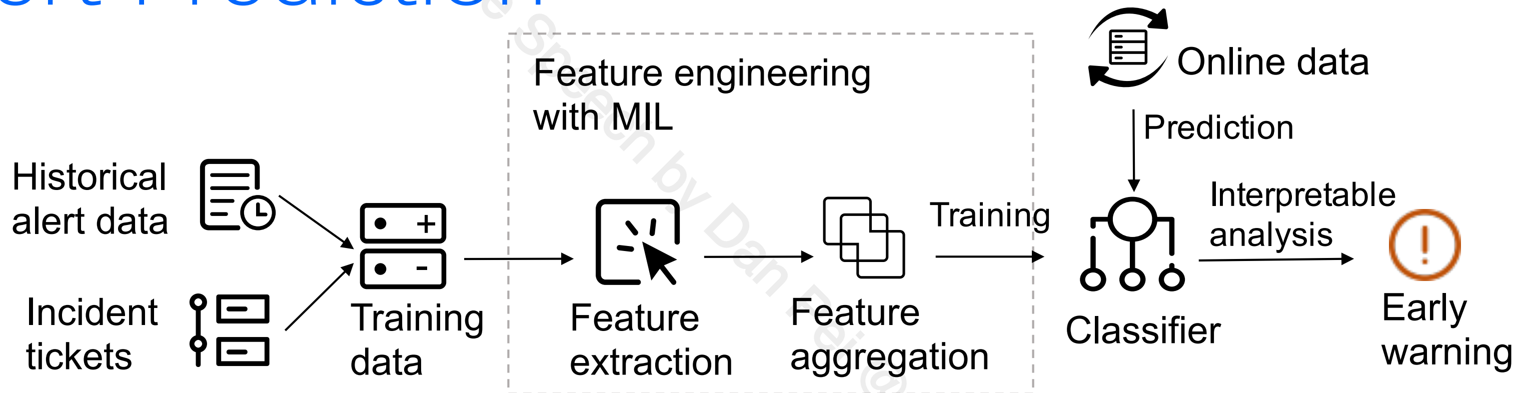
- Similarity measurement
  - Textual similarity: Jaccard distance
  - Topological similarity: graph path

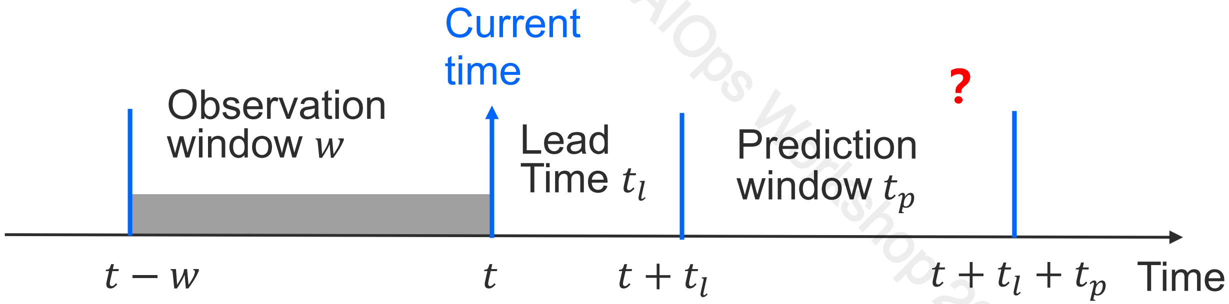
Datasets	Raw	Severity	Denoising	**Summary**
A	0%	88.7%	6.9%	**98.8%**
B	0%	85.6%	5.1%	**98.2%**
C	0%	84.1%	8.4%	**99.1%**

$$\text{centroid} = \underset{i \in \text{cluster}}{\arg\min} \frac{1}{n} \sum\nolimits_{j=1}^{n} \text{similarity}(i, j)$$



Alert Prediction





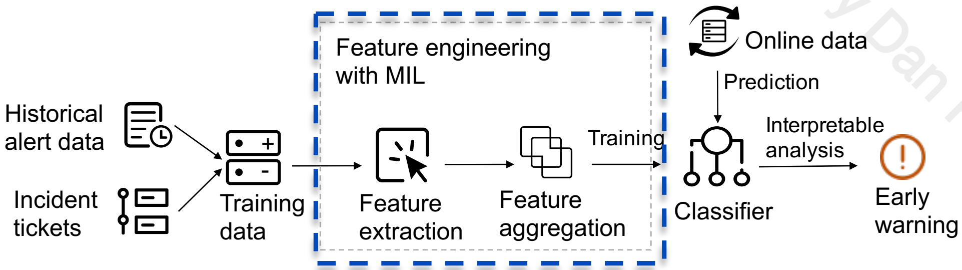
Time window classification

- Positive: early warning of an incident
- Negative: no incident

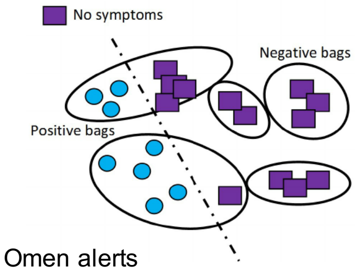
Lead time: the minimum time interval that engineers need to react to an early warning

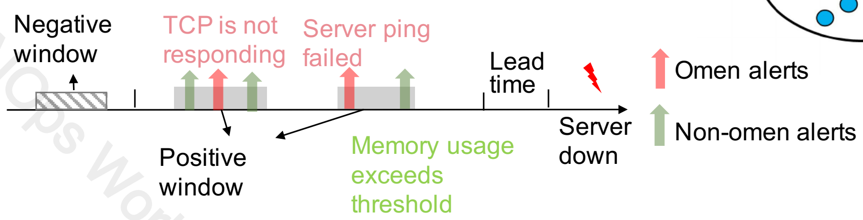
Feature Engineering

Approach	*eWarn*			AirAlert			TF-IDF-LSTM			FP-Growth		
System	P	R	F	P	R	F	P	R	F	P	R	F
S1	0.86	0.82	**0.84**	0.46	0.82	0.59	0.93	0.73	0.82	0.08	0.05	0.06
S2	0.86	0.97	**0.91**	0.81	0.94	0.87	0.80	0.88	0.84	0.25	0.22	0.23
S3	0.61	0.83	**0.70**	0.41	0.24	0.31	0.23	0.76	0.35	0.05	0.09	0.07
S4	0.92	0.84	**0.88**	0.34	0.81	0.48	0.58	0.39	0.46	0.16	0.27	0.20
S5	0.75	0.86	**0.80**	0.34	0.29	0.32	0.14	0.31	0.19	0.12	0.25	0.17
S6	0.96	1.00	**0.98**	0.21	1.00	0.35	0.91	1.00	0.95	1.00	0.05	0.09
S7	0.73	0.71	**0.72**	0.65	0.53	0.59	0.67	0.73	0.69	0.00	0.00	0.00
S8	0.56	0.92	**0.69**	0.22	1.00	0.36	0.17	1.00	0.30	0.13	0.10	0.11
S9	0.92	0.98	**0.95**	0.53	1.00	0.69	0.92	0.98	0.95	0.03	0.02	0.02
S10	0.70	0.79	**0.76**	0.55	0.86	0.67	0.52	0.90	0.66	0.53	0.06	0.11
S11	0.81	0.69	**0.75**	0.28	0.57	0.37	0.25	0.52	0.34	0.01	0.06	0.01
Average	–	–	**0.82**	–	–	0.51	–	–	0.60	–	–	0.10



Multi-instance learning





1 Feature extraction

- Textual features: Topic model
- Statistical features: count, window time, Inter-arrival time, etc.

Clustering-based feature aggregation

- Omen alerts: assign larger weight
- Non-omen alerts: assign small weight, to bypass noisy alerts

Alert Analysis: Lessons learned

1. Ranking instead of manual rules
2. Features from multiple data sources instead of alerts alone
3. Divide and Conquer: e.g. Storm detection, Storm Clustering, Representative Alert Selection
4. Problem Formulation important: (e.g. MIL in eWarn)

Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Case Study
  - Unsupervised Anomaly Detection in Ops
    - Time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019, KDD2019, INFOCOM2021)
    - Log anomaly detection (IWQoS 2017, IJCAI 2019, IPCCC2020a, IPCCC2020b, ISSRE2020)
    - Trace anomaly detection (ISSRE 2020)
    - Zero-day attack detection (INFOCOM2020a)
  - Alert Analysis in Ops
    - INFOCOM2020b, ICSE SEIP 2020, FSE 2020
- *Lessons Learned*

Pitfalls: use general ML algorithms as Blackbox to tackle Ops challenges

Failure Discovery → Failure Mitigation → Failure Repair → Failure Avoidance

↑

Huge Gap

General Machine Learning Algorithms

ARIMA, Time Series Decomposition, Holt-Winters, CUSUM, SST,DiD,DBSCAN, Pearson Correlation, J-Measure, Two-sample test, Apriori, FP-Growth, K-medoids, CLARIONS, Granger Causality, Logistic Regression, Correlation analysis (event-event, event-time series, time series-time series) , hierarchical clustering, Decision tree, Random forest, support vector machine, Monte Carlo Tree search, Marcovian Chain, multi-instance learning, transfer learning, CNN, RNN ,VAE, GAN, NLP

The capability boundary of current AI technologies

AI is good at solving problems that satisfy the following five conditions simultaneously:

(1) With abundant data or knowledge

(2) With deterministic Information

(3) With complete Information

(4) Well-defined

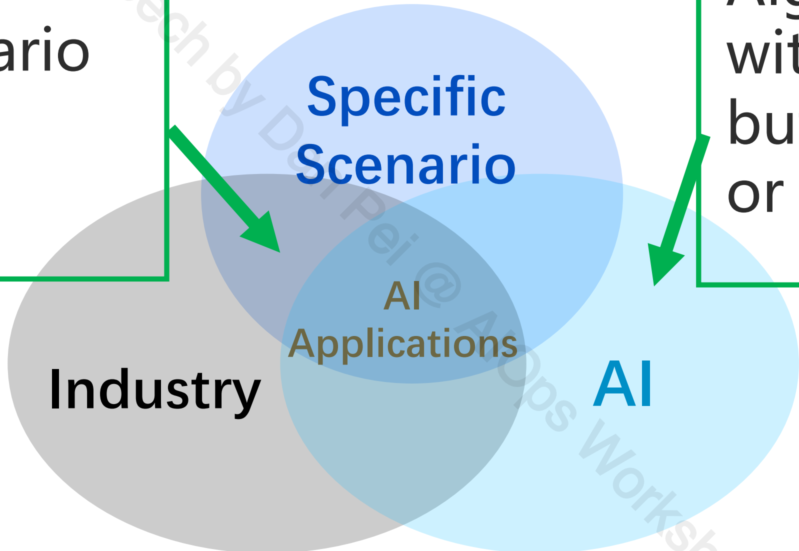
(5) Single-domain or limited-domain

——*CAS Fellow, Prof Bo Zhang*

Why success only in specific application scenario in specific area in specific industry?

Industry people familiar with scenario and industry

Algorithm people familiar with general AI,
but not specific industry or specific scenario



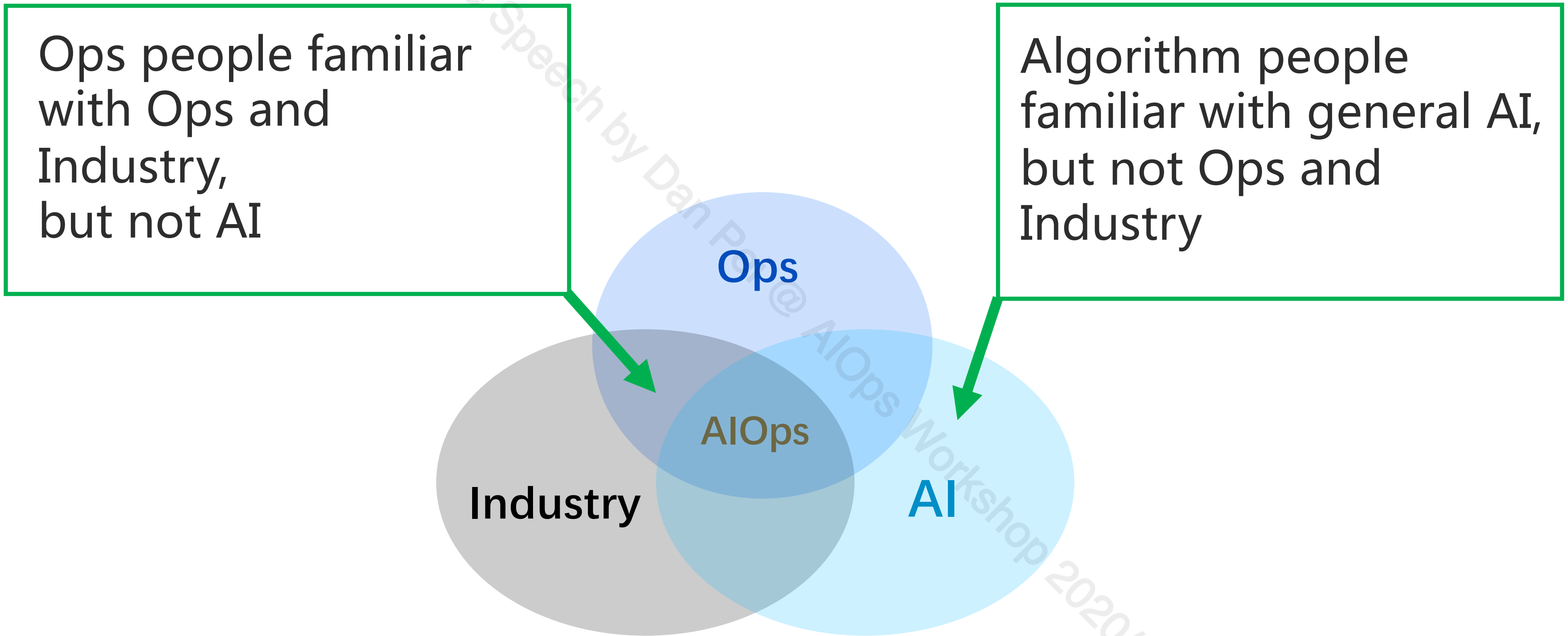
Traditional programming language:

hard-coded logic

AI as a programming language

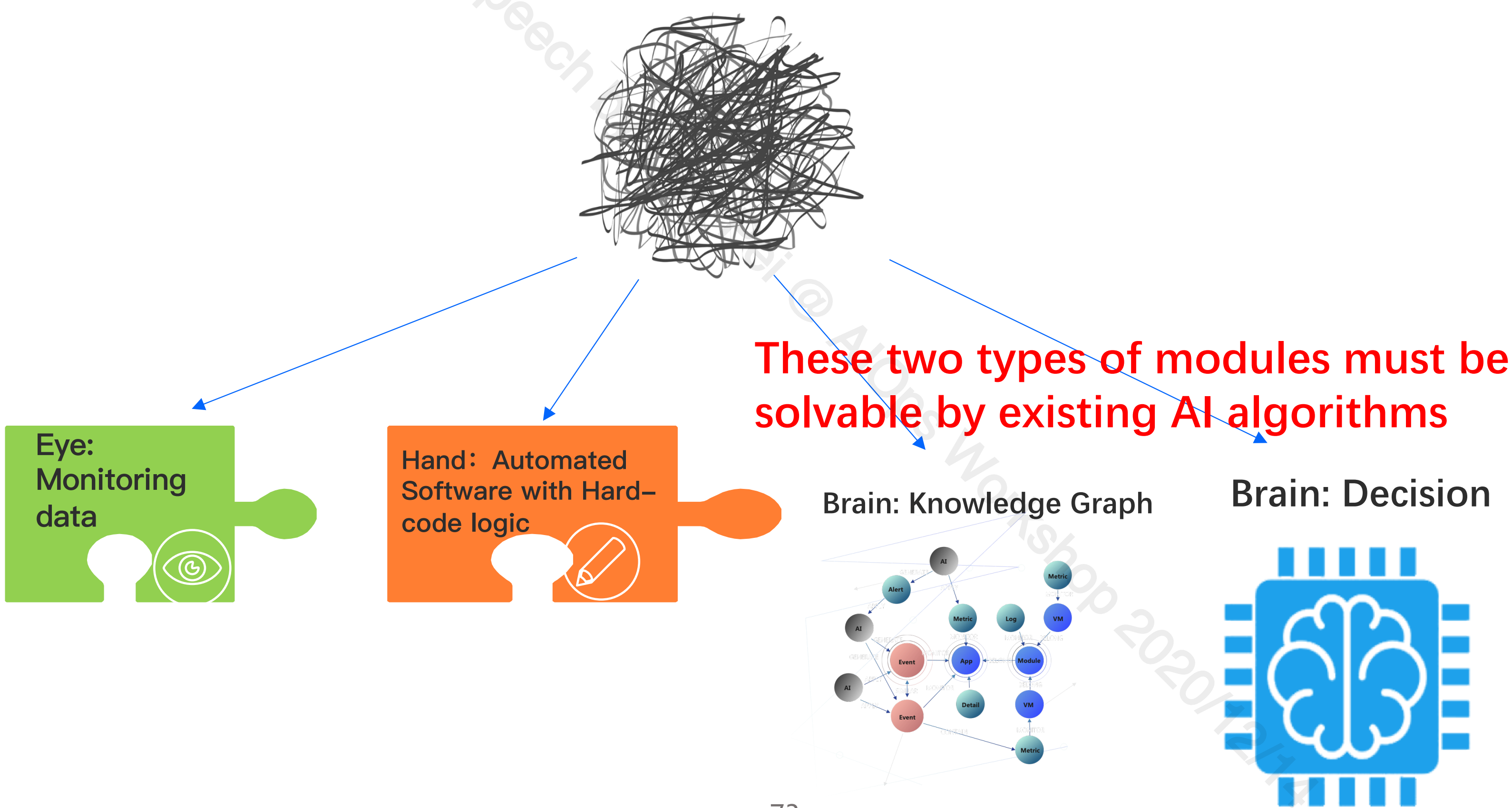
hard-coded logic + fuzzy logic learned from data

AIOps is still challenging because its interdisciplinary nature

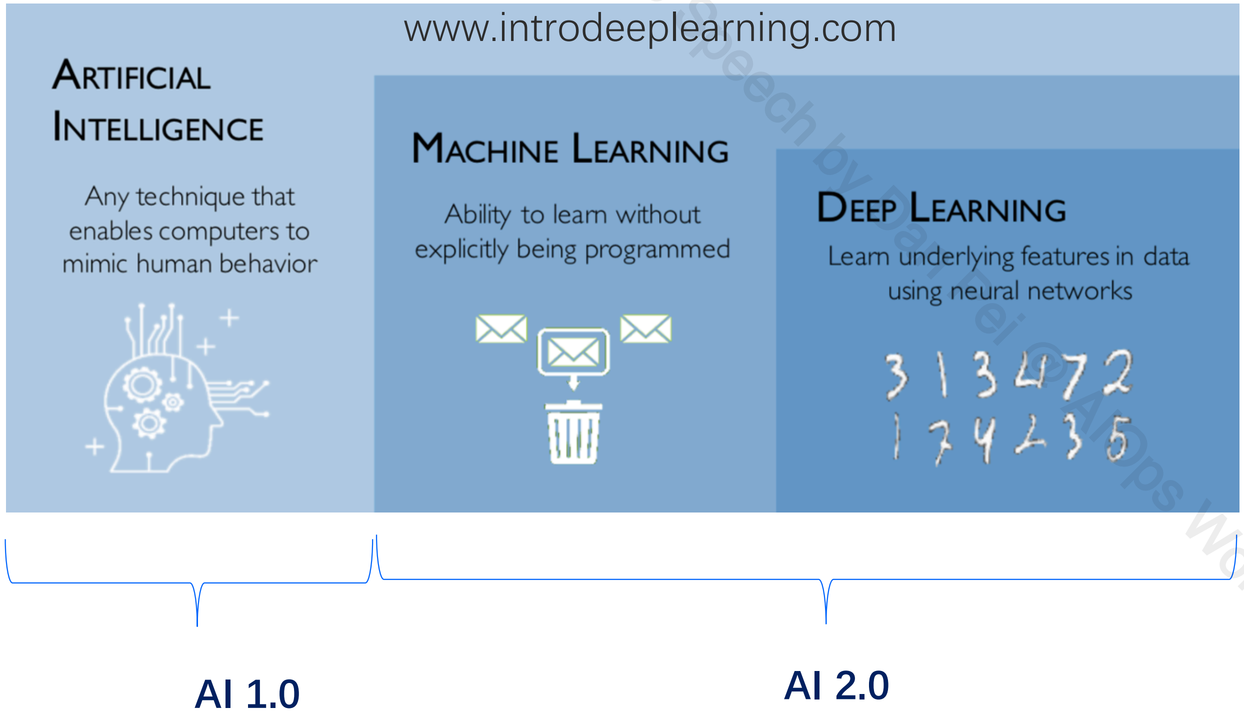


Lesson 1：Divide and Conquer instead of Using Black Box

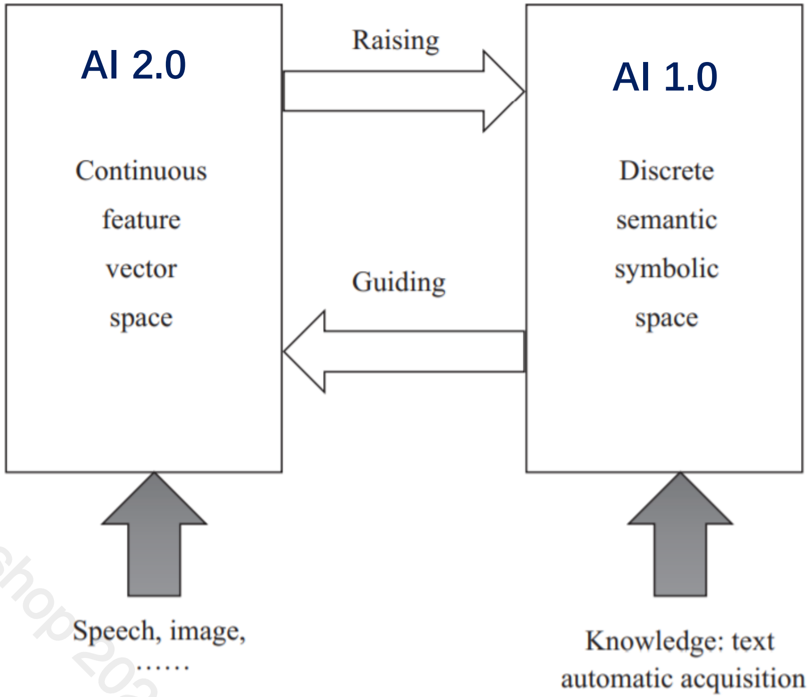
Using domain knowledge to divide



AI 3.0： Deep Learning + Knowledge Engineering



Bo Zhang, Jun Zhu, Hang Su, AI 3.0



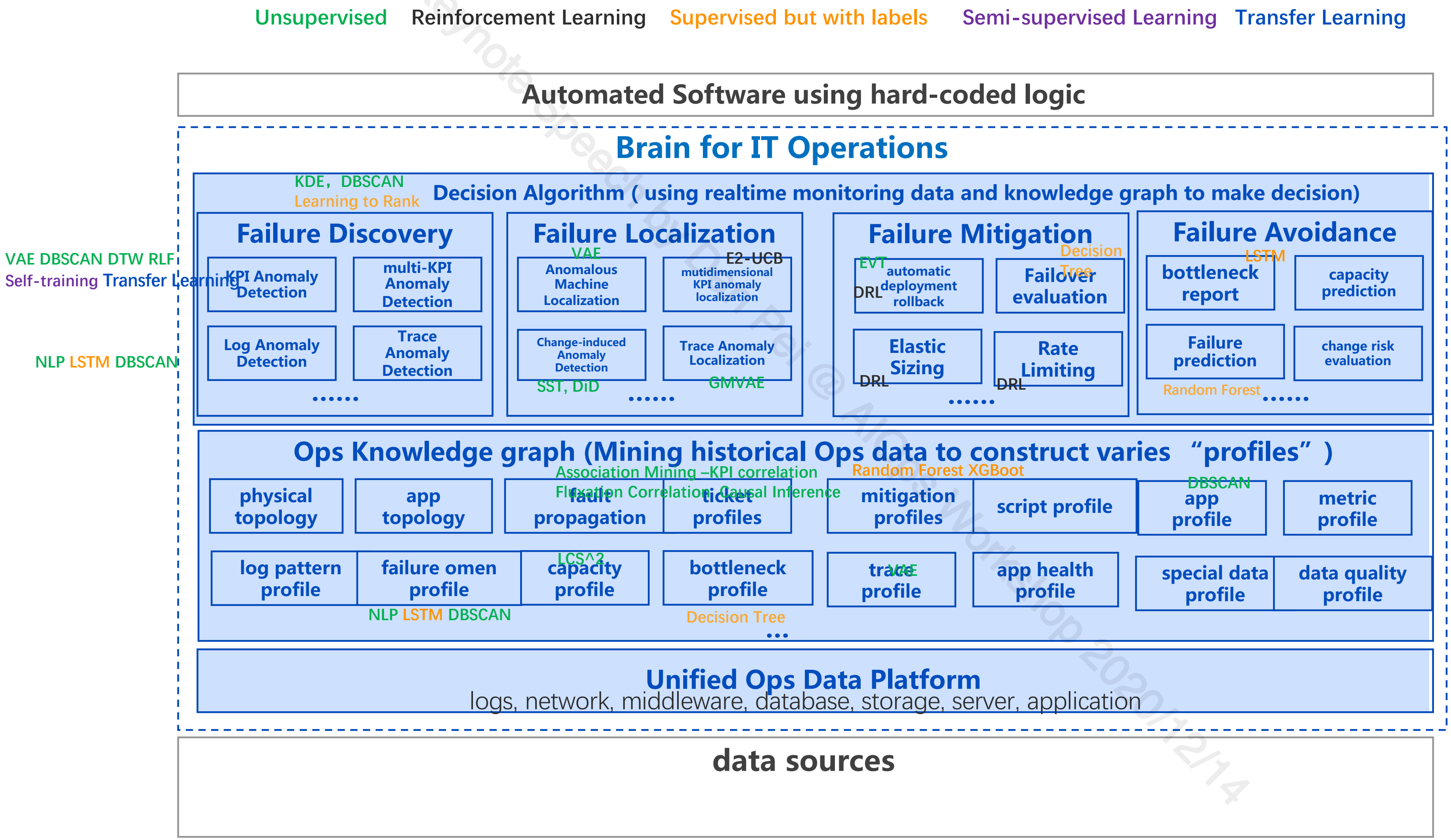
AI 3.0 = AI 1.0 + AI 2.0, still in its early research stage

Artificial Intelligence for IT Operations (AIOps)

- The major topics of AIOps often coincide with its more general counterparts in Machine Learning:
  1. Anomaly Detection in Time Series, Logs (semi-structured text), Traces (program execution trace), and Graphs
  2. Anomaly Localization
  3. Failure/Event Prediction
  4. Causal Inference and its application in Root Cause Analysis

- State-of-art Machine Learning Algorithms are applied to solve the unique challenges in AIOps:
  1.Deep Neural Networks for Time Series or Sequence
  2.Deep Generative Model (VAE, GAN)
  3.Deep Reinforcement Learning
  4.Natural Language Processing
  5.Causal Inference

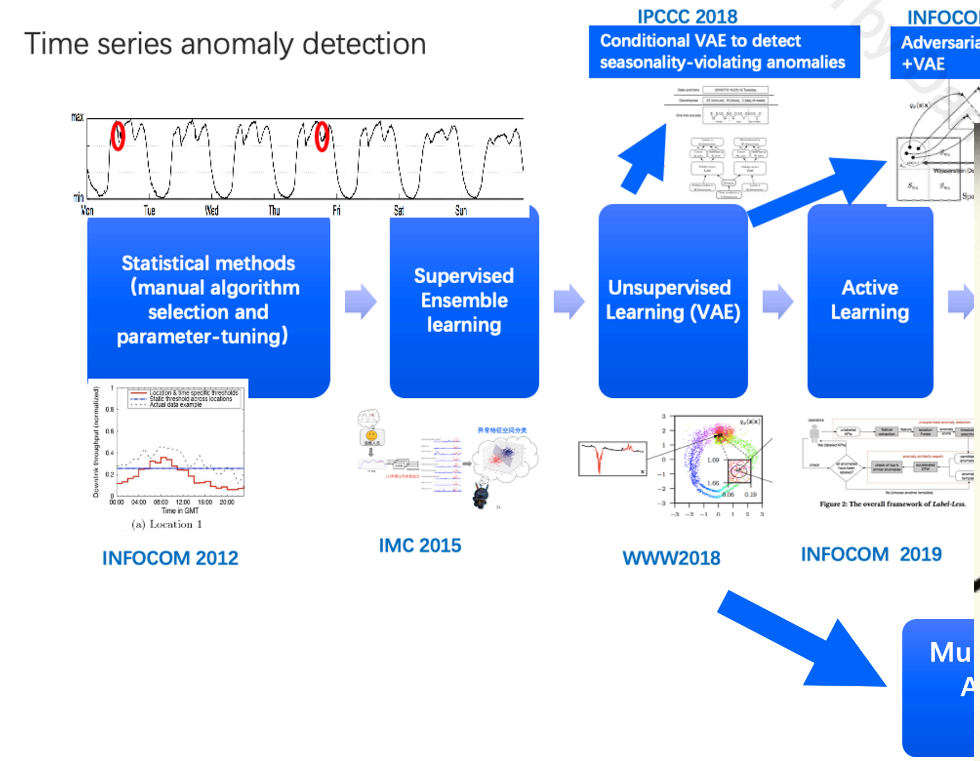
Lesson 2: Wide range of AI algorithms for AIOps

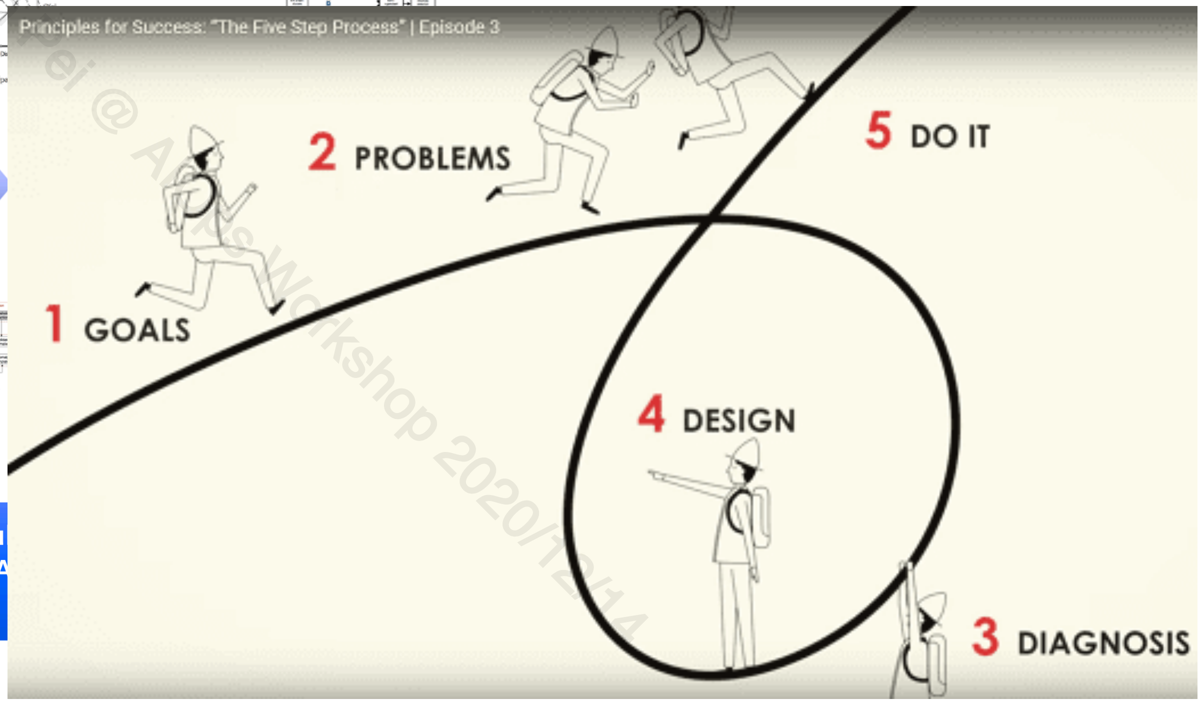


Lesson 3: From Practice, Into Practice

- 1. Discover challenging problems from Practice (specifically, IT Operations)
- 2. Design ML Algorithms to solve a problem
- 3. Deploy the algorithms in practice. If not working perfectly? go to step 1.

Time series anomaly detection





Lesson 4 : As little labeling as possible

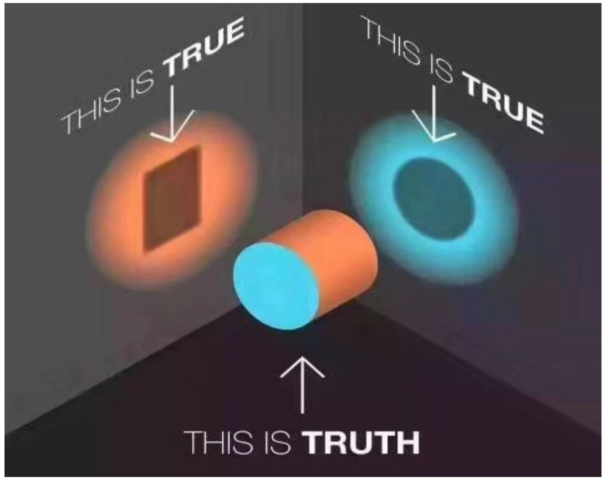
In sharp contrast with computer vision, labeling in Ops cannot be crowdsourced.

Although the users are themselves experts who can label, their preferences are still in this order:

1. **Unsupervised approaches**
2. **Unsupervised approaches + active learning (whitelisting)**
3. **Semi-supervised approaches; supervised approaches +transfer learning**
4. **Supervised approaches**

Lesson 5: Utilize as many data sources as possible

- Features
- Correlation
- Glues: topology, call graph, causal relationship



Lesson 6: it really takes time and community efforts to solve real-world IT Operations problems

"Most people overestimate what they can do in one year and underestimate what they can do in ten years."

-- Bill Gates

AIOps Challenge (http://iops.ai) to bring together community members

- 2018 AIOps Challenge: time series anomaly detection. Published labeled data from 5 Internet companies. More than 50 teams participated. Papers based on these data were published in KDD, IWQoS, etc.
  Data Downloadable @ https://github.com/NetManAIOps/KPI-Anomaly-Detection)

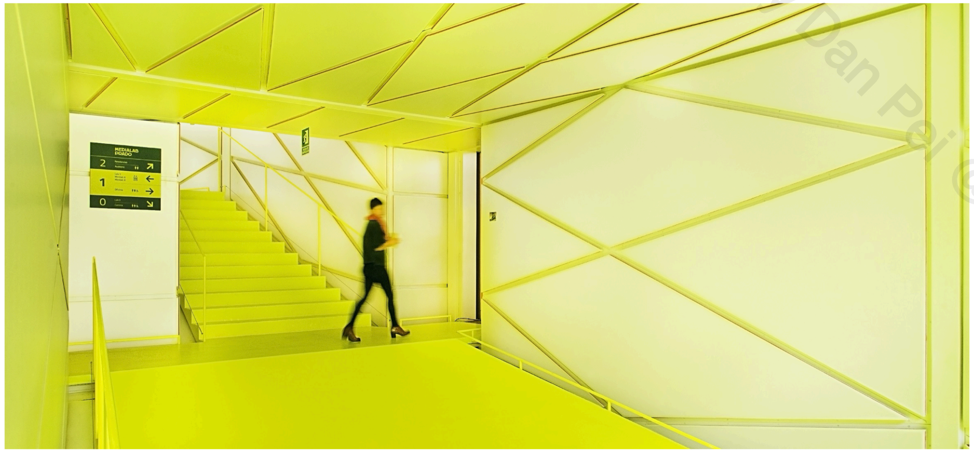
- 2019 AIOps Challenge: multi-attribute time series anomaly localization. Published data from an Internet company. More than 60 teams participated.
  Data Downloadable @ https://github.com/NetManAIOps/MultiDimension-Localization

- 2020 AIOps Challenge: Anomaly detection and localization in a microservice system. Published data from a telecom company.
  Data Downloadable @ https://github.com/NetManAIOps/AIOps-Challenge-2020-Data

ICNP HDR-Nets Workshop（Networking + Machine Learning)

HDR-Nets Workshop

The 28th IEEE International Conference on Network Protocols (ICNP 2020)
Madrid, Spain, October 13, 2020 Follow @IEEE_ICNP



1st Workshop on
Harnessing the Data Revolution in Networking

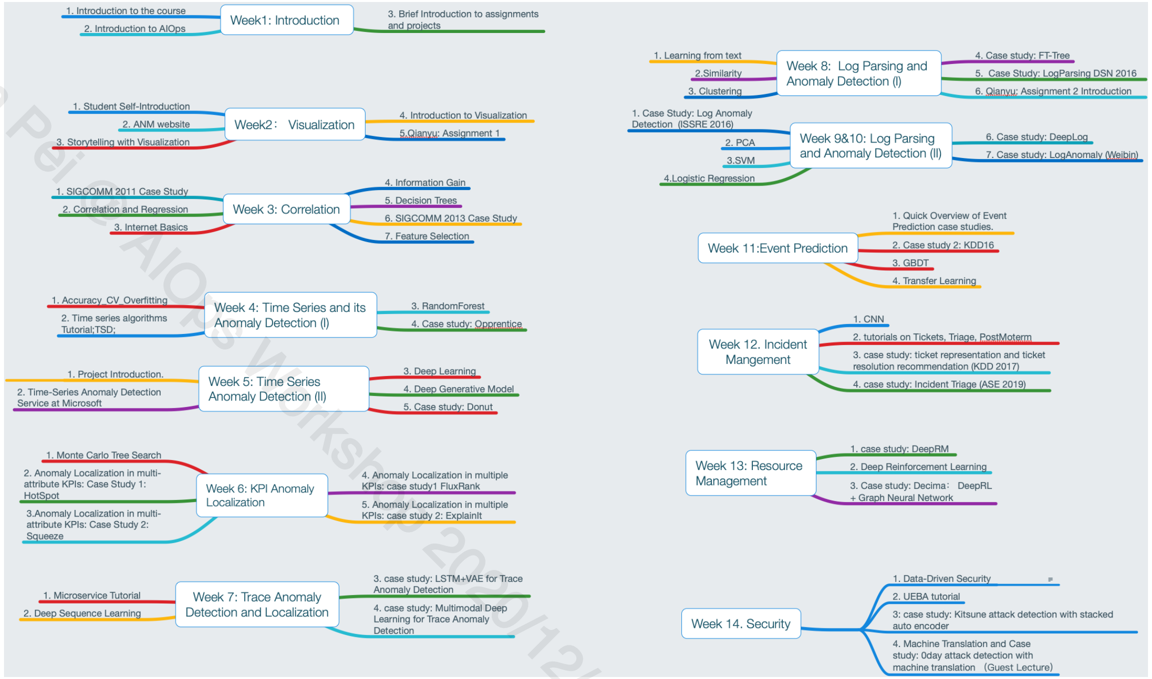
Workshop co-located with ICNP 2019 @ Chicago, Illinois, USA, October 7, 2019

HDR-Nets 2020:
https://icnp20.cs.ucr.edu/hdrnetsprogram.html

HDR-Nets 2019:
https://aiops.org/icnpworkshop.html

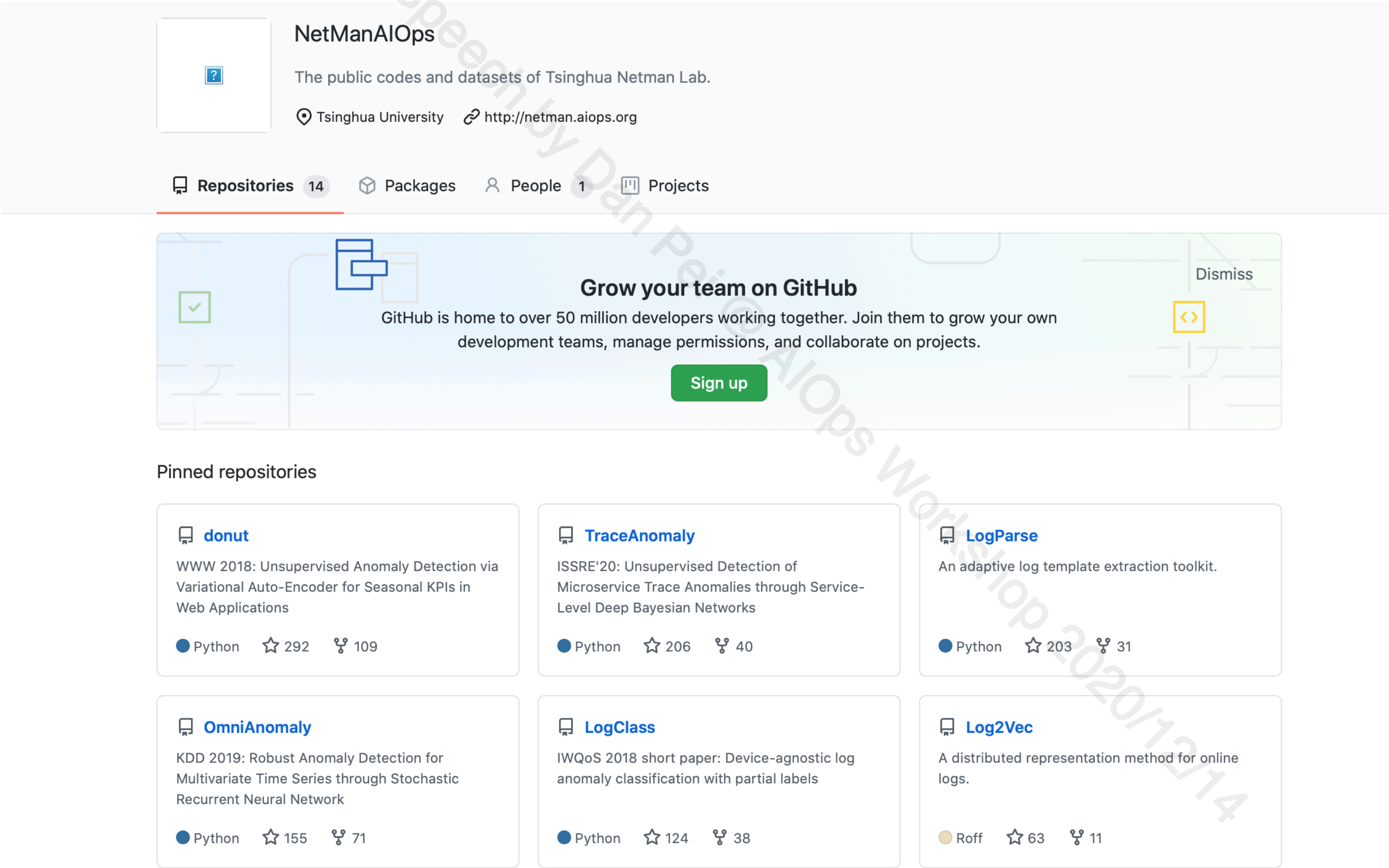
AIOps Course (in English) at Tsinghua: http://course.aiops.org

with literature collected and sorted by AIOps topics



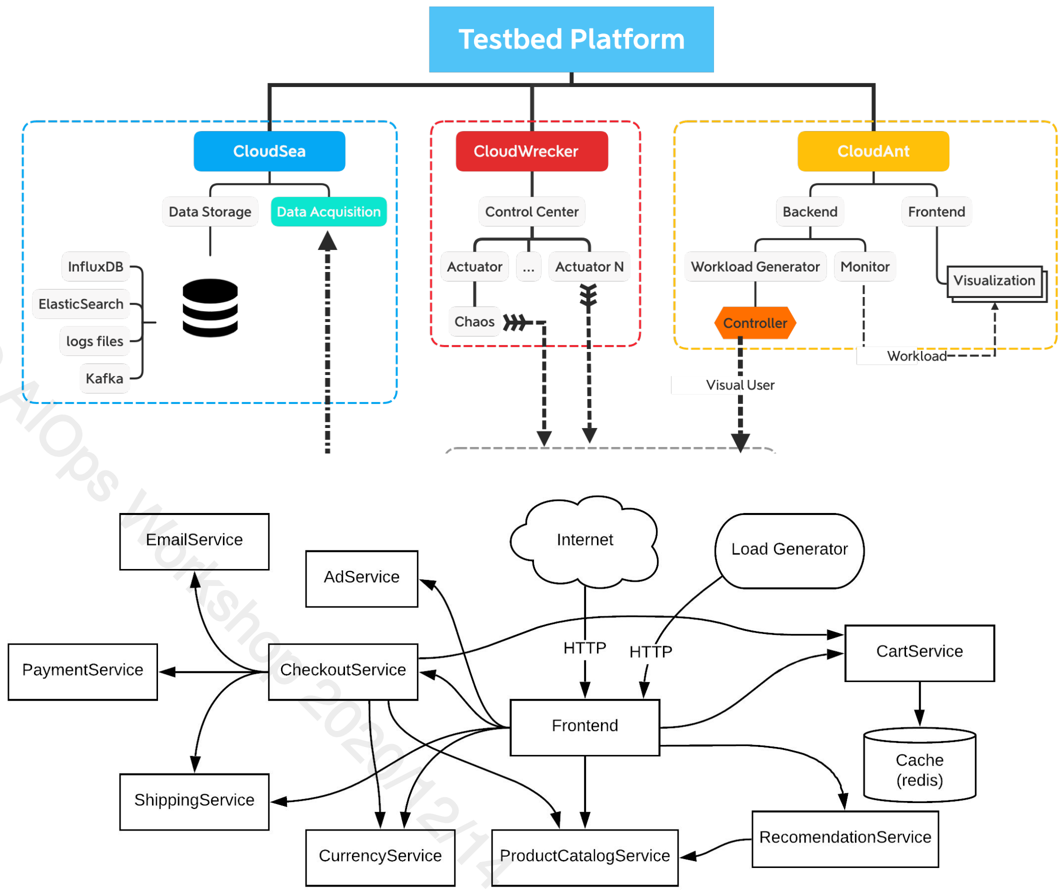
Some open-sourced algorithms from NetMan

https://github.com/netmanaiops



More community efforts needed

- Many missing pieces for a representative AIOps testbed:
  - Large-enough Industry-grade microservice based system
  - Failure patterns from industry
  - Failure injection systems
  - Realistic evaluation metrics



Summary

- AI for IT Operations (AIOps) is an interdisciplinary research field between AI and Systems/Networking/Software Engineering/Security
  - Towards Autonomous IT Operations.
- AIOps will be a foundational technology in the increasingly digitalized world
- Many deep and challenging research problems to be solved in AIOps
- Lessons learned so far:
  - Divide and conquer instead of using black box
  - Wide range of AI algorithms for AIOps
  - From practice, into practice
  - As little labeling as possible
  - Problem formulation matters
  - Utilize as many data sources as possible
- Long-term community efforts are needed to solve AIOps problems

Thanks!

Q&A

Wechat: peidanwechat

清華大學 | NetMan Tsinghua