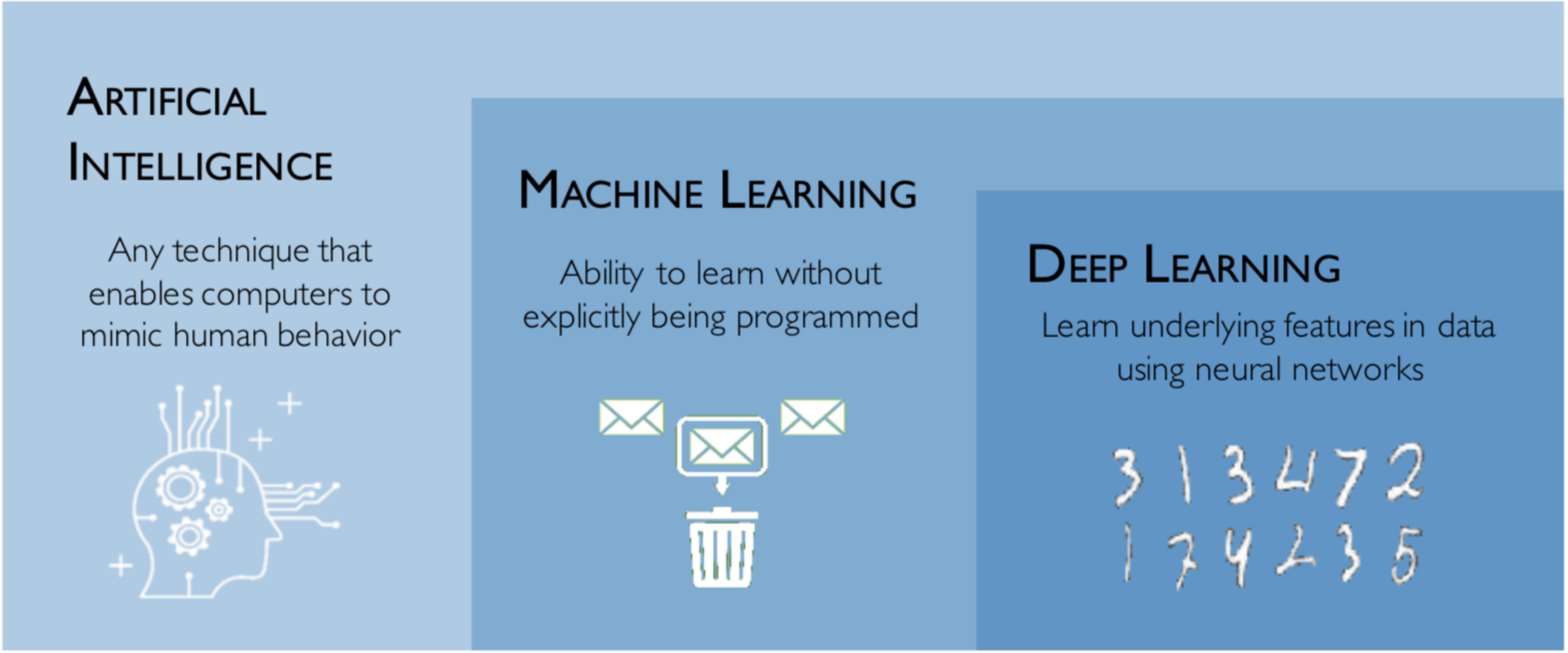
# Towards Autonomous IT Operations through Machine Learning

Dan Pei

清華大學 | Tsinghua NetMan

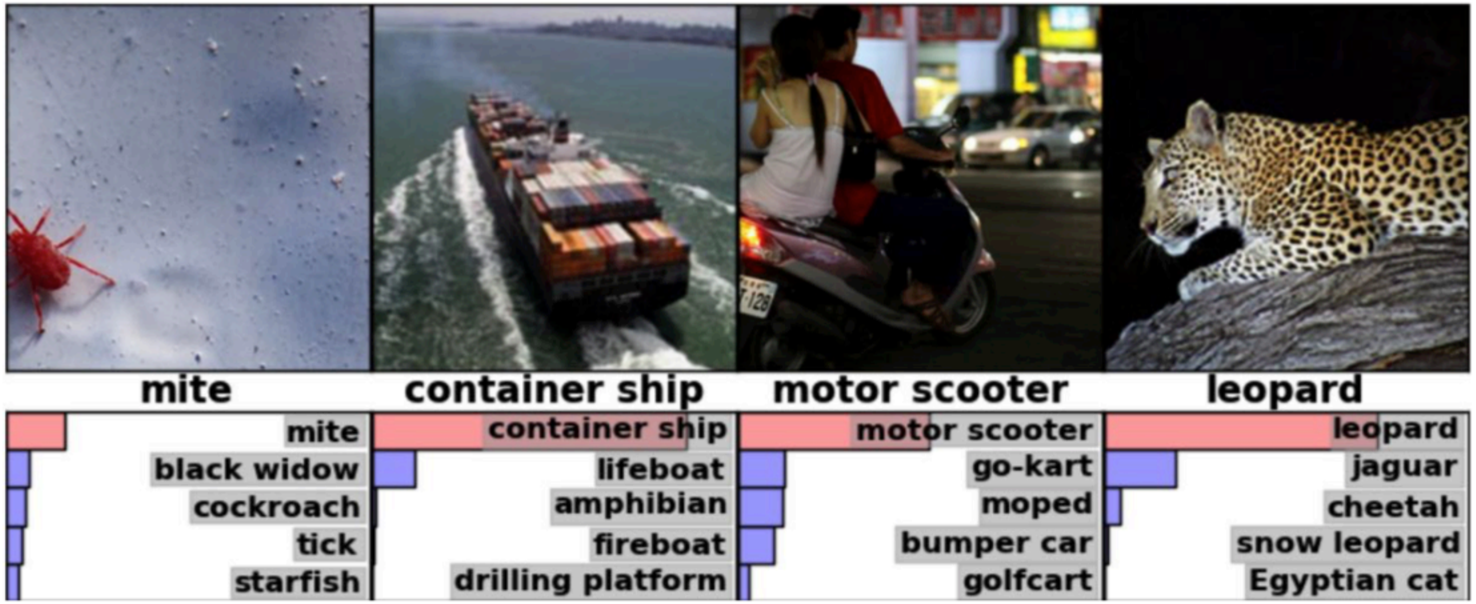
# What are AI, Machine Learning and Deep Learning?



# Deep Learning Success: Vision

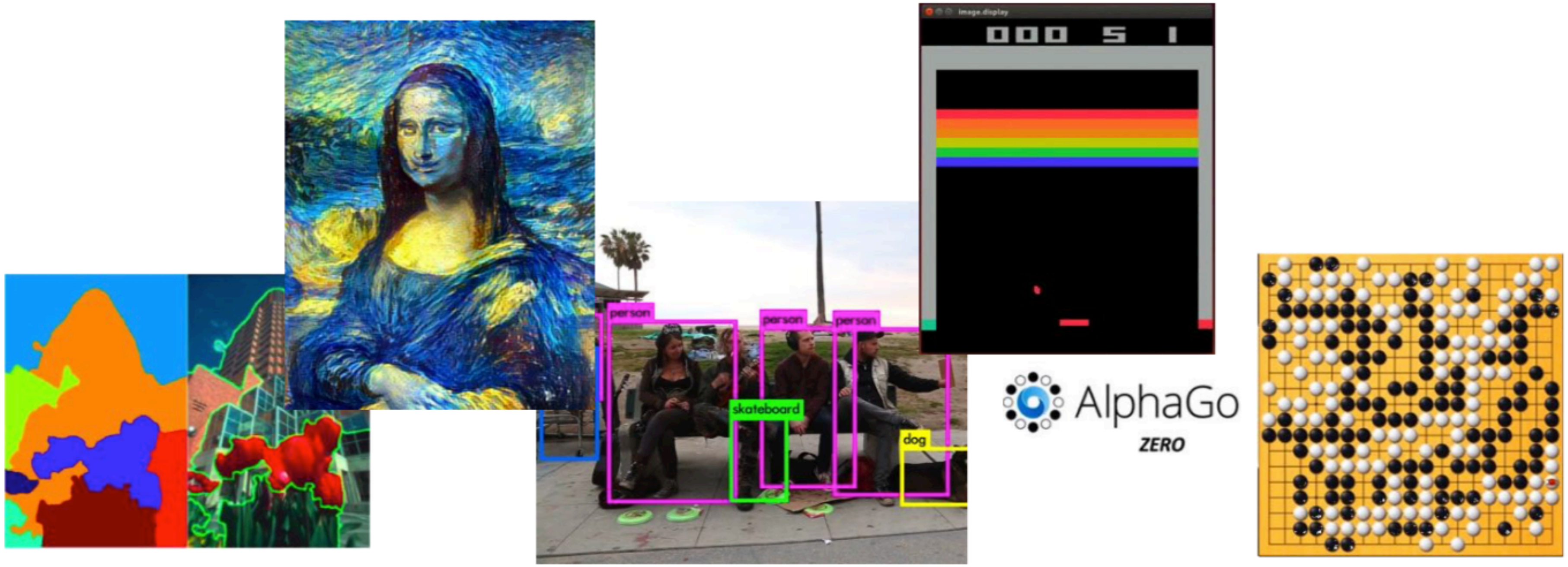
Image Recognition

IMAGENET



# Deep Learning Success

**And so many more…**



# Why Now?

1952 — Stochastic Gradient Descent

1958 — Perceptron
- Learnable Weights

⋮

1986 — Backpropagation
- Multi-Layer Perceptron

1995 — Deep Convolutional NN
- Digit Recognition

⋮

Neural Networks date back decades, so why the resurgence?

## 1. Big Data
- Larger Datasets
- Easier Collection & Storage

IMAGENET

WIKIPEDIA
The Free Encyclopedia

## 2. Hardware
- Graphics Processing Units (GPUs)
- Massively Parallelizable


## 3. Software
- Improved Techniques
- New Models
- Toolboxes

TensorFlow

# Industries being changed by AI

- Finance
- Education
- **TMT**
- **Medical & Health**
- **Automobile**
- **Manufacturing**

# Deep Learning Success: Audio

Other sequences-model applications:

- predict stock price
- machine translation
- …

## Music Generation

**Temporal dependence**

# Deep Learning Success: Vision

Detect pneumothorax in real X-Ray scans

# 5 Applications Of AI In The Automotive Industry

1

**Driving Features**

AI lends itself perfectly to powering advanced safety features for connected vehicles.

2

**Cloud Services**

The application of artificial intelligence cloud platforms ensure that data is available when needed.

3

**Automotive Insurance**

AI speeds up the process of filing claims when accidents do occur.

4

**Car Manufacturing**

Robots are driving optimisation and the rethinking of processes and production in innovative new ways.

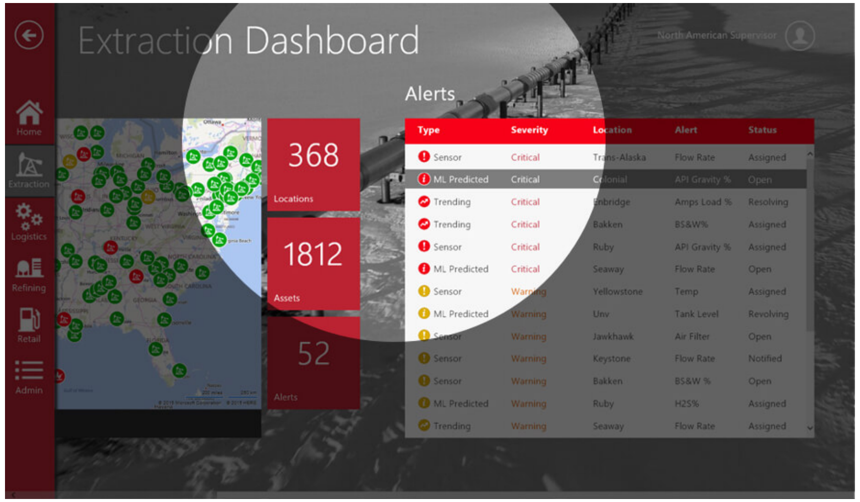
5

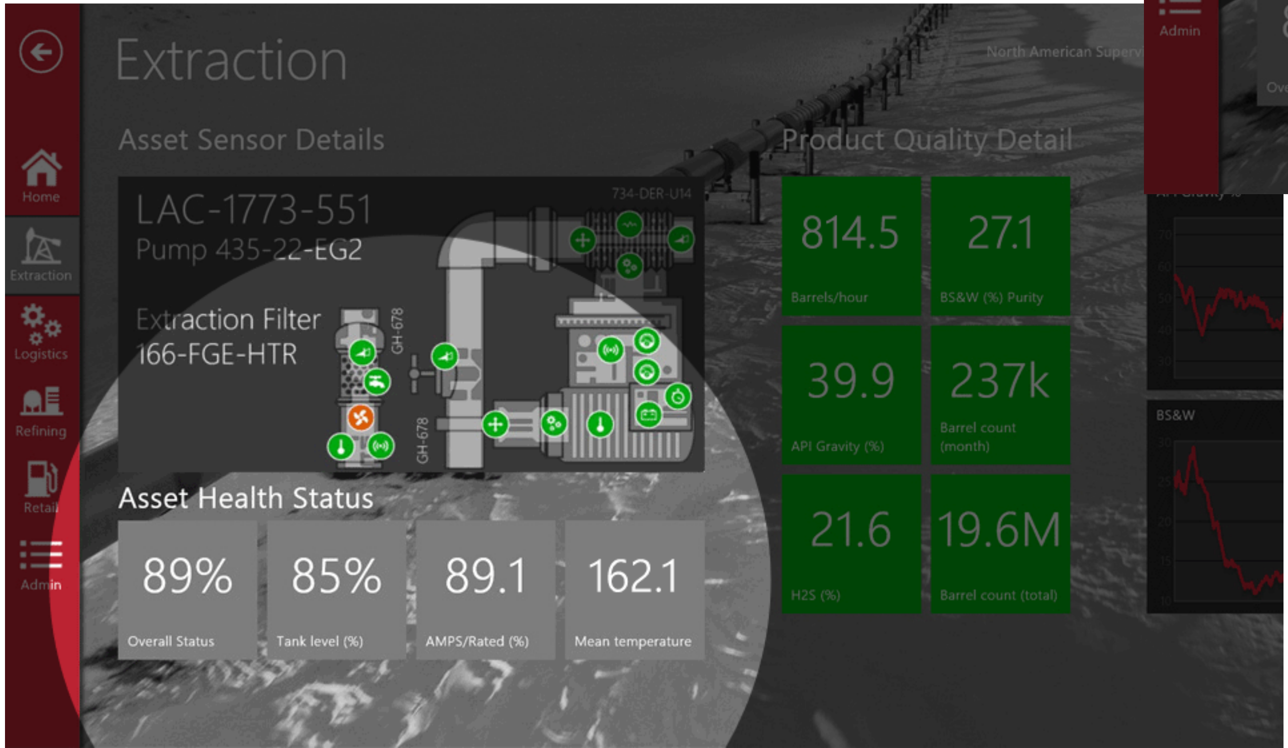
**Driver Monitoring**

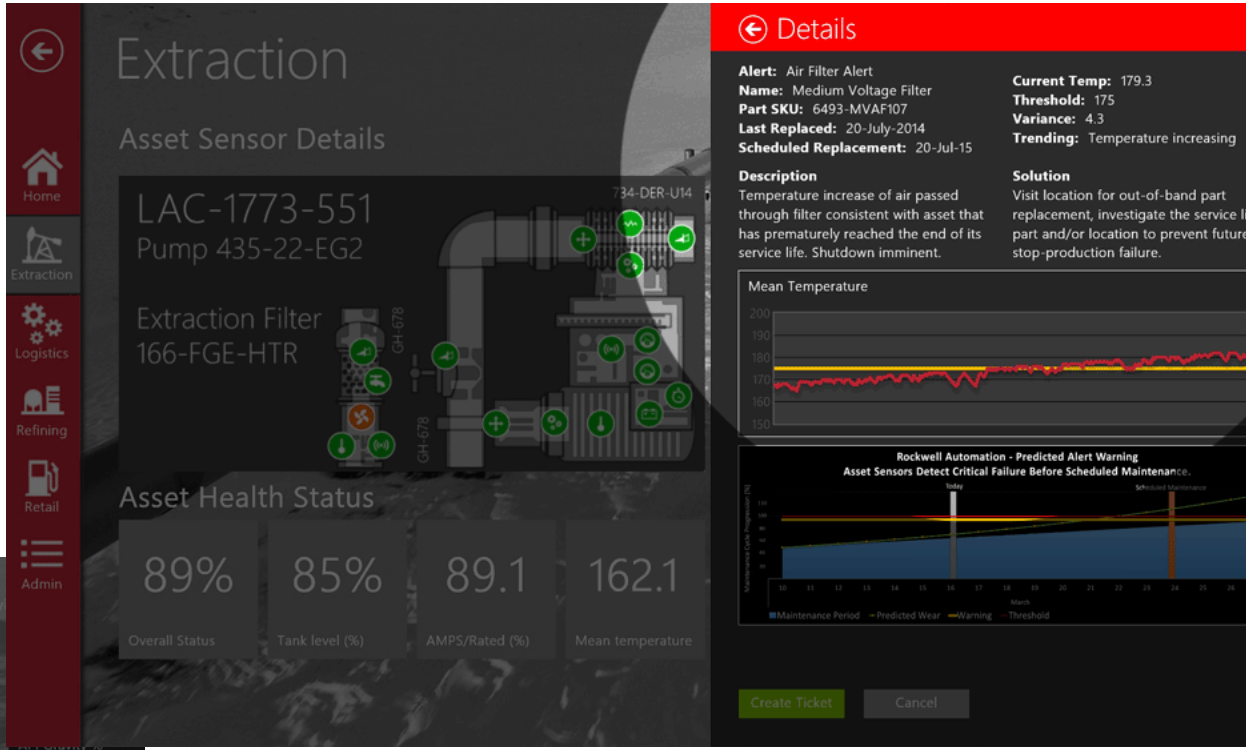
AI software detects driver behavior in four key areas: driver identification, recognition, monitoring and infotainment control.

https://youtu.be/nBs3K0bsxyc

# Predictive Maintenance



# Machine Learning is a high-level programming language

**Success in specific application scenario  in specific area in specific industry**:
quality assurance in manufacturing industry


Tobacco Leaf

Wood Floor

Steel Industry

8K video monitoring of the production line

(Play video)

Traditional programming language:

hard-coded logic

Machine learning as a programming language

hard-coded logic + fuzzy logic learned from data

# 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

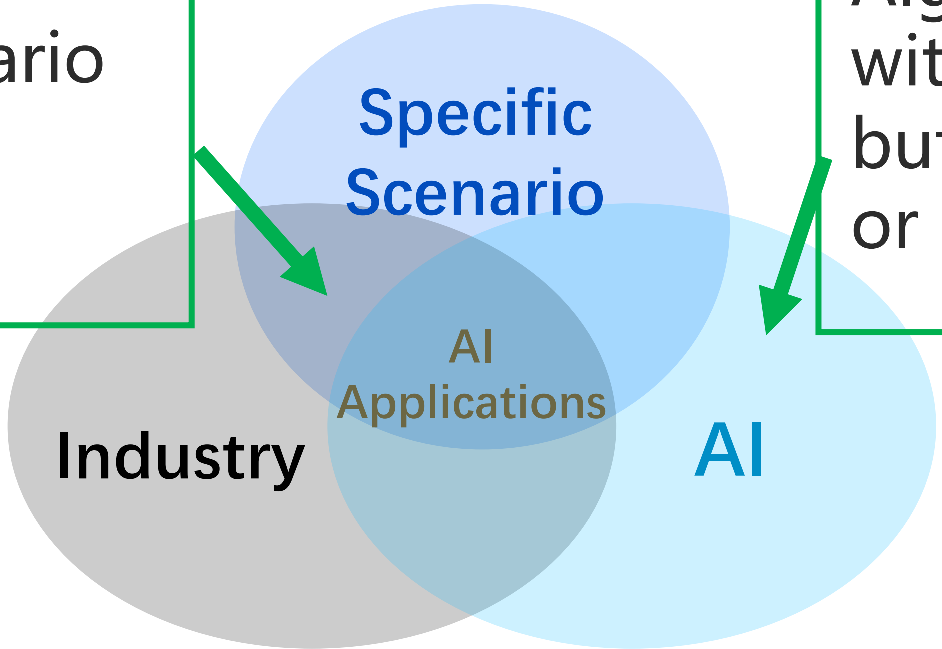
(4) Well-defined

(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

Machine learning as a programming language

hard-coded logic + fuzzy logic learned from data

# Pitfalls: use ML algorithms as Blackbox to tackle a specific scenario in a specific industry

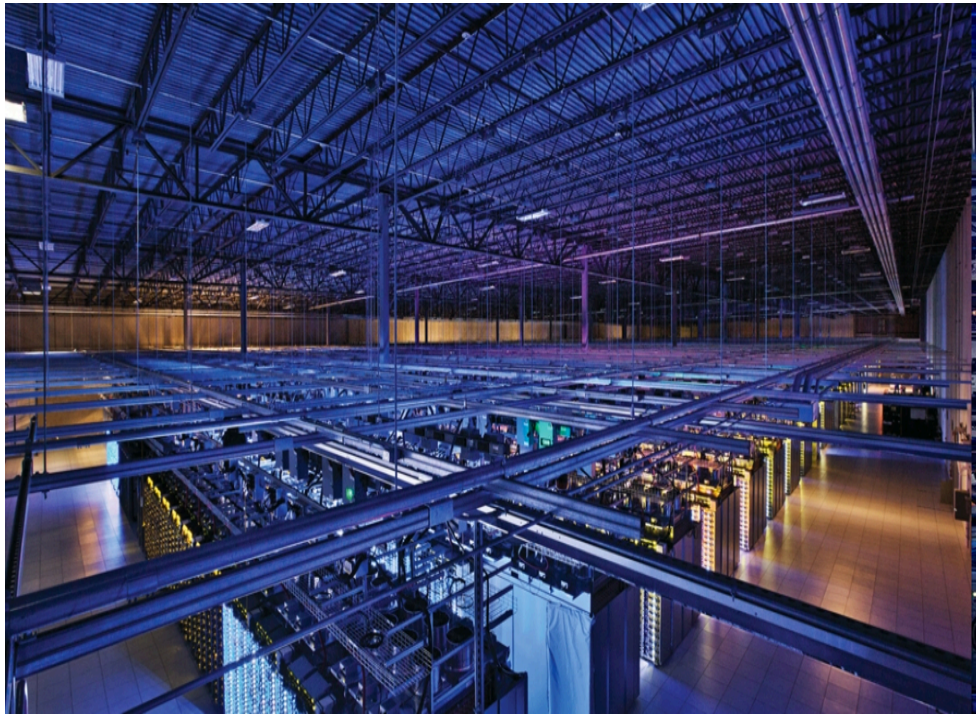
a specific scenario in a specific industry

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

# *IT Operations:* one of the technology foundations of the increasingly digitalized world




# A real case in a global top bank: labor-intensive, stressful, and ineffective

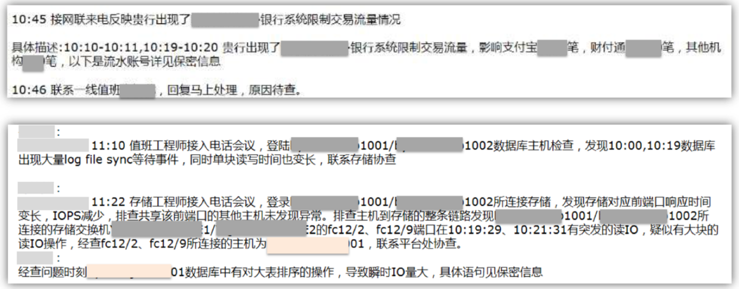
## Manual

30 Engineers involved

10:20 large number of transaction failures

Realized there was a failure when customers called 10:45

Failure localized 11:10



10:45 接网联来电反映贵行出现了[redacted]银行系统限制交易流量情况

具体描述:10:10-10:11,10:19-10:20 贵行出现了[redacted]银行系统限制交易流量，影响支付宝[redacted]笔，财付通[redacted]笔，其他机构[redacted]笔，以下是流水账号详见保密信息

10:46 联系一线值班[redacted]，回复马上处理，原因待查。

[redacted]:
[redacted] 11:10 值班工程师接入电话会议，登陆[redacted]1001/[redacted]1002数据库主机检查，发现10:00,10:19数据库出现大量log file sync等待事件，同时单块读写时间也变长，联系存储协查

[redacted]:
[redacted] 11:22 存储工程师接入电话会议，登录[redacted]1001/[redacted]1002所连接存储，发现存储对应前端口响应时间变长，IOPS减少，排查共享该前端口的其他主机未发现异常。排查主机到存储的整条链路发现[redacted]1001/[redacted]1002所连接的存储交换机[redacted]1/[redacted]2的fc12/2、fc12/9端口在10:19:29、10:21:31有突发的读IO，疑似有大块的读IO操作，经查fc12/2、fc12/9所连接的主机为[redacted]01，联系平台处协查。
[redacted]:
经查问题时刻[redacted]01数据库中有对大表排序的操作，导致瞬时IO量大，具体语句见保密信息

Failure discovery: 25mins after the failure happened
Failure localization: 25mins after failure discovery

## Replayed the data with our ML-based failure discovery and localization algorithms

10:21 automatically detected the failure

10:23 automatically localized the failure

Failure mitigation time reduced by 90%

# Autonomous IT Operations: use machine learning 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 |

"In addition to control plane and data plane, 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

# Industry opinions on machine learning’ 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, ...

# Some IT Operations Companies

*All collect IT Operations data and offer AIOps (AI for IT Operations) productions*

servicenow

Valued at 91 Billion USD

splunk

Valued at 29 Billion USD

elastic

Valued at 9 Billion USD

dynatrace

Valued at 11 Billion USD

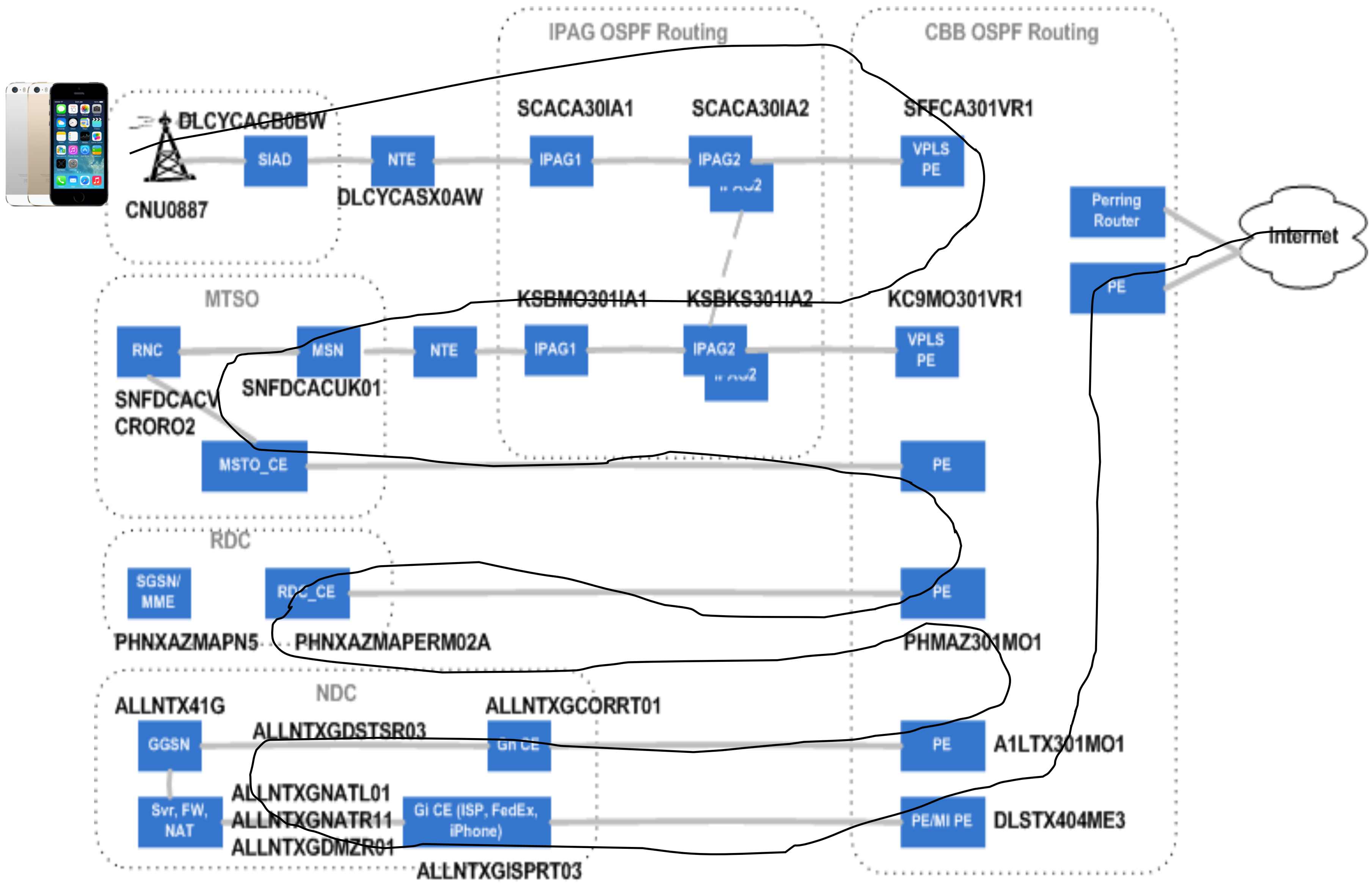
DATADOG

Valued at 27 Billion USD

# Outline

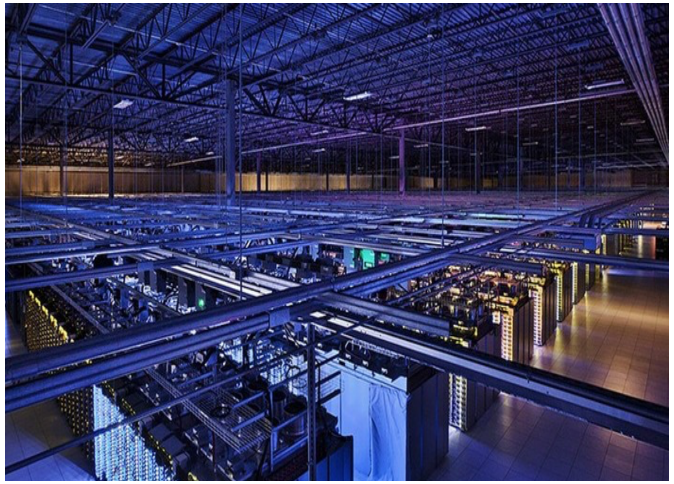
- IT Operations (Ops) background
- *Is machine learning necessary for Ops?*
- Brief Case Studies
- Unsupervised Anomaly Detection in Ops
- Lessons Learned

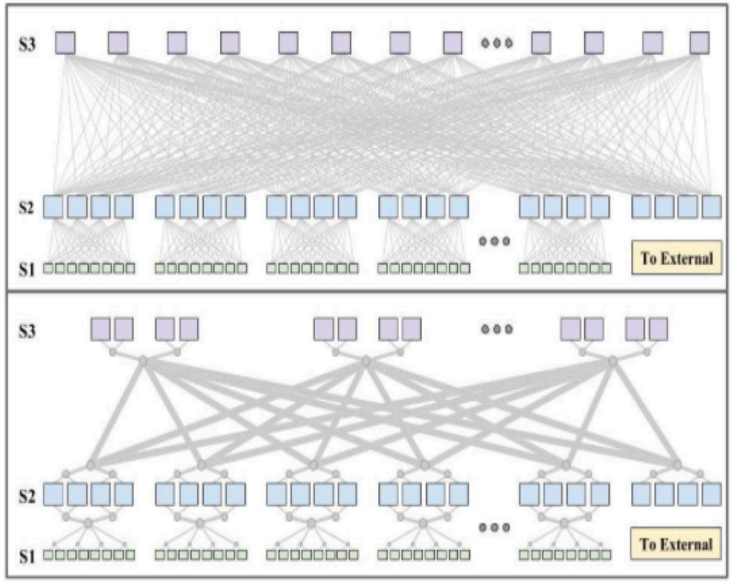
# Complex Edge Networks

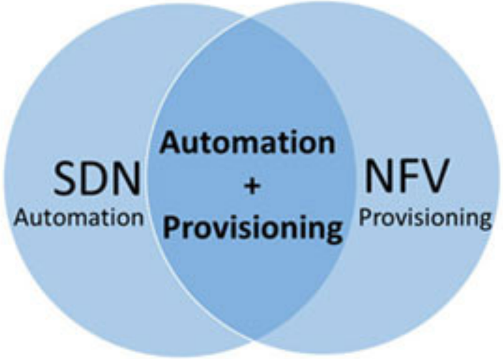


# Complex and Evolving Data Center Hardwares

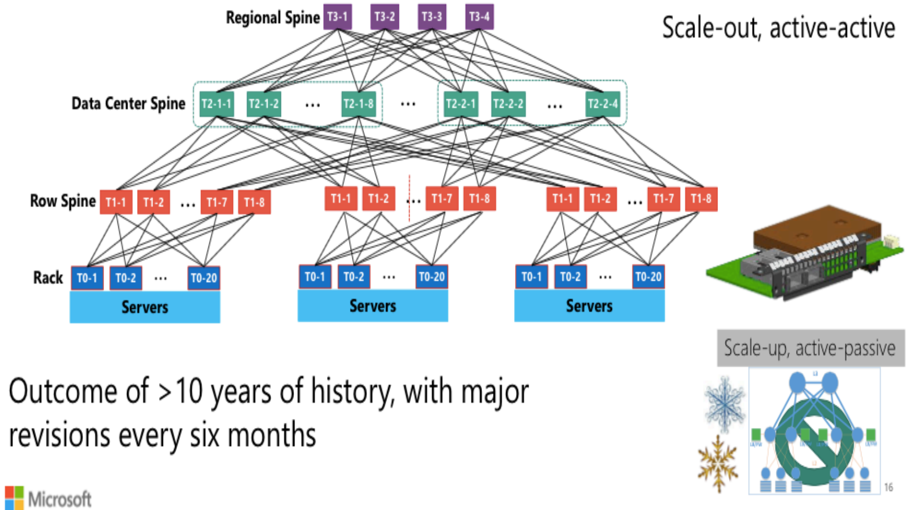
10s of thousands of servers





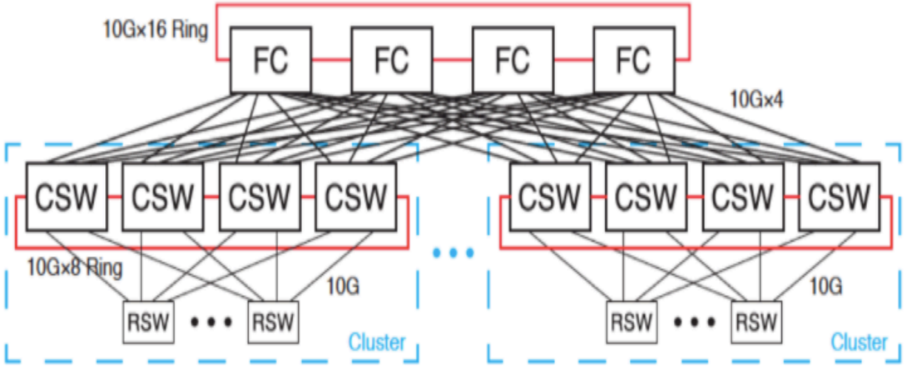




Frequent topology changes



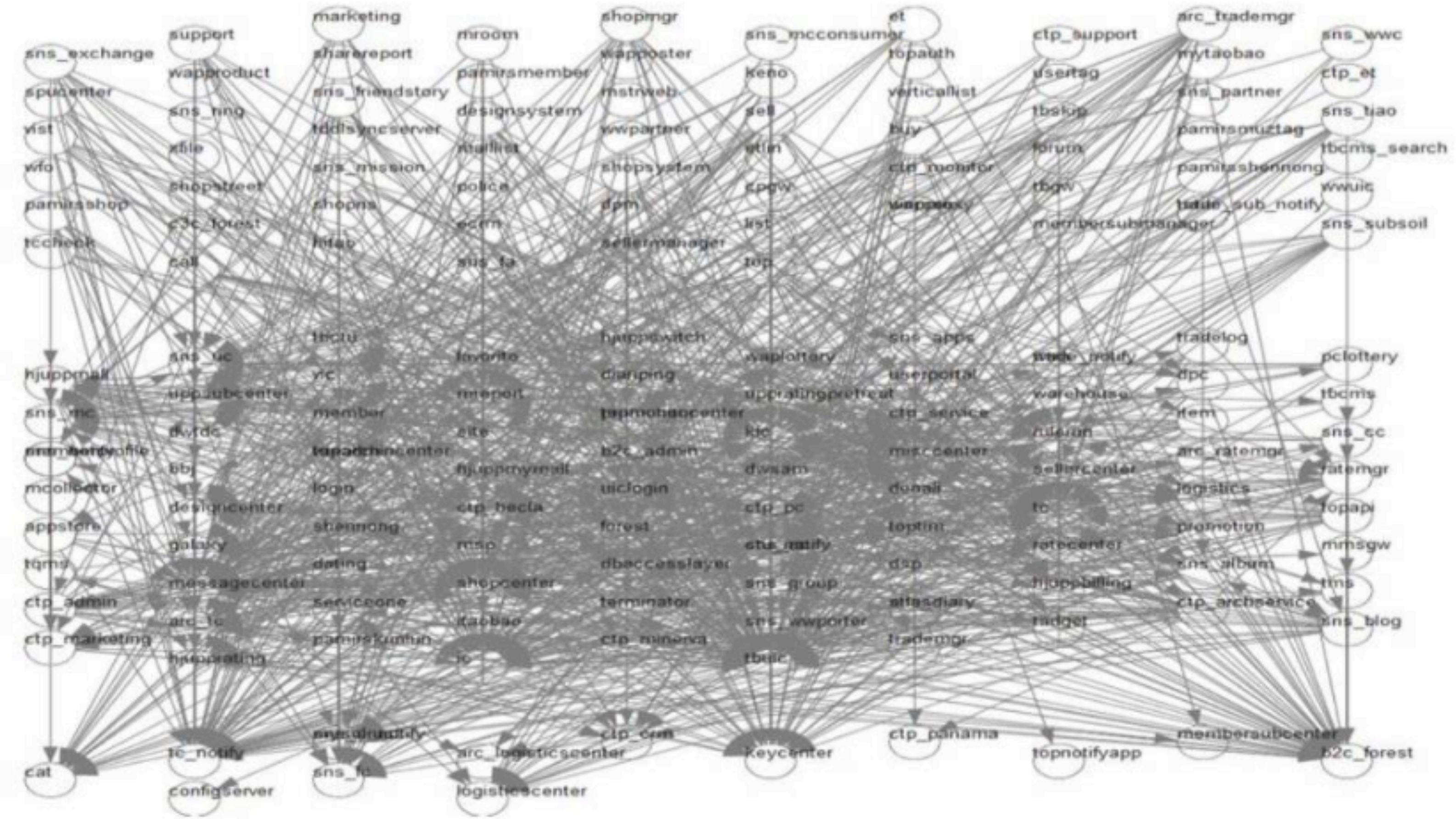

Outcome of >10 years of history, with major revisions every six months

Microsoft



# Complex Software Module Dependences

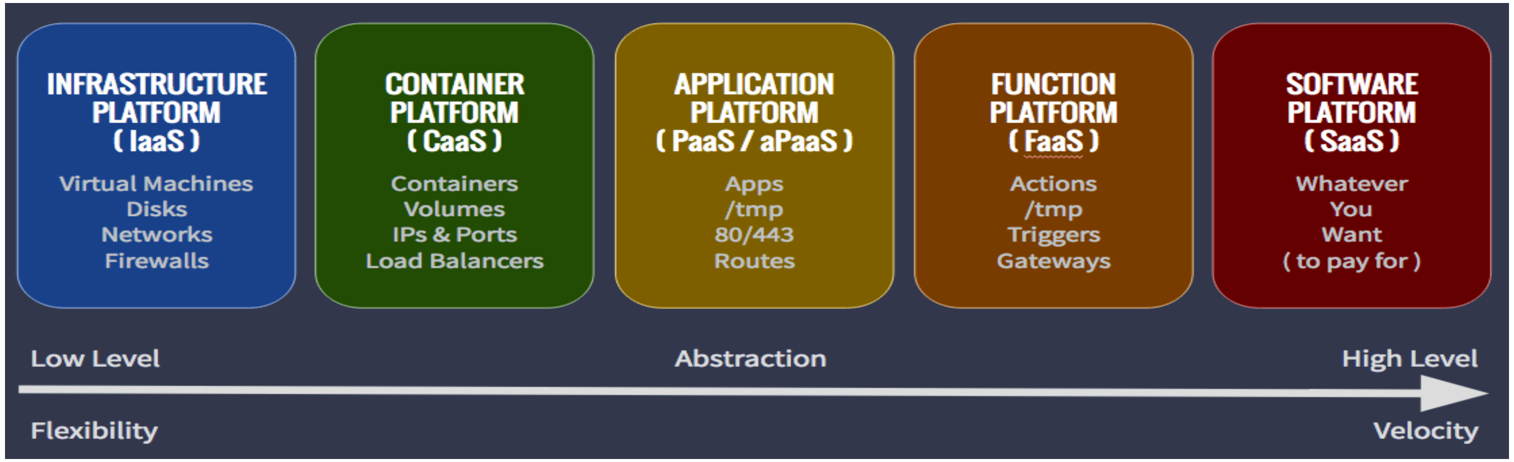
*Application dependency atTaobao (largest online shopping website in China) in 2012*

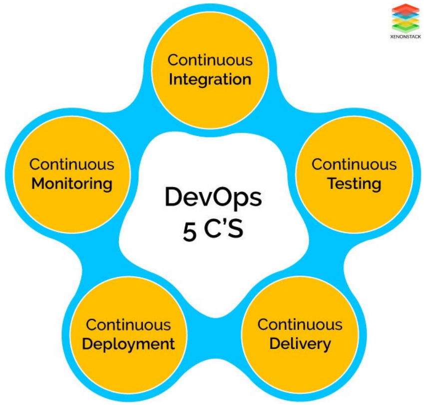


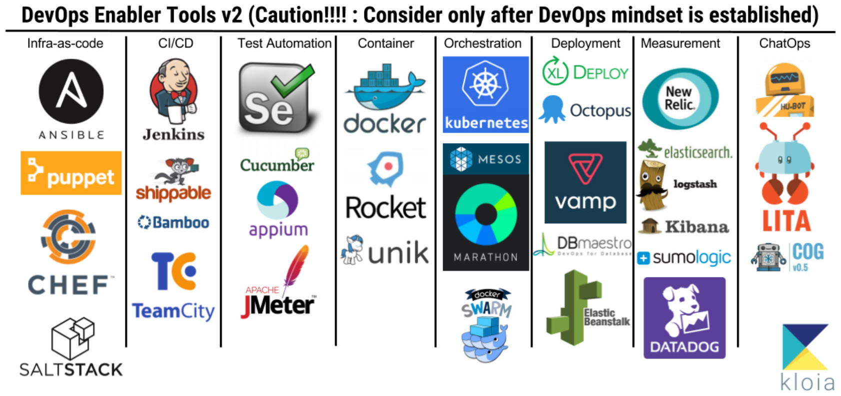

2012 淘宝核心链路应用拓扑图

# Evolving Techniques Enable Frequent Software Changes

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



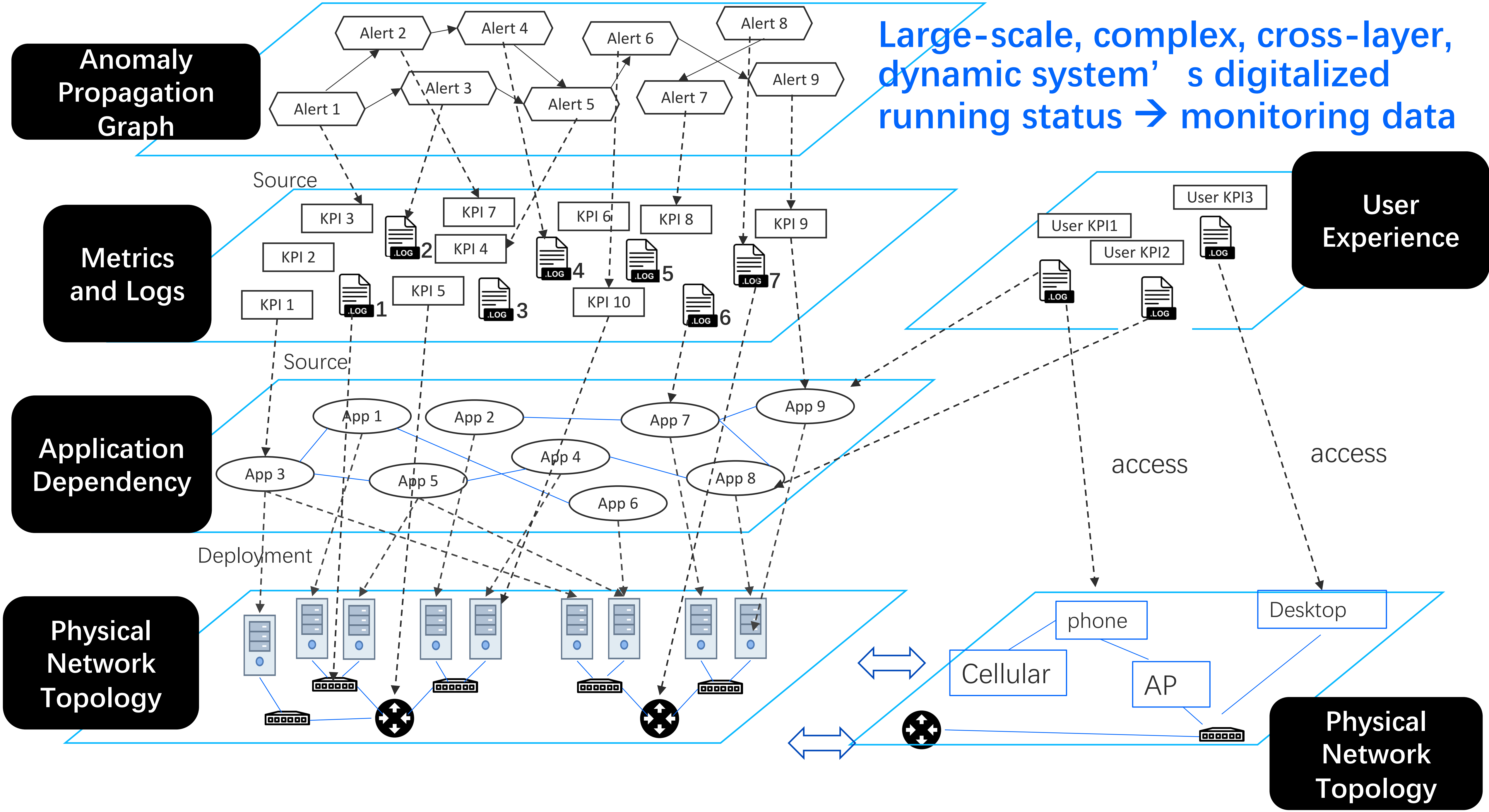




DevOps

Continuous Integration/Continuous Delivery

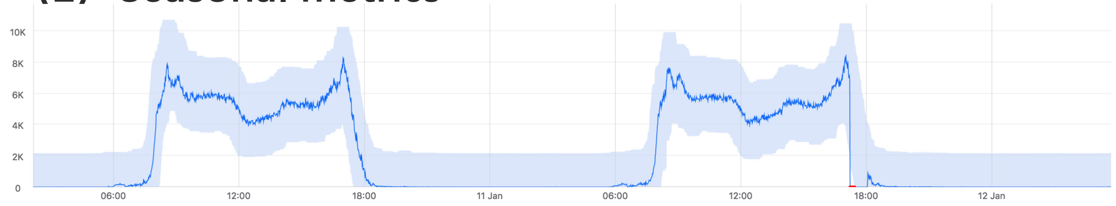
## Large-scale, complex, cross-layer, dynamic system’ s digitalized running status → monitoring data



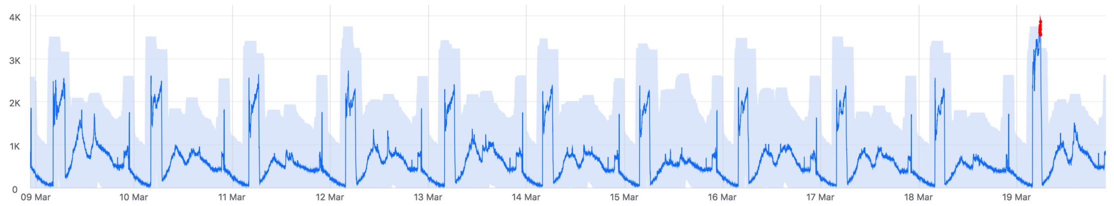
# Diverse Metrics and Their Diverse Anomalies

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

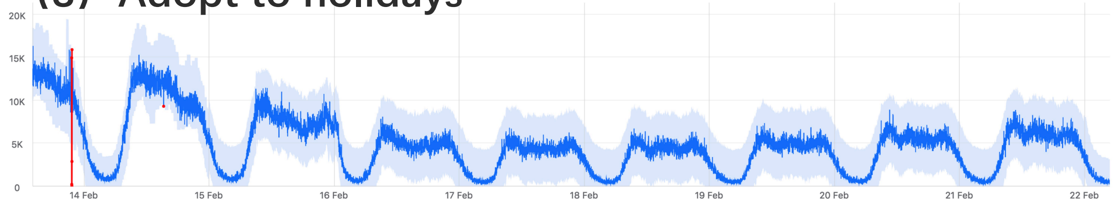
## (1) Seasonal metrics



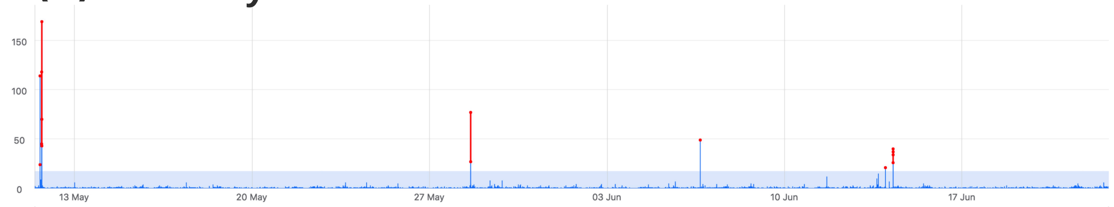

## (2) Periodicity shift



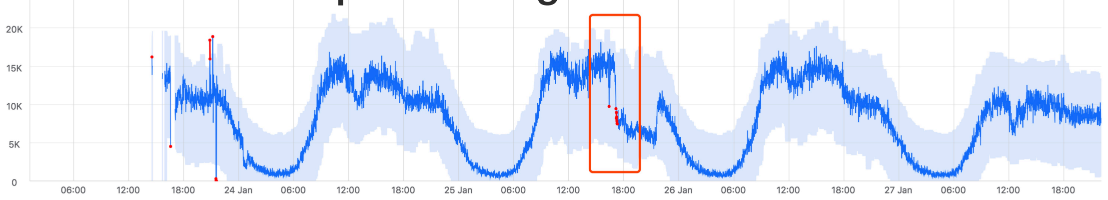

## (3) Adopt to holidays



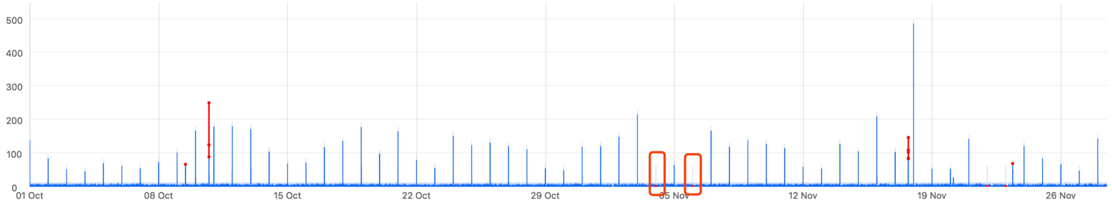

## (4) Identify variable metrics and obtain extreme threshold



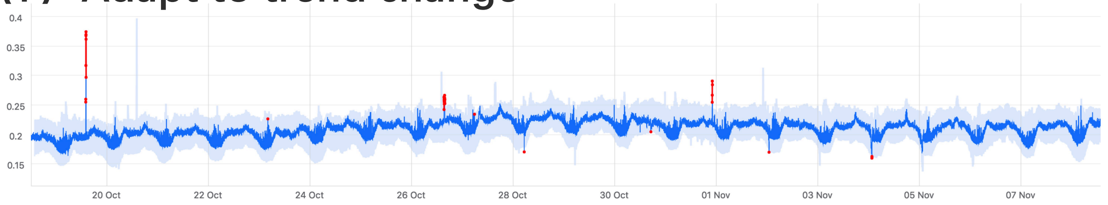

## (5) Detect too rapid a change



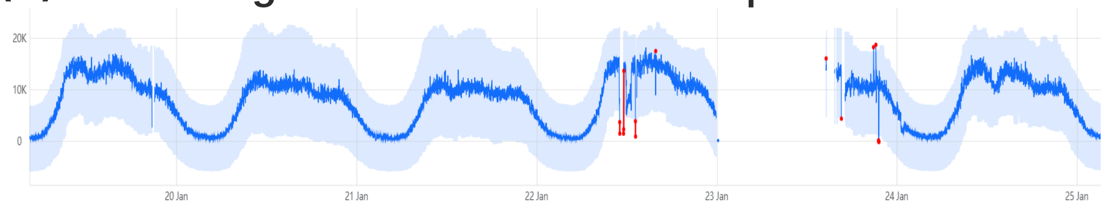

## (6) Detect the lack of seasonality.




## (7) Adapt to trend change




## (8) Robust against data loss or interruption



# 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
- …

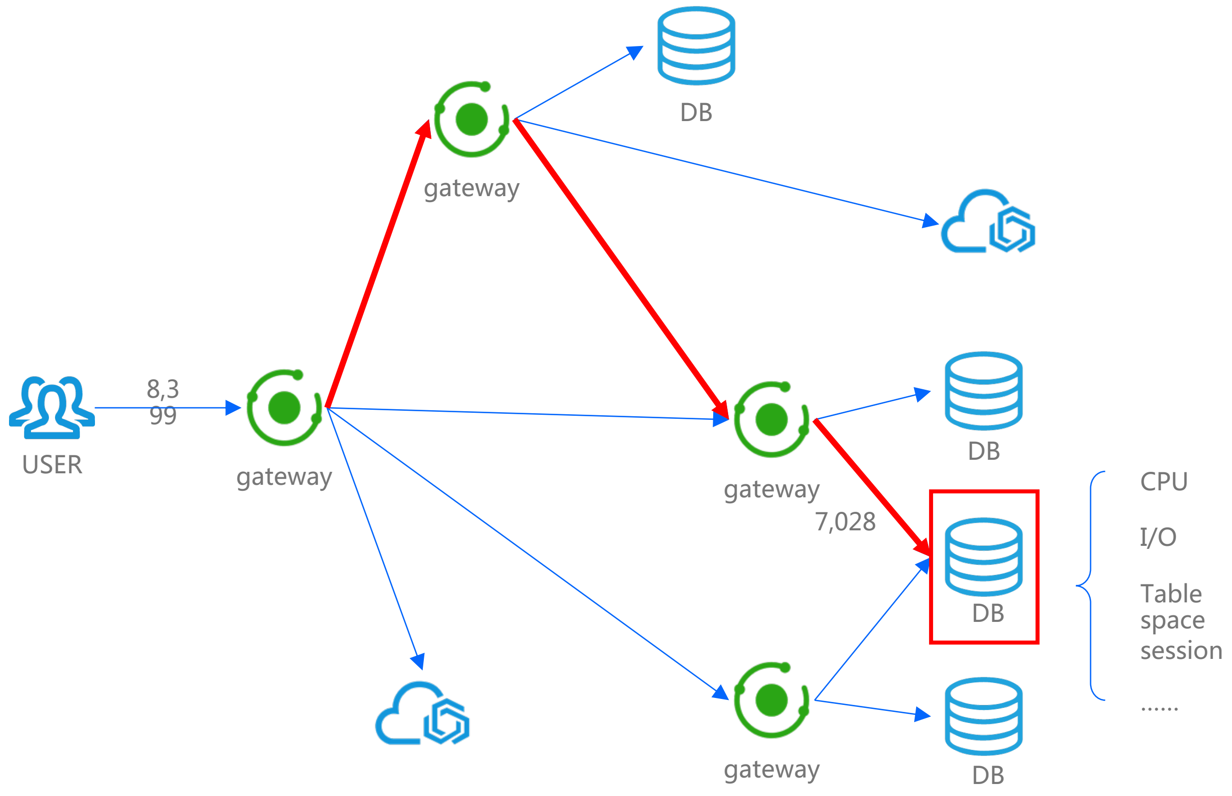
```
2018-10-10 20:53:51,194 [JAgentSocketServer.cpp:121] WARN  agent 9995 - Listening Port : 20510
2018-10-10 20:53:51,194 [RequestHandlerService.cpp:189] WARN  agent 9995 - RequestHandlerService::handle_input(ACE_HANDLE=38)
2018-10-10 20:53:51,195 [ResponseCOUNT.cpp:159] INFO  agent 9995 - IO: Command (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
```

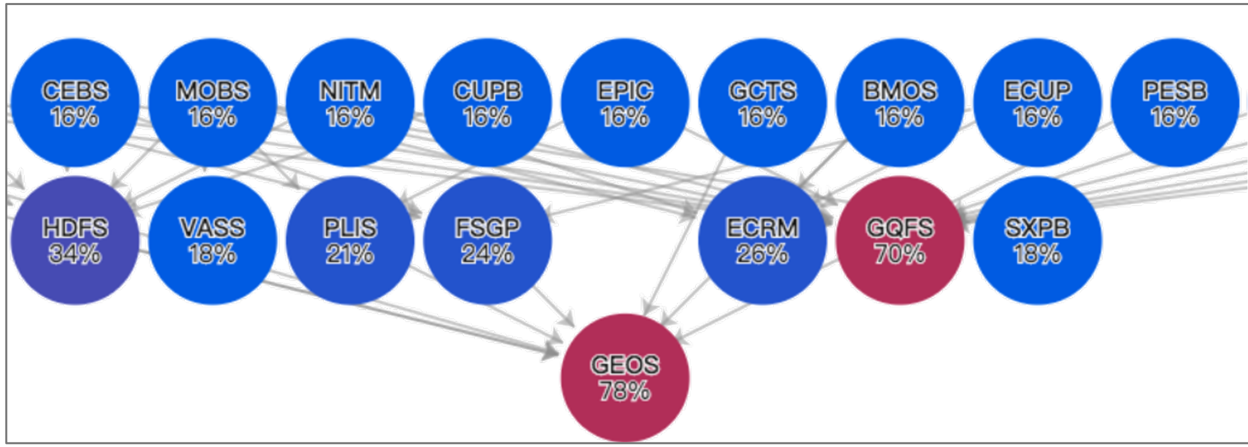
```
  INFO [WebContainer : 15] - queryForList:IDA_TEMPLATE.LISTDATA_MOST_CLICK
  INFO [WebContainer : 8] - queryForList:IDA_NOTICE.LISTDATA_BY_USER
 com.teradata.ida.auth.dto.SysUserVO@2c3d3e1d
[8/10/18 8:29:31:581 CST] 00000032 SystemOut     O  INFO [WebContainer : 1] - queryForList:IDA_TEMPLATE_AUTH.findTemplateByRoleId
 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.
```

# 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



# 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)

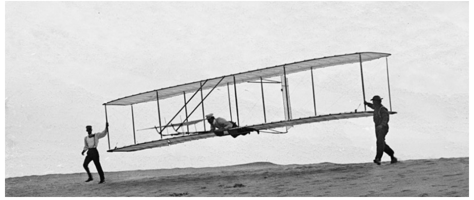
Traffic dump

Social Media

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

- Volume
- Velocity
- Variety
- Value

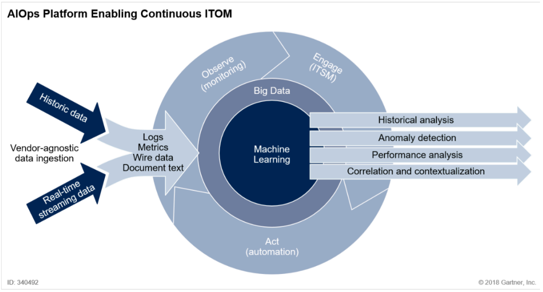
# Towards Autonomous IT Operations



Manual and few data

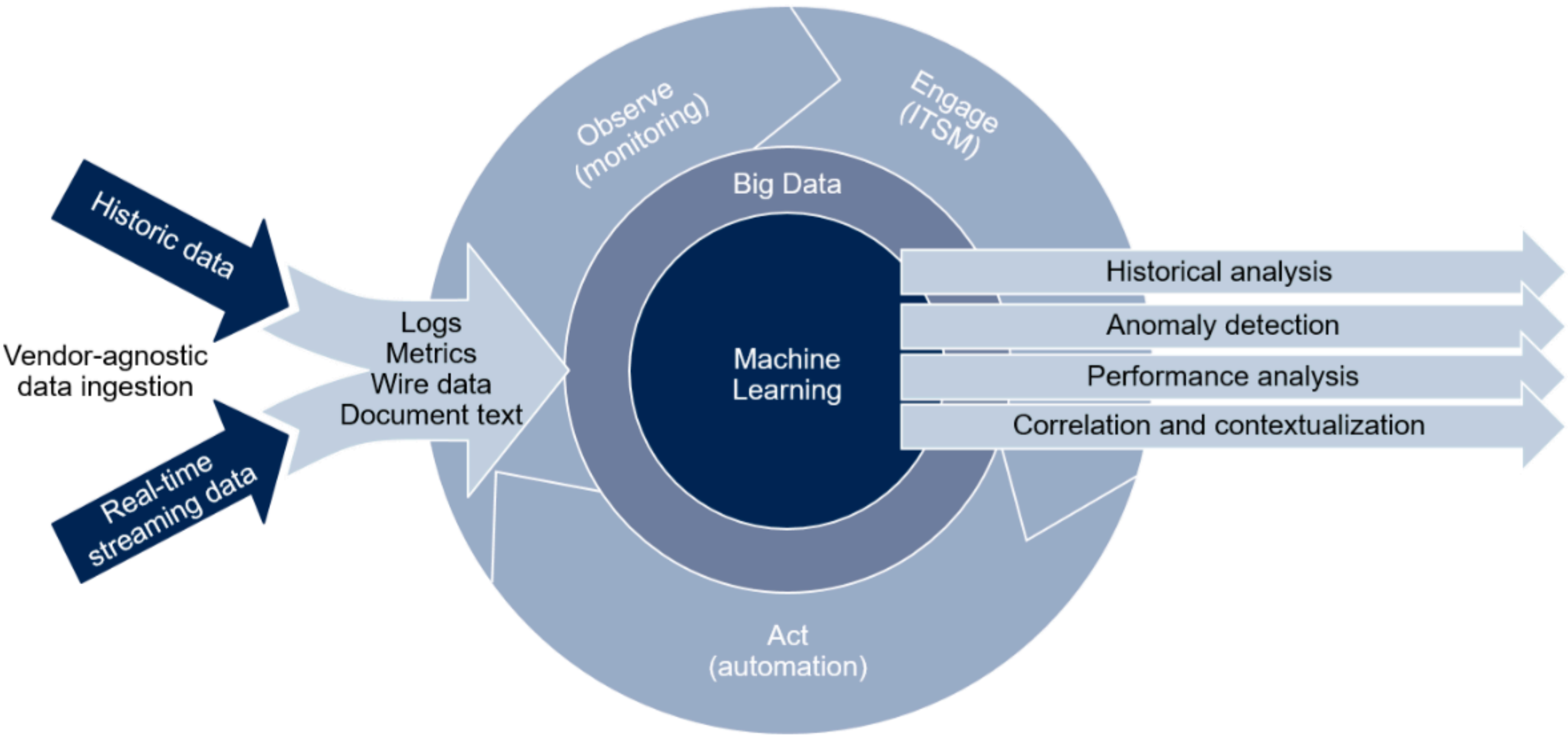
Lots of data but manual decision


Autonomous



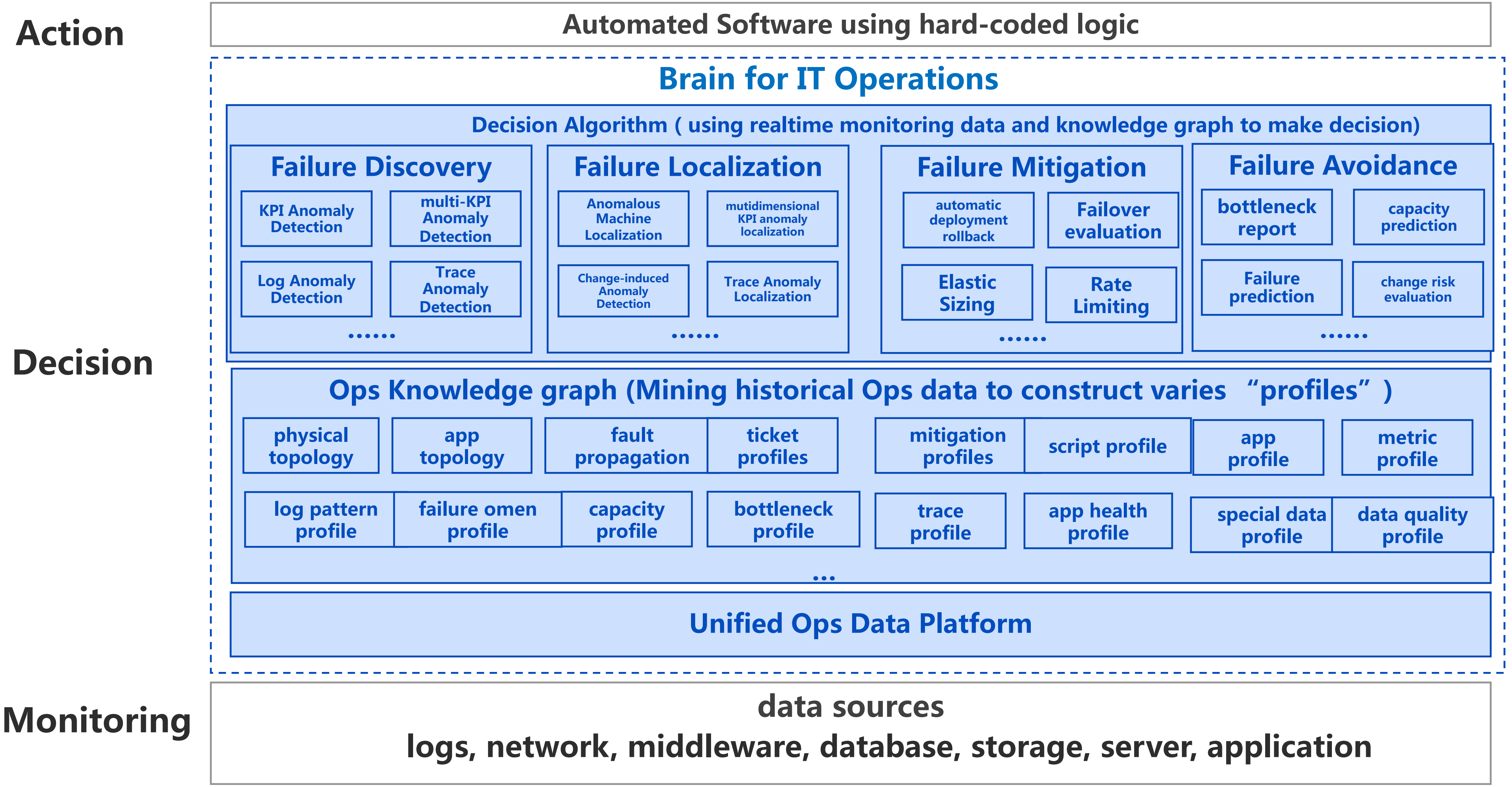

Spaceship Avalon: 5000 passengers and 258 crew members in hibernation. Flying towards Planet Homestead II, 120-year trip.

## AIOps Platform Enabling Continuous ITOM




ID: 340492

# Brain for IT Operations



# Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- *Brief Case Studies*
  - Impact assessment of software changes (SST, Causal Analysis)
  - Anomaly localization for multi-attribute time series (MCTS)
  - ***Data center switch failure prediction (Random Forest*)**
  - Web performance bottleneck identification (Decision Trees)
- Unsupervised Anomaly Detection in Ops
- Lessons Learned

# All case studies are from joint work with Industry Collaborators

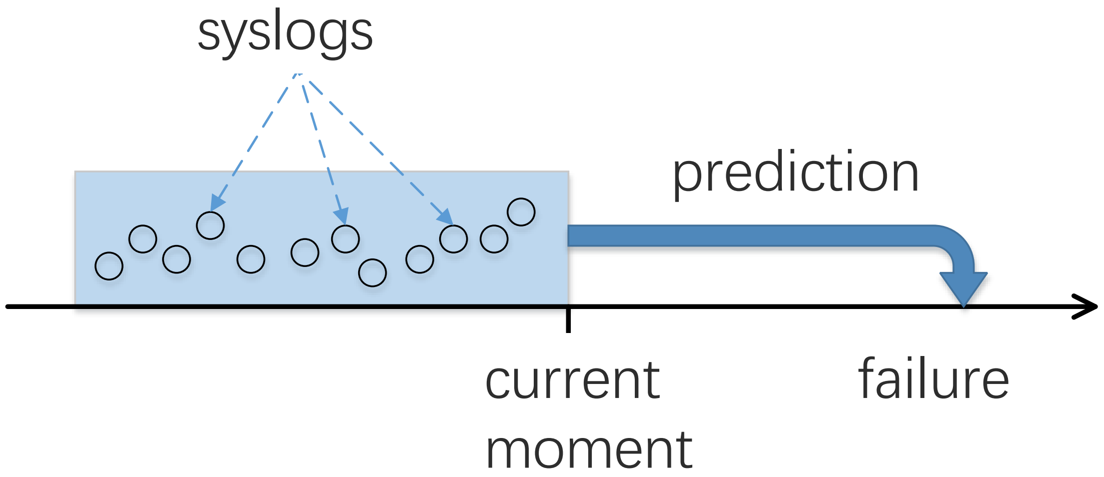
# Data Center Switch Failure Prediction->Preventive Replacement

Problem: Baidu-customized switches intermittently drop/delay packets, causing performance degrade at the application layer.

Reboot the switch stops the problem for some while.

Question: Can we  predict the this problem 2 hours before it happens again? Then just switch the traffic away from this switch using load balancer and reboot it.

Our solution PreFix: Features that capture omen log sequence + Random Forest.




- Precision: 82.15%
- Recall: 74.74%
- FPR: $3.75 \times 10^{-5}$

Joint work with Baidu. Published in SIGMETRICS 2018

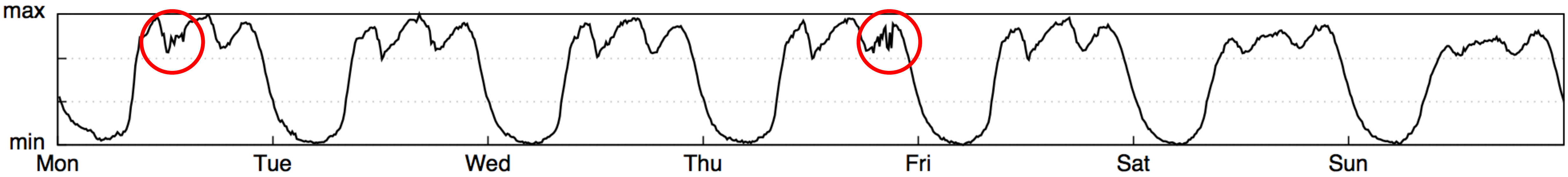
# Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Brief Case Studies
- Unsupervised Anomaly Detection in Ops
  - *Univariate time series anomaly detection* (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019)
  - Multivariate time series anomaly detection (KDD 2019)
  - Log anomaly detection (IWQoS 2017, IJCAI 2019)
  - Zero-day attack detection
- Lessons Learned

# Unsupervised Anomaly Detection

- Rule-based (e.g. static threshold, regular expression) anomaly detection does not work
- Labels are in general not available
  - Have to be labeled by experts, thus cannot be crowdsourced
  - Experts are unwilling to label, even though they are the users of the tool
- Common idea: somehow capture the "normal" patterns in the historical data (metrics, logs, HTTP requests), then any new data points that "deviate" from the normal patterns are considered "anomalous".

# Metrics (Univariate Time Series) Anomaly Detection




Page views (PV) of Baidu

**Metrics:** A set of performance measures that evaluate the service quality or entity status

**Metric anomalous (unexpected) behaviors** → Potential failures, bugs, attacks…

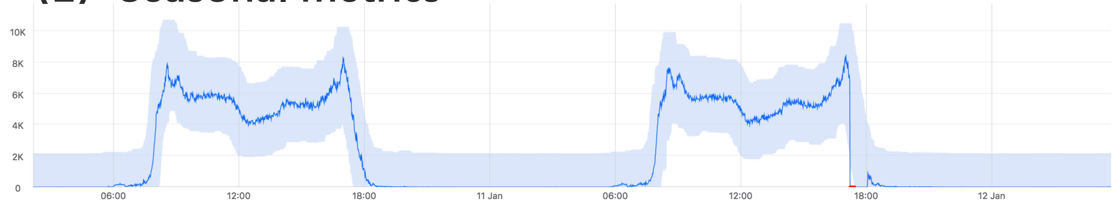
**Anomaly detection matters:** Find anomalous behaviors of the metric curve

→ Diagnose and fix it

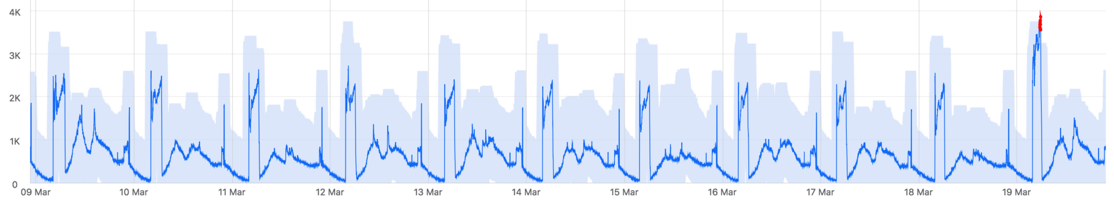
→ Avoid further influences and revenue losses

# Diverse Metrics and Their Diverse Anomalies

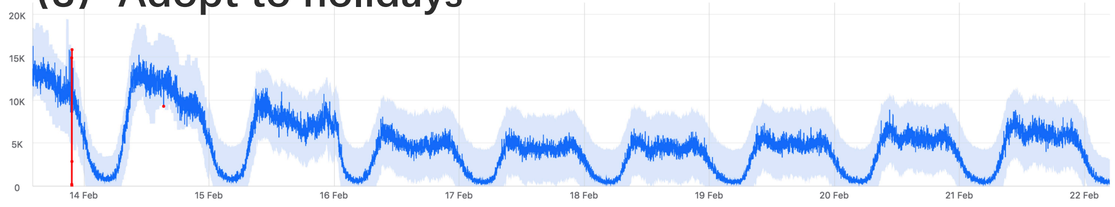
## (1) Seasonal metrics



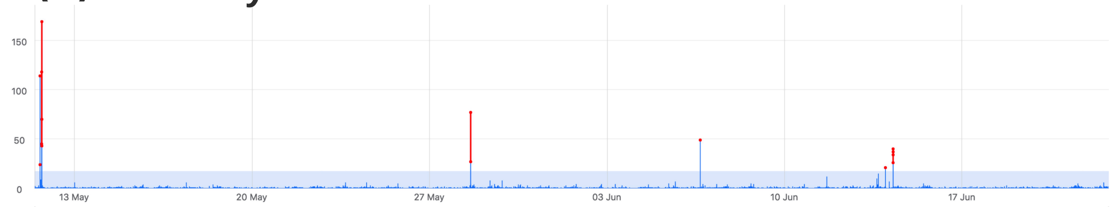

## (2) Periodicity shift



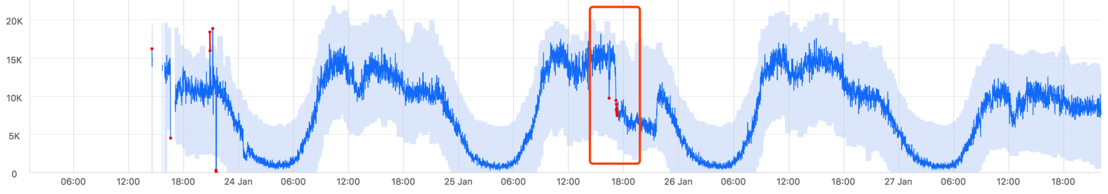

## (3) Adopt to holidays



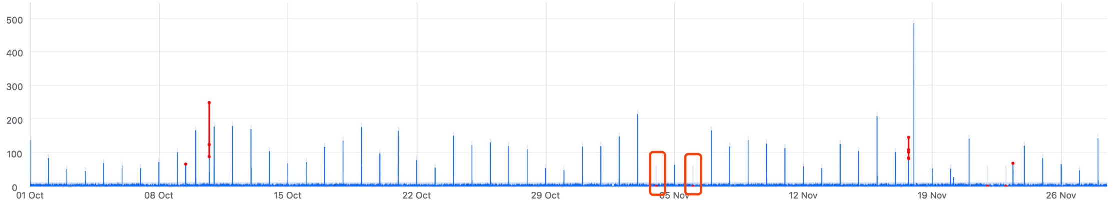

## (4) Identify variable metrics and obtain extreme threshold



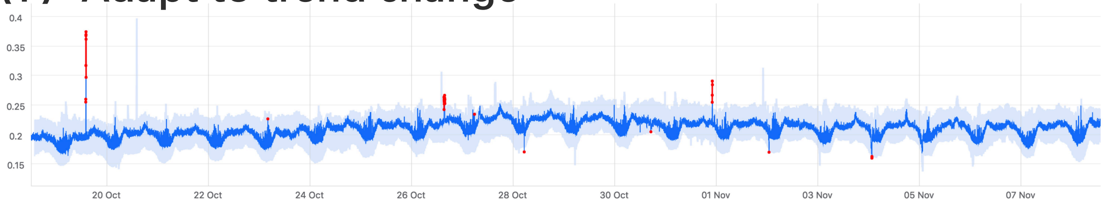

## (5) Detect too rapid a change



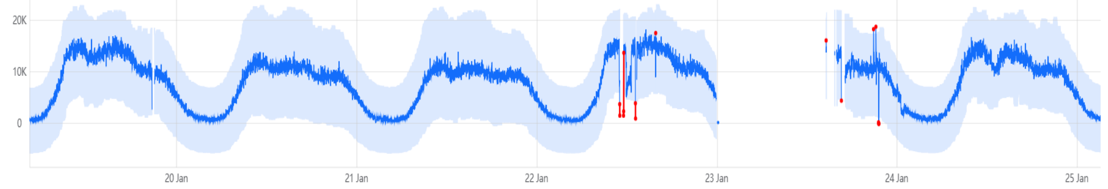

## (6) Detect the lack of seasonality.



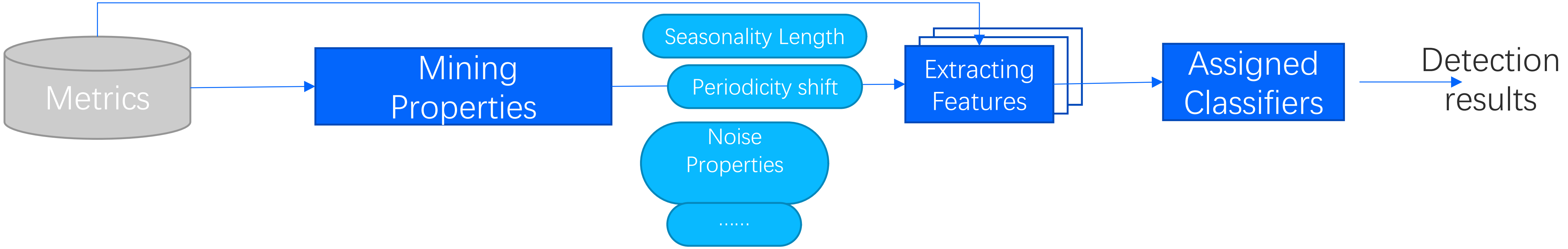

## (7) Adapt to trend change

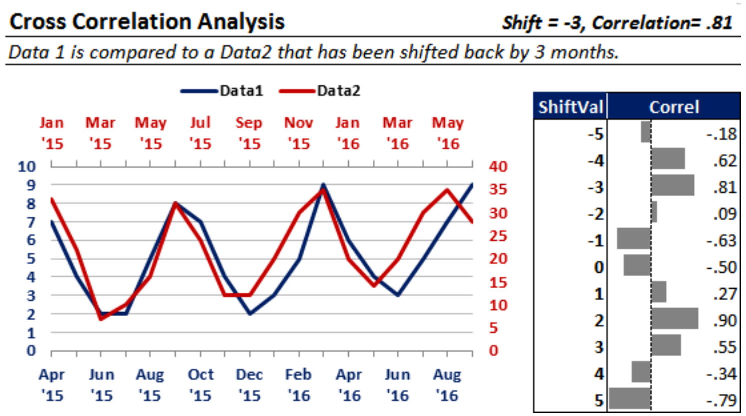


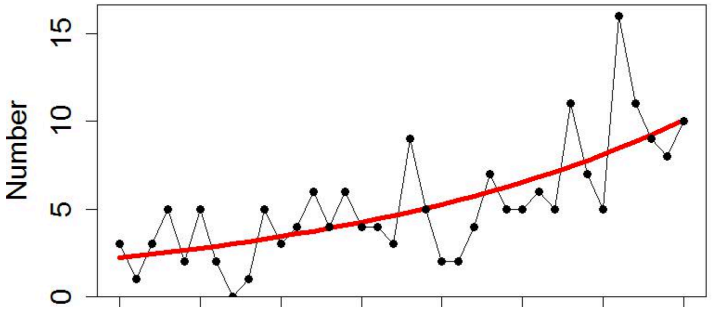

## (8) Robust against data loss or interruption



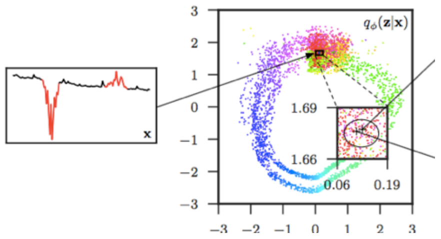
# Profiling metrics and then assign appropriate algorithms



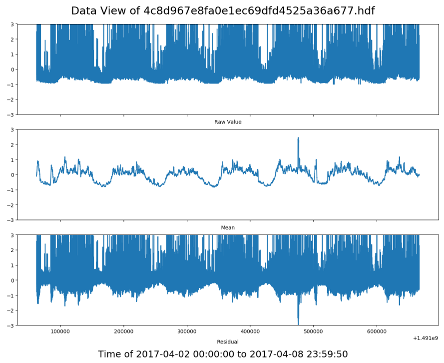



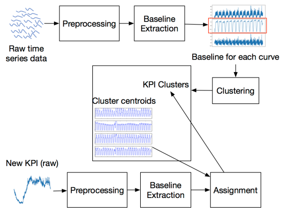

**Donut: WWW2018 for smooth time series with Gaussian noises**



**Buzz: INFOCOM 2019 when noises are non-Gaussian**




**ROCKA: use cluster centroid's trained model IWQOS 2018**



# Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications

Haowen Xu[1] Wenxiao Chen[1] Nengwen Zhao[1] Zeyan Li[1]
Jiahao Bu[1] Zhihan Li[1] Ying Liu[1] Youjian Zhao[1] Dan Pei[1]
Yang Feng[2] Jie Chen[2] Zhaogang Wang[2] Honglin Qiao[2]

[1]Tsinghua University

[2]Alibaba Group

April 26, 2018

# Existing Methods

- Statistical
  - Anomaly detectors based on traditional statistical models [INFOCOM2012]
- Supervised
  - Supervised ensemble learning with above detectors – Opprentice[IMC2015], EGADS [KDD2015]

# Donut: unsupervised anomaly detection assuming smooth time series

- A recent past of W data points at time t is called a window at time t. Donut tries to model the distribution of normal windows by VAE (Variational Auto Encoder) and find anomalies by likelihood.

  - The Variational Autoencoder model:
    - Kingma and Welling, *Auto-Encoding Variational Bayes*, *International Conference on Learning Representations (ICLR)* 2014.
    - Rezende, Mohamed and Wierstra, *Stochastic back-propagation and variational inference in deep latent Gaussian models*. ICML 2014.

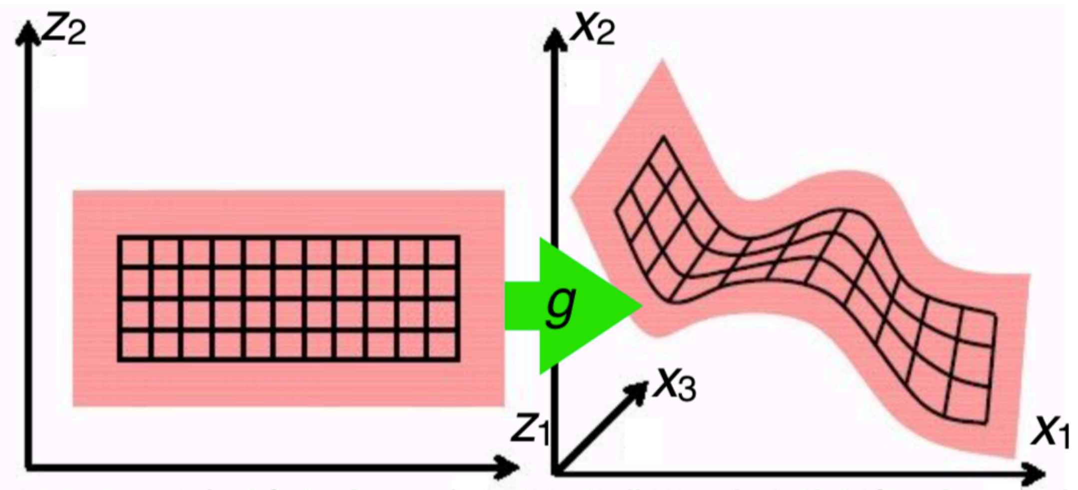
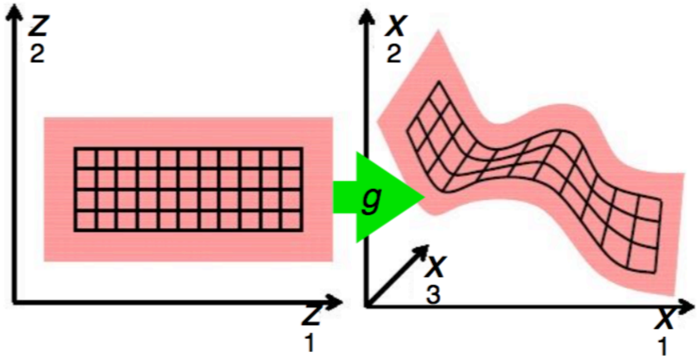
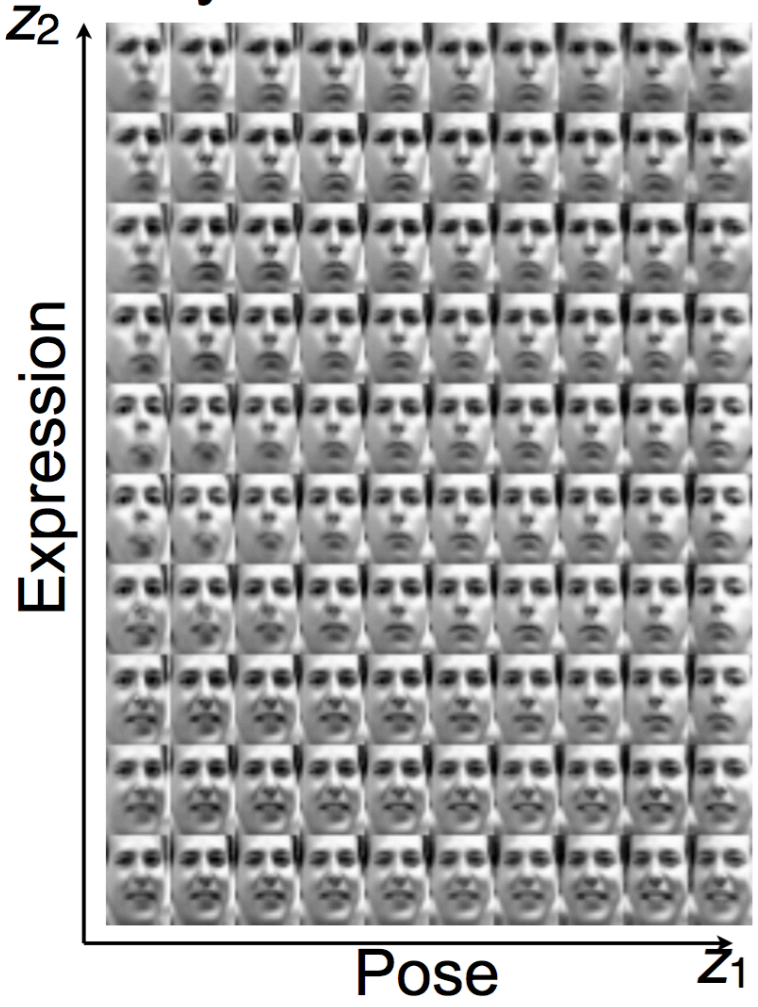



Image from: Ward, A. D., Hamarneh, G.: **3D Surface Parameterization Using Manifold Learning for Medial Shape Representation**, *Conference on Image Processing, Proc. of SPIE Medical Imaging*, 2007 7

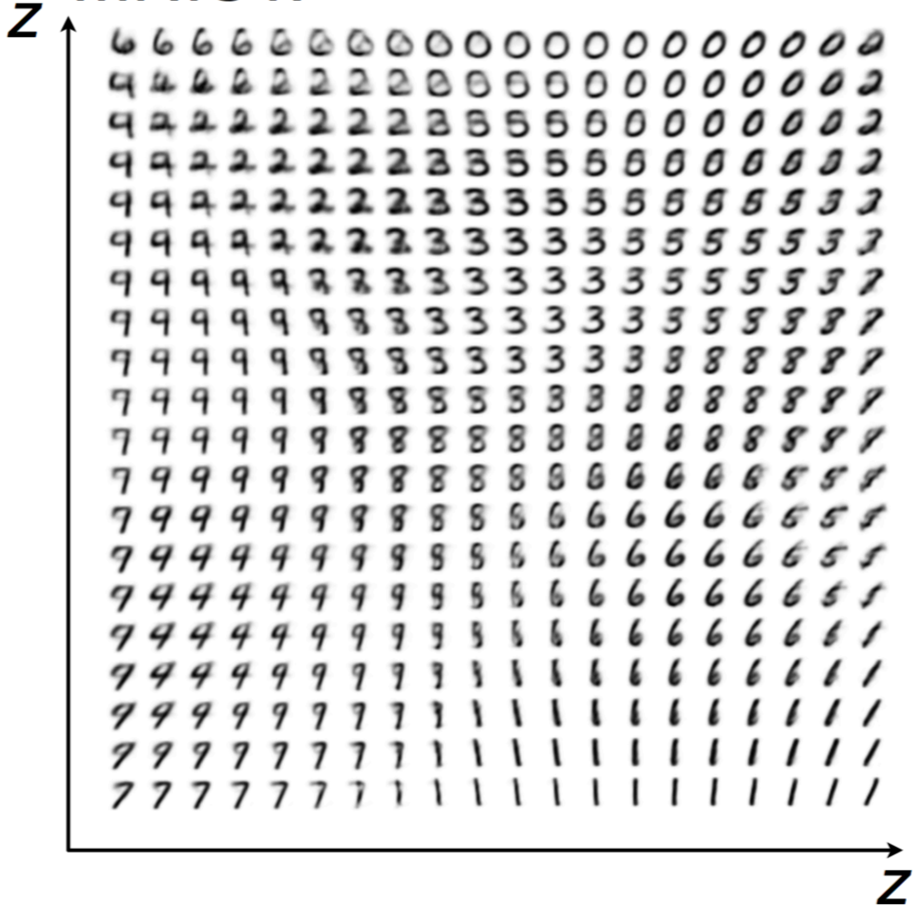
# Latent Variable Models



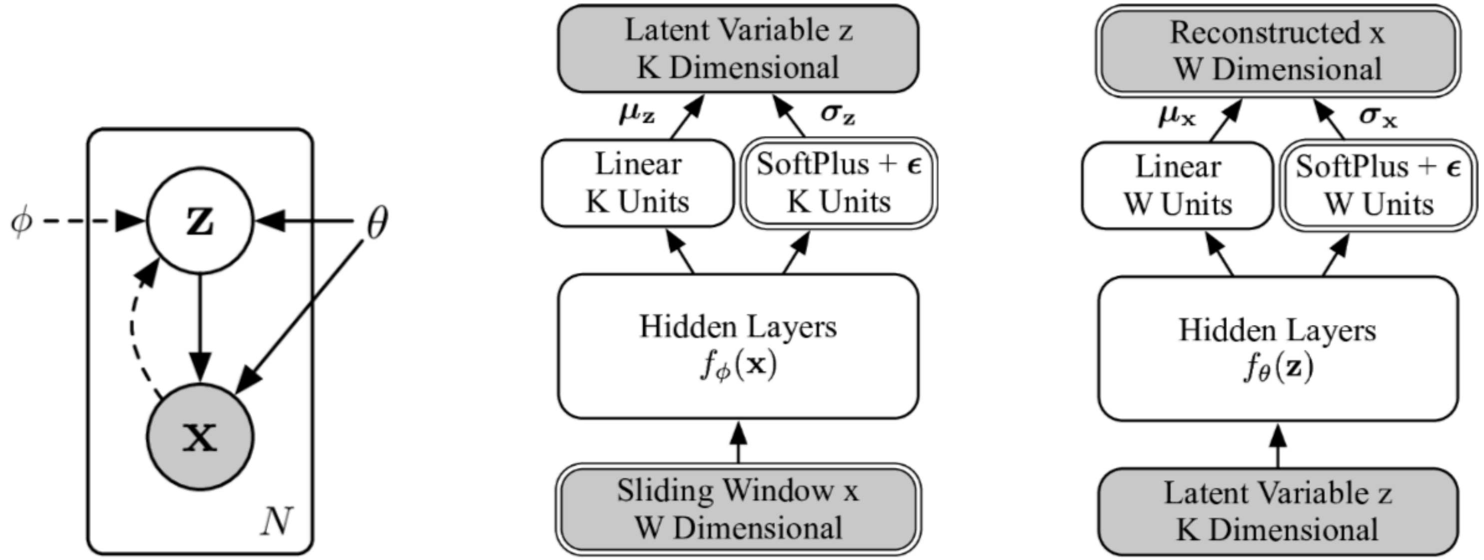

Frey Faces:




MNIST:



# Network Structure




(a) VAE General Structure (b) $q_\phi(\mathbf{z}|\mathbf{x})$ of *Donut* (c) $p_\theta(\mathbf{x}|\mathbf{z})$ of *Donut*

- Variational net: $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\sigma}_{\mathbf{z}}{}^2\mathbf{I})$.
- Generative net: $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}{}^2\mathbf{I})$.
- SoftPlus Trick: $\boldsymbol{\sigma}_{\mathbf{z}} = \mathrm{SoftPlus}[\mathbf{W}_{\boldsymbol{\sigma}_{\mathbf{z}}}^\top f_\phi(\mathbf{x}) + \mathbf{b}_{\boldsymbol{\sigma}_{\mathbf{z}}}] + \boldsymbol{\epsilon}$, $\mathrm{SoftPlus}[a] = \log[\exp(a) + 1]$. Similar for $\boldsymbol{\sigma}_{\mathbf{x}}$. (otherwise, unbounded)

$$\mathcal{L}_{vae} = \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z})\right]\right]$$

# 3D Visualization of the Latent Space

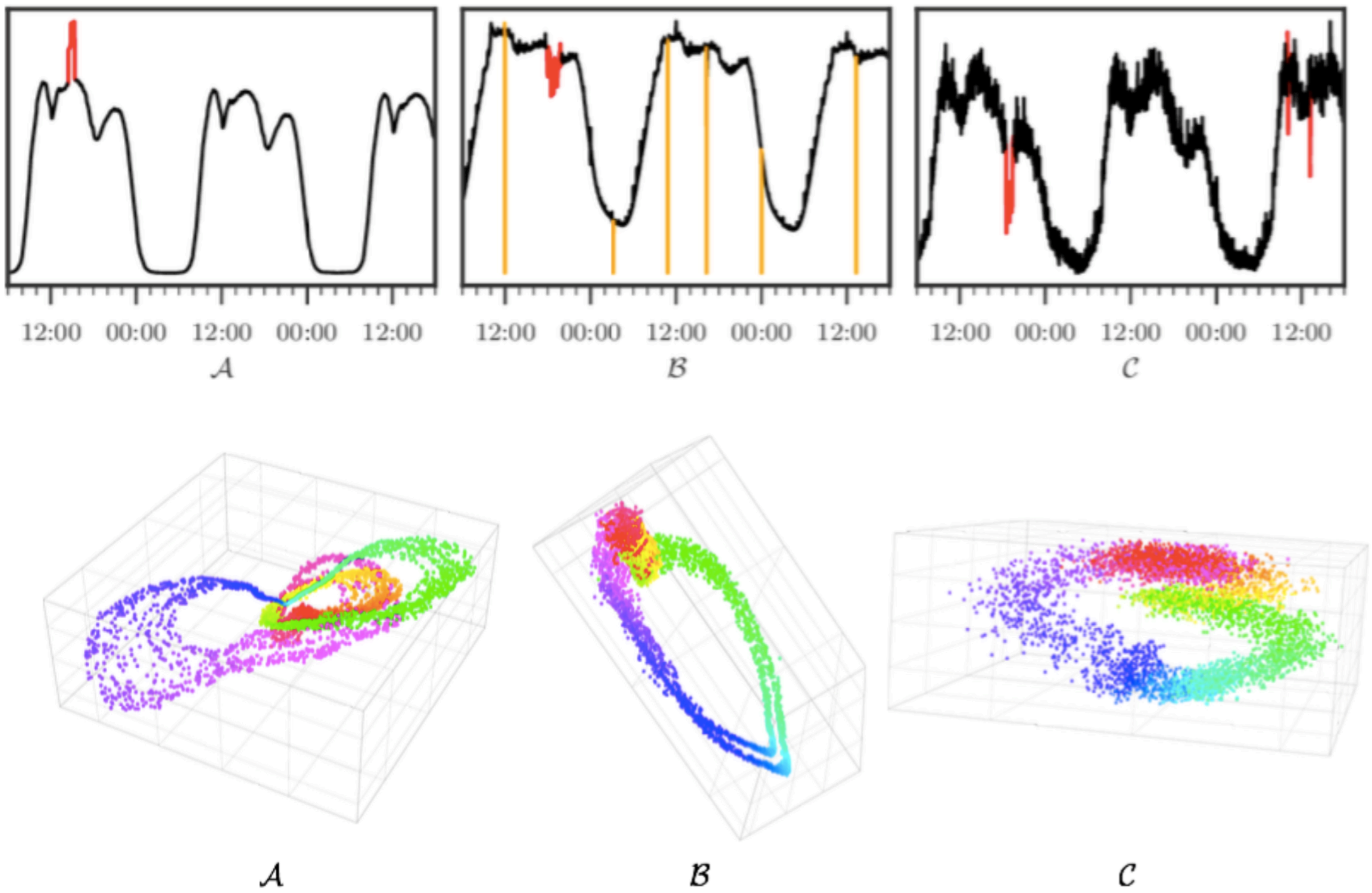
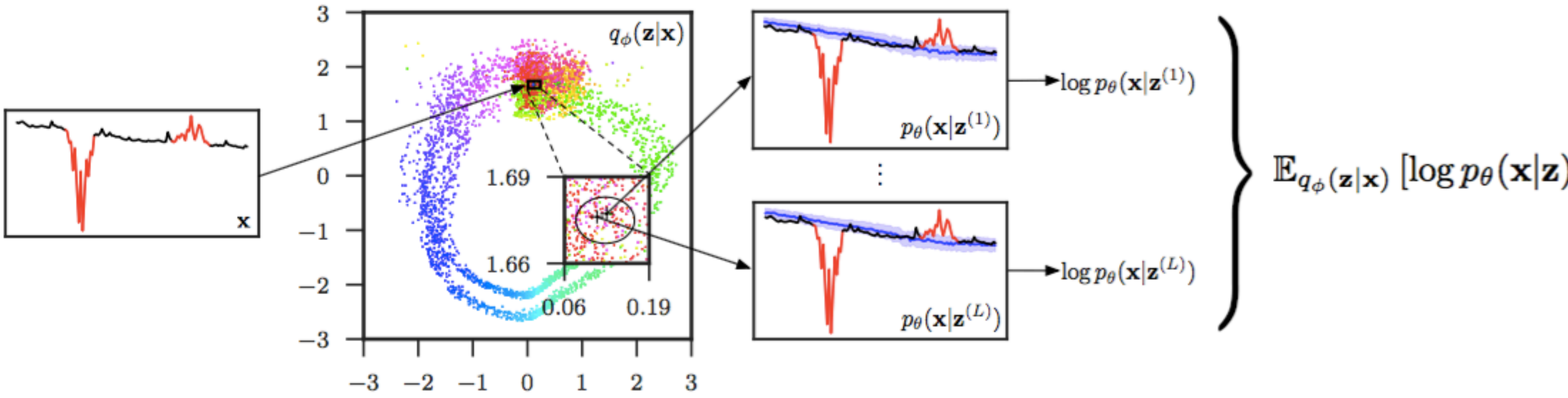
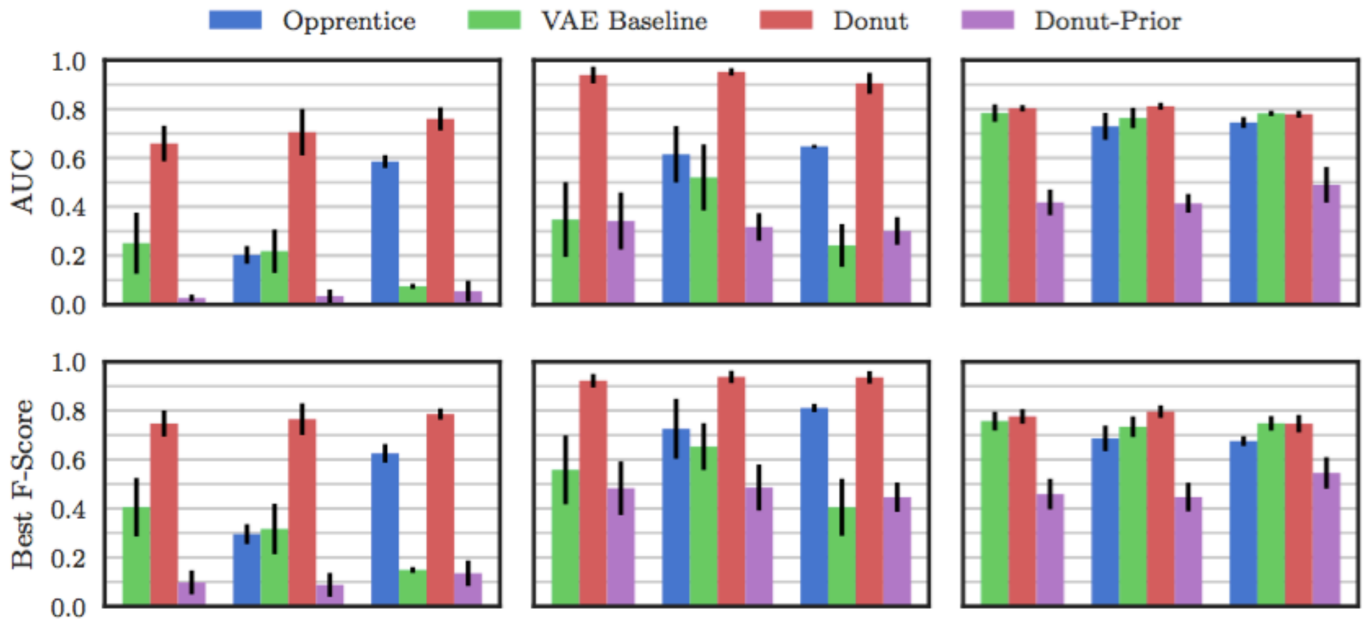



**Figure 12: 3-d latent space of all three datasets.**



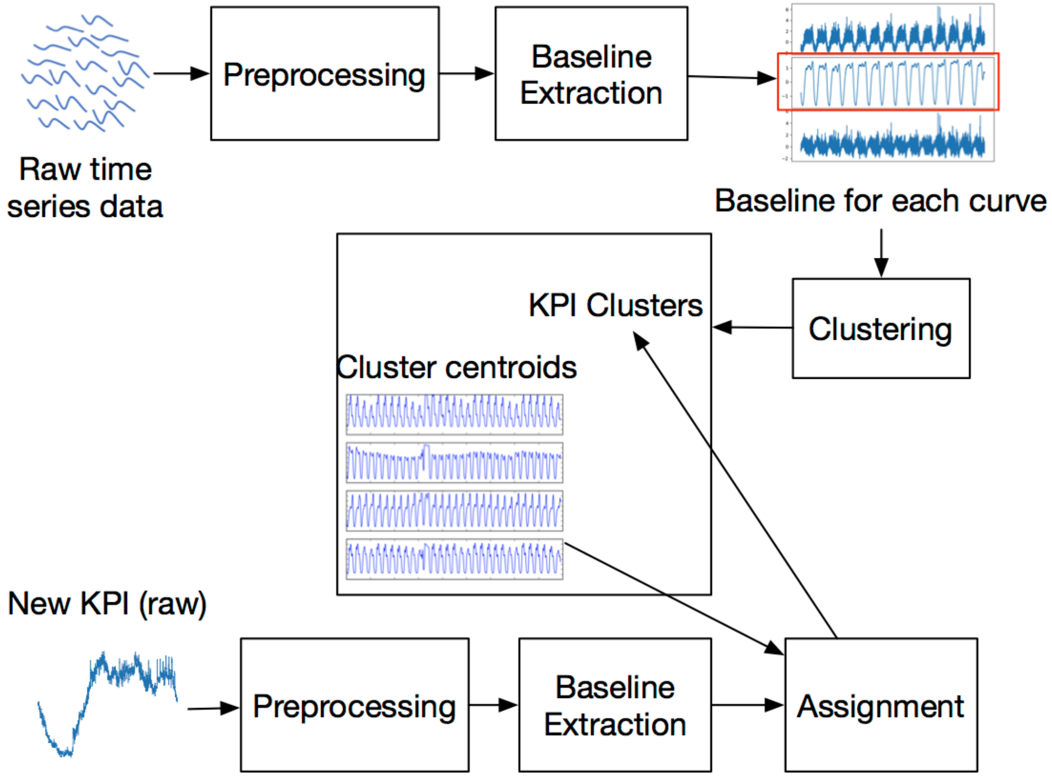




"Unsupervised KPI Anomaly Detection Through Variational Auto-Encoder"

Joint work with Alibaba, published in WWW 2018

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

# Clustering + Transfer Learning to reduce training overhead




IWQoS 2018

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

# Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Brief Case Studies
- Unsupervised Anomaly Detection in Ops
  - Univariate time series anomaly detection (IMC 2015, WWW 2018, IWQoS 2019, INFOCOM 2019a, INFOCOM2019b, ISSRE 2018, IPCCC 2018a, IPCCC 2018b, TSNM 2019)
  - Multivariate time series anomaly detection (KDD 2019)
  - Log anomaly detection (IWQoS 2017, IJCAI 2019)
  - *Zero-day attack detection (INFOCOM 2019)*
- Lessons Learned

# ZeroWall: Detecting Zero-Day Web Attacks through Encoder-Decoder Recurrent Neural Networks

Ruming Tang*, Zheng Yang*, Zeyan Li*, Weibin Meng*, Haixin Wang+,
Qi Li*, Yongqian Sun#, Dan Pei*, Tao Wei^, Yanfei Xu^ and Yan Liu^

*Tsinghua University, China

+University of Science and Technology Beijing, China

#Nankai University, China

^Baidu

IEEE International Conference on Computer Communications, 27-30 April 2020 // Beijing, China

# WAFs Do Not Capture Zero-Days

- **WAFs** (**W**eb **A**pplication **F**irewalls) are **wildly deployed** in industry, however, such **signature-based** methods are not suitable to detect zero-day attacks.
- Zero-day attacks in general are hard to detect and zero-day Web attacks are particularly challenging because:
  1. have **not been previously seen**
     → most **supervised** approaches are inappropriate
  2. can be carried out by a **single** malicious HTTP
     → **contextual** information is not helpful
  3. very **rare** within a large number of Web req
     → **collective** and **statistical** information are not effe

**ZeroWall**

An **unsupervised** approach, which can **work with an existing WAF in pipeline**, to effectively detecting a zero-day Web attack hidden in **an individual Web request**.

# What We Want

- 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**.

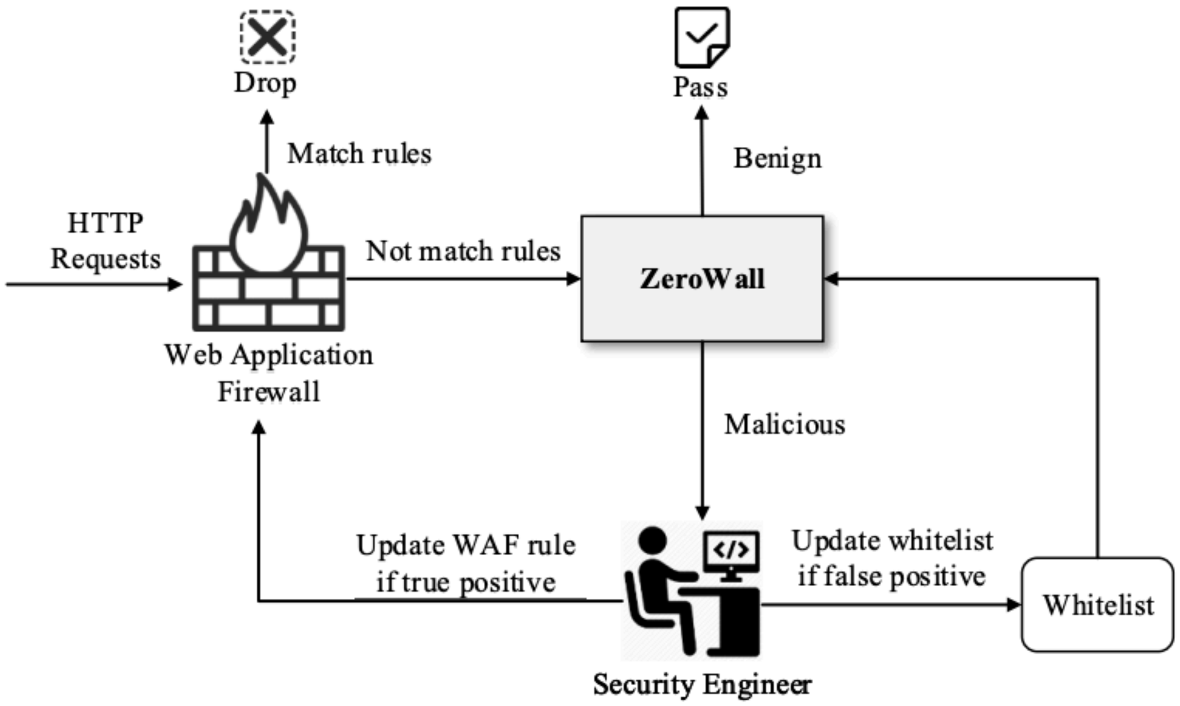
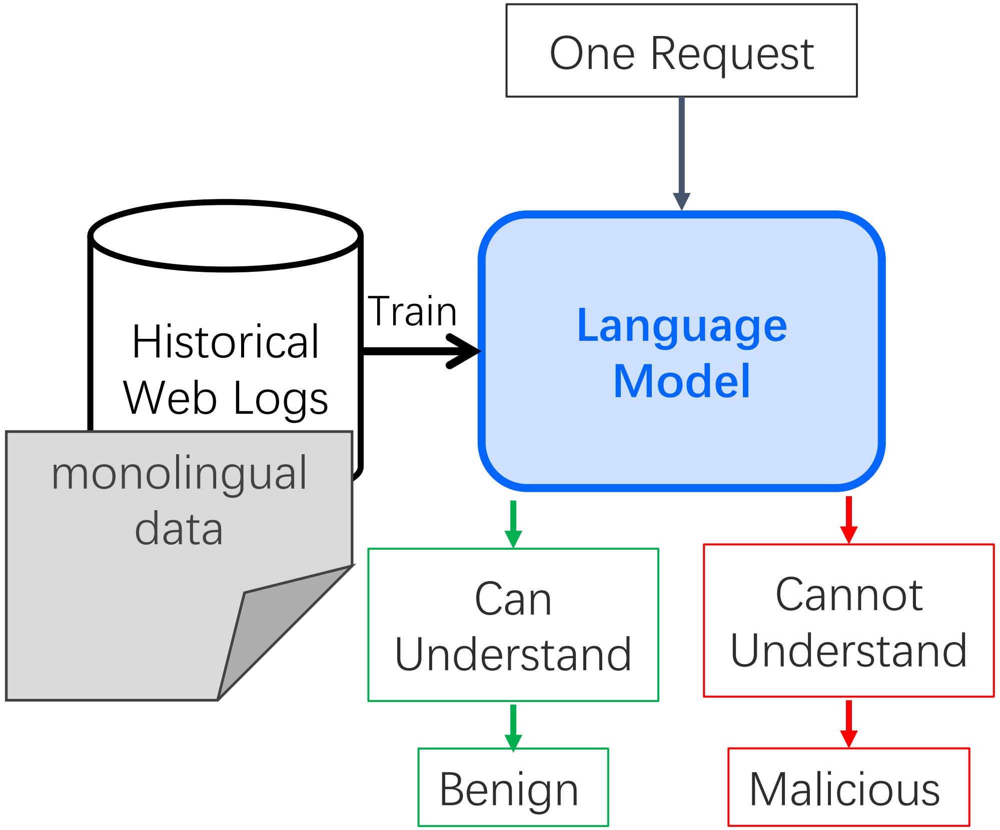



Figure 1: The workflow of *ZeroWall*.

# Idea

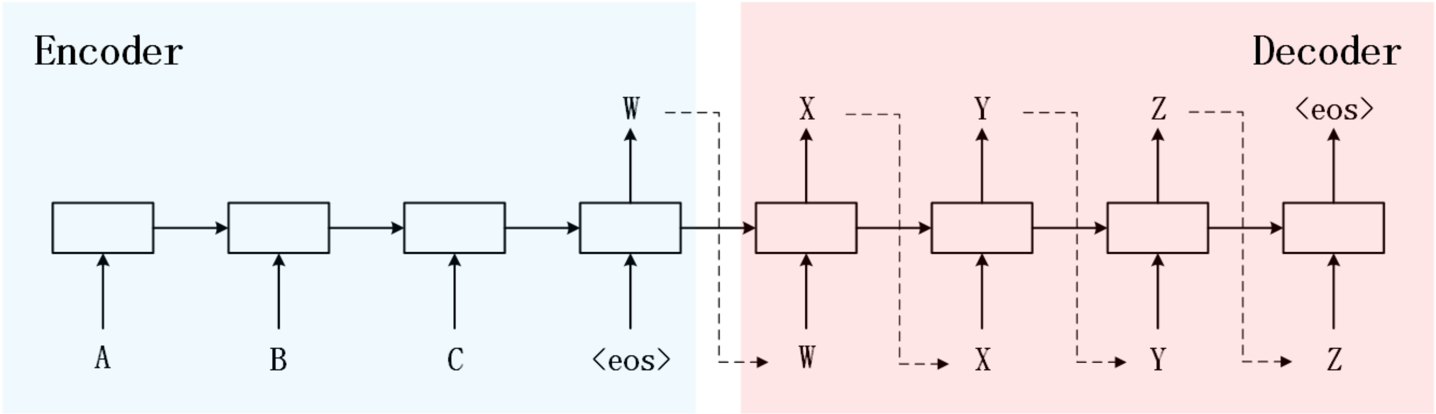
- HTTP request is a **string following HTTP**, and we can consider an HTTP request as one **sentence** in the ***HTTP request language***.
- **Most** requests are **benign**, and **malicious** requests are **rare**.
- Thus, we train a kind of **language model** based on historical logs, to **learn this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



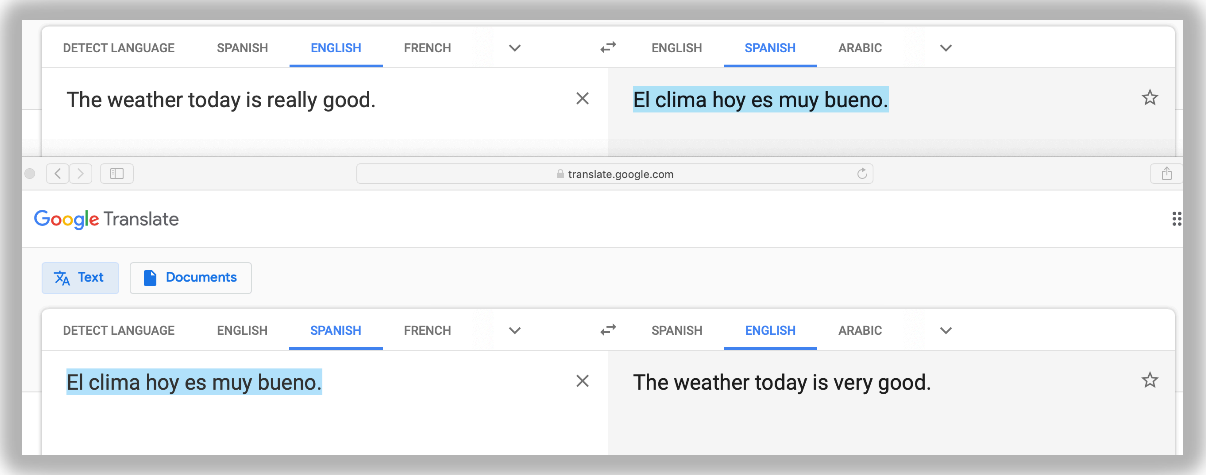
# Self-Translate Machine

- How to learn this "**Hyper-TEXT**" language?

- Use **Neural Machine Translation** model to train a **Self-Translate Machine**
  - **Encode** the original request into one *representation*
  - Then **Decode** it back

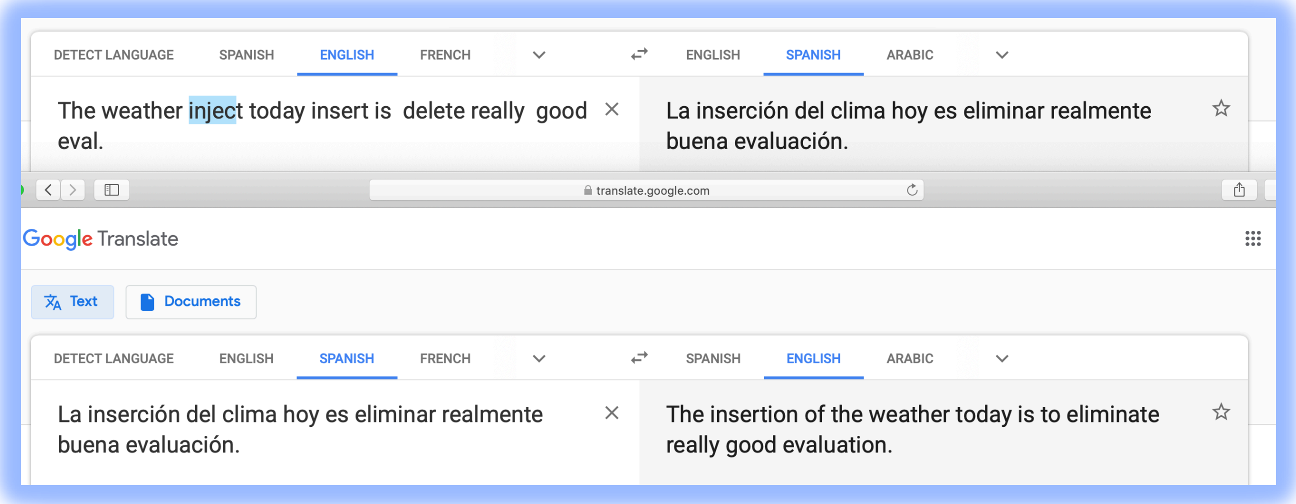


# Self-Translate Machine



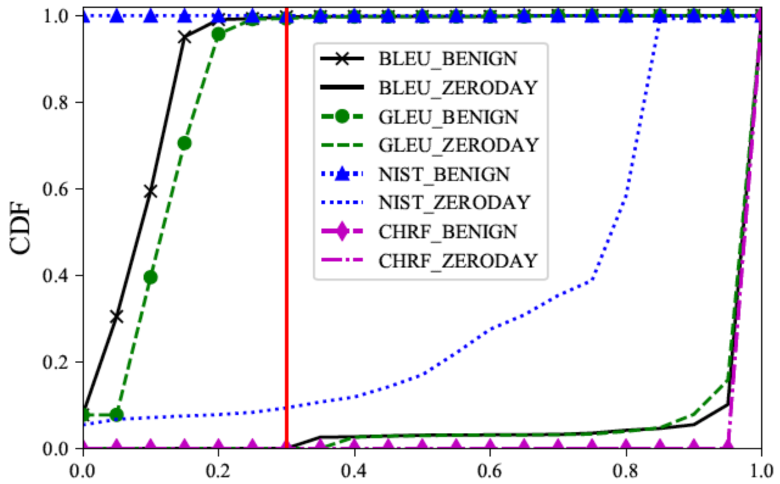

Self-translation works **well** for **normal** sentences

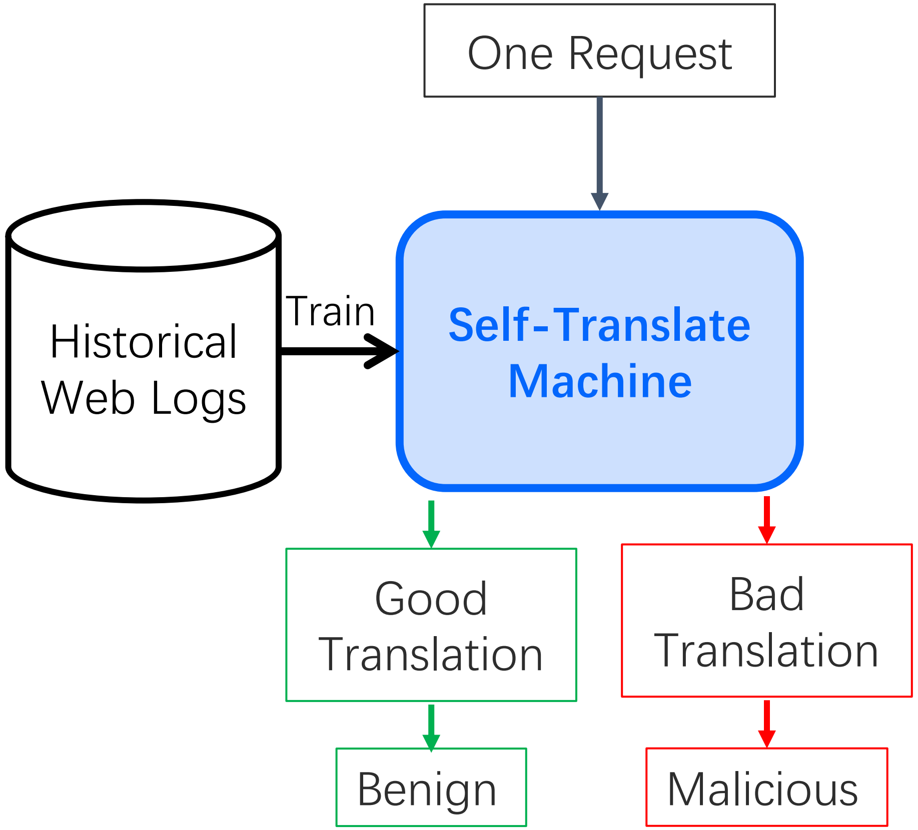
Output **deviates** significantly from the input, when the input is a sentence **not previously seen** in the training dataset of the self-translation models.



# Self-Translate Machine

- **Translation Quality → Anomaly Score**
- How to quantify the self-translation quality (anomaly score)?
  → Use **machine translation metrics**




**An attack detection problem → A machine translation quality assessment problem**

# Self-Translated Sequence

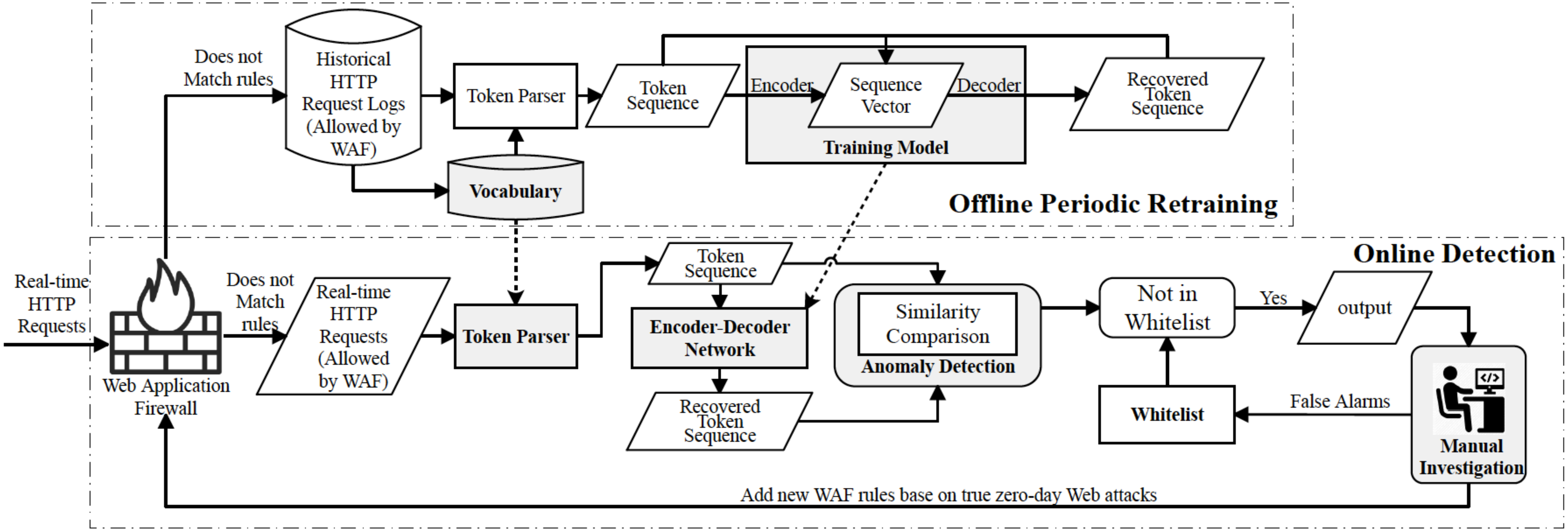
- **Translation Quality → Anomaly Score**
  - → Use **BLEU** as an example
  - → Malicious Score $= 1 - BLEU\_Score$

| | | | |
|---|---|---|---|
| **Original Request** | POST http://localhost:8080/tienda1/publico/autenticar.jsp modo=entrar&login=caria&pwd=egipciaca&remember=off&B1=Entrar | | |
| **Tokenized** | tienda1 publico autenticar jsp modo entrar login _OTHER_ pwd _OTHER_ remember off b1 entrar | | |
| **Translated** | tienda1 publico autenticar jsp modo entrar login _OTHER_ pwd _OTHER_ remember on b1 entrar | | |
| **BLEU** | 0.8091 | **Malicious Score** | 0.1909 |

| | | | |
|---|---|---|---|
| **Original Request** | POST http://m.thepaper.cn/admin_UploadDataHandler.ashx ------WebKitFormBoundaryRvkd1dbq3x1OJhUH\x0D\x0AContent-Disposition: form-data; name=\x22uploadify\x22; filename=\x2220170215180046.jpg\x22\x0D\x0A *Content-Type: image/jpeg*\x0D\x0A\x0D\x0A **<%eval request(\x22T\x22)%>**\x0D\x0A------WebKitFormBoundaryRvkd1dbq3x1OJhUH\x0D\x0AContent-Disposition: form-data; name=\x22saveFile\x22\x0D\x0A\x0D\x0At.asp\x0D\x0A------WebKitFormBoundaryRvkd1dbq3x1OJhUH\x0D\x0AContent-Disposition: form-data; name=\x22Upload\x22\x0D\x0A\x0D\x0ASubmit Query\x0D\x0A------WebKitFormBoundaryRvkd1dbq3x1OJhUH-- | | |
| **Tokenized** | _OTHER_ ashx _OTHER_ content disposition form data name uploadify filename _pnum_0_ jpg content type image jpeg eval request onechr _OTHER_ content disposition form data name _OTHER_ onechr asp _OTHER_ content disposition form data name upload submit query _OTHER_ | | |
| **Translated** | _OTHER_ _OTHER_ do php _OTHER_ eval get_magic_quotes_gpc stripslashes _post chr _pnum_0_ chr _pnum_1_ _post chr _pnum_2_ chr _pnum_3+_ z0 _pnum_3+_ ini_set display_errors _pnum_3+_ set_time_limit _pnum_3+_ set_magic_quotes_runtime _pnum_3+_ echo onechr dirname _server script_filename if onechr onechr dirname _server path_translated | | |
| **BLEU** | 0 | **Malicious Score** | 1.0 |

An attack detection problem → A machine translation quality assessment problem

# ZeroWall Workflow




- Offline Periodic Retraining
  - Build and update **vocabulary** and re-train the **model**
- Online Detection
  - Detect **anomalies** in real-time requests for **manual investigation**

# Real-World Deployment

- Data Trace:
  - 8 real world trace from an Internet company.
  - Over 1.4 billion requests in a week.
- Overview
  - Captured 28 different types of zero-day attacks, which contribute to 10K of zero-day attack requests in total.
  - False positives: 0~6 per day

| # | D-1 | D-2 | D-3 | D-4 | D-5 | D-6 | D-7 | D-8 | Total |
|---|---|---|---|---|---|---|---|---|---|
| **Malicious*** | 51839 | 186066 | 19515 | 53394 | 33724 | 2136811 | 42088623 | 90982519 | 135552491 |
| **Zero-Day** | 25 | 1118 | 283 | 4209 | 1188 | 2003 | 49011 | 83746 | 141583 |
| **Benign** | 1576235 | 3142793 | 13572827 | 15618518 | 31718124 | 177993528 | 528158912 | 534048878 | 1305829815 |
| **Total** | 1628099 | 3329977 | 13592625 | 15676121 | 31753036 | 180132342 | 570296546 | 625115143 | 1441523889 |
| **B2M(1)** | 30.4 | 16.9 | 695.5 | 292.5 | 940.5 | 83.3 | 12.5 | 5.9 | 9.6 |
| **B2Z(2)** | 63049.4 | 2811.1 | 47960.5 | 3710.7 | 26698.8 | 88863.5 | 10776.3 | 6377.0 | 9223.1 |

* Known malicious filtered by WAF. (1) Ratio of **B**enign to **M**alicious (in WAF); (2) Ratio of **B**enign to **Z**ero-Day

# A Zero-Day Case

- These attack is detected by **ZeroWall**, **CNN** and **RNN**.
- **WAF** are usually based on **keywords**, e.g., eval, request, select and execute.
- **ZeroWall** is based on the "**understanding**" of benign requests. The structure of this zero-day attack request is more like a programming language.

...
searchword=d&**order=}{end if}{if:1)print_r( $_POST[func]($_POST[cmd]));//} {end if}&func=assert&cmd=phpinfo();**

Token Sequence: search php searchtype _pnum_0_ _OTHER_ onechr order end if if _pnum_1_ _OTHER_ _post _OTHER_ _post cmd end if _OTHER_ assert cmd phpinfo

contains **none** of **WAF keywords**

| | |
|---|---|
| 1 | plus ad_js php aid _pnum_0_ onechr **assert** _pnum_1_ execute execute function bd byval onechr for onechr _pnum_2_ to len onechr step _pnum_3+_ onechr mid onechr _pnum_3+_ if isnumeric mid onechr _pnum_3+_ then execute bd bd chr onechr else execute bd bd chr onechr mid onechr _pnum_3+_ onechr _pnum_3+_ **end if** chr _pnum_3+_ next end function response write execute on error resume next bd _phex_0_ response write response end |
| 2 | preview php _OTHER_ php **assert** _OTHER_ onechr |
| 3 | lib _OTHER_ module inc php _OTHER_ eval _OTHER_ onechr class _OTHER_ onechr **phpinfo** |
| 4 | cms _OTHER_ uploads _OTHER_ php id **assert** _OTHER_ eval base64_decode **_post** z0 z0 _pbas_0_ |
| 5 | myship php cmd eval base64_decode **_post** z0 z0 _pbas_0_ |

# Summary

- Present a zero-day web attack detection system **ZeroWall**
  - **Augmenting** existing **signature-based WAFs**
  - Use **Encoder-Decoder Network** to learn patterns from normal requests
  - Use **Self-Translate Machine** & **BLEU Metric**

An **attack detection** problem → A **machine translation quality assessment** problem

- **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

**Thanks!**
**And Questions**

Ruming Tang: trm14@mails.tsinghua.edu.cn

# 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".

- Different approaches based on
  - Sequence Top-k Prediction (Sequential model such as LSTM/GRU)
  - Reconstruction Probability (encoder-decoder)
  - "Self-Translation" quality (sentence/request level detection)
  - …

- A combination of stochastic deep Bayesian model and deterministic RNN model can help.

- Latent variables help capture the stochasticity
  - Connection in latent space can help capture temporal dependency
  - Use flows to capture non-Gaussian distributions.

# Outline

- IT Operations (Ops) background
- Is machine learning necessary for Ops?
- Brief Case Studies
- Unsupervised Anomaly Detection in Ops
- *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

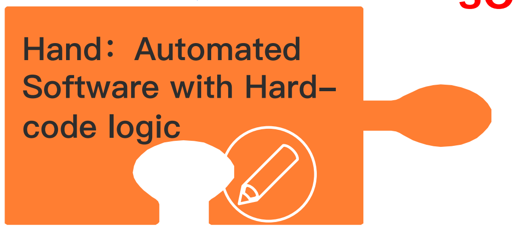
ARIMA, Time Series Decomposition, Holt-Winters, CUSUM, SST,DiD,DBSCAN, Pearson Correlation, J-Measure, Two-sample test, Apriori, FP-Growth, K-medoids, CLARIONS, Granger Causality, Logistic Regression, Correlation analysis (event-event, event-time series, time series-time series) , hierarchical clustering,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

# Lesson 1：Divide and Conquer instead of Using Black Box

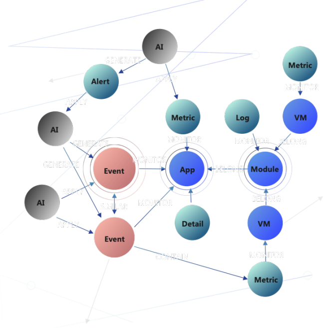
（1） Abundant data
(2)  Deterministic information
（3） Complete information
（4） Well defined
（5） Single domain

——*Prof. Bo Zhang, CAS Fellow*

These two types of modules must be solvable by existing ML algorithms

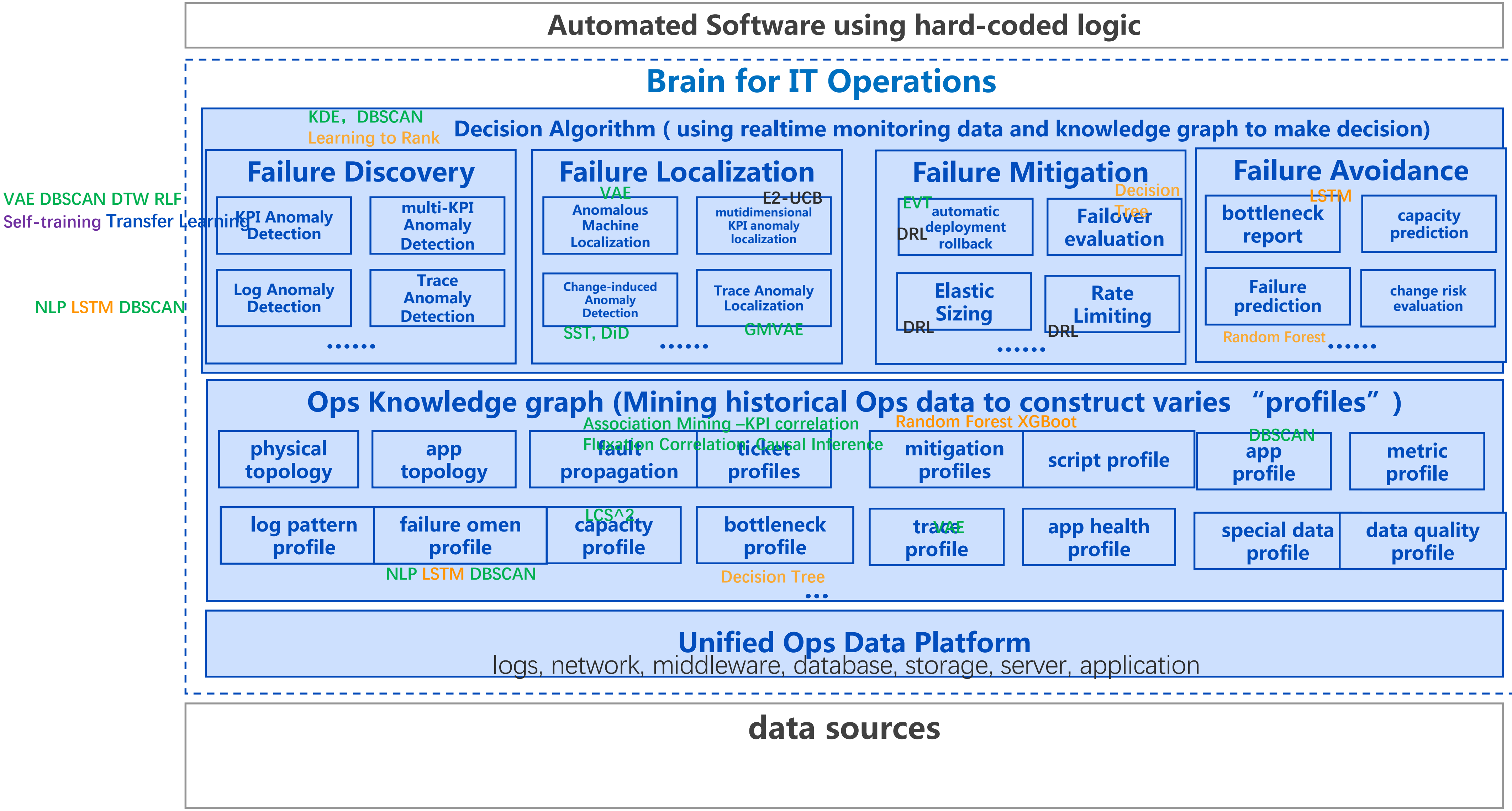

Brain: Knowledge Graph




Brain: Decision

# Various ML algorithms used in AIOps

Unsupervised Reinforcement Learning Supervised but with labels Semi-supervised Learning Transfer Learning



# Lesson 2: 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.

Univariate time series anomaly detection

IPCCC 2018

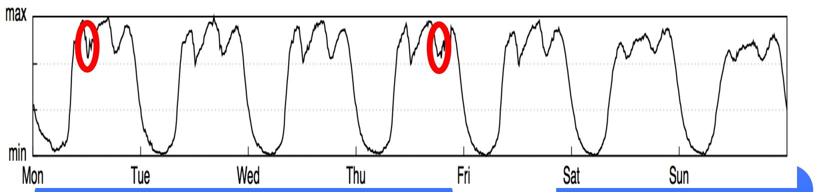
Conditional VAE to detect seasonality-violating anomalies

INFOCOM  2019

Adversarial  Training +VAE

IPCCC 2018

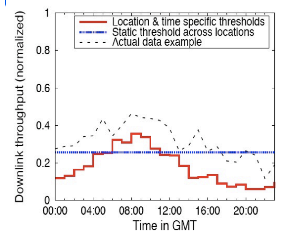
Semi-supervise learning for fast anomaly detection of new time




Statistical methods (manual algorithm selection and parameter-tuning)

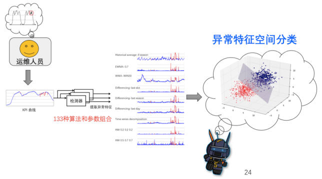
Supervised Ensemble learning

Unsup
Learni

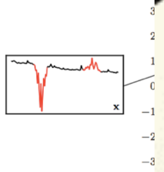



(a) Location 1

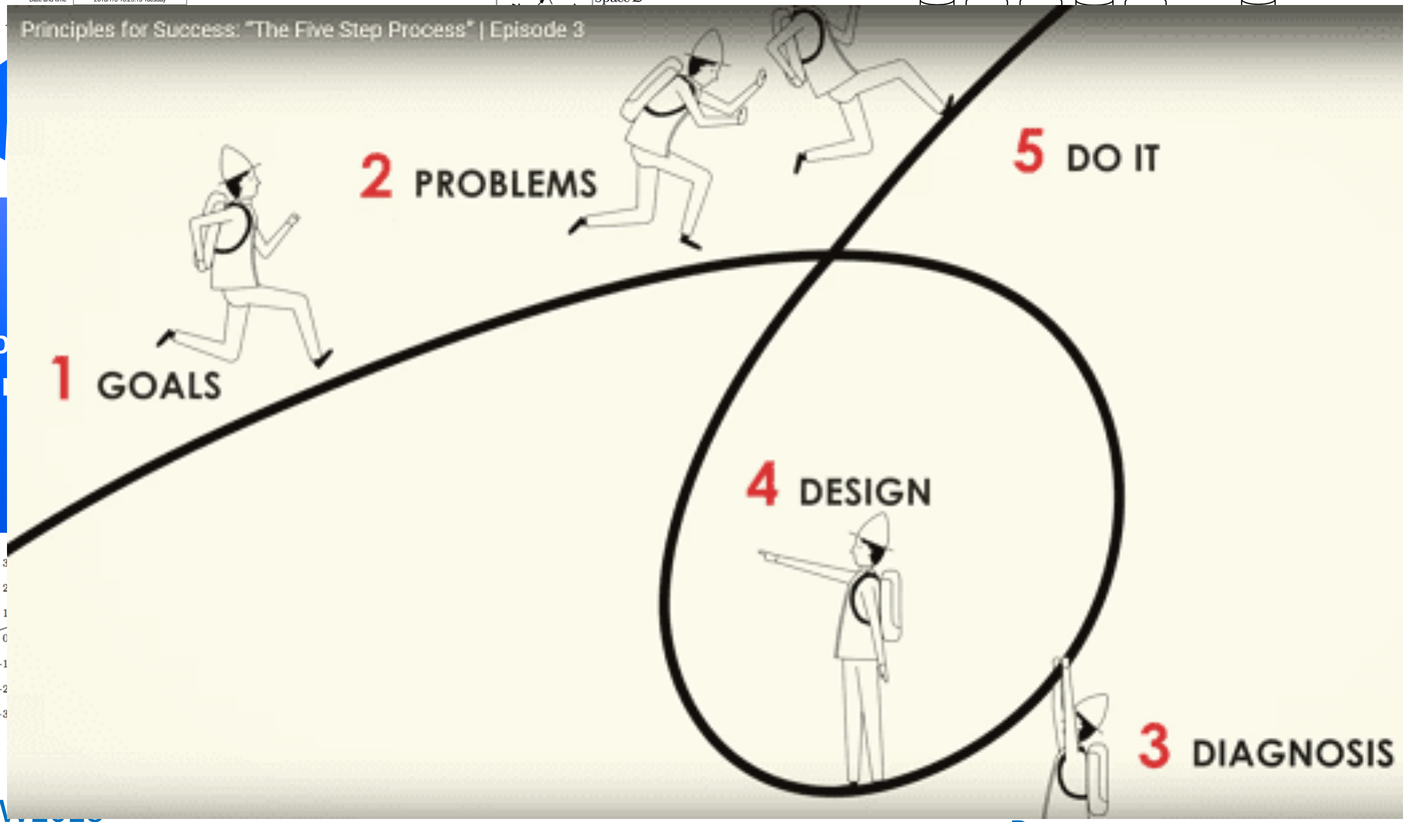
INFOCOM 2012




IMC 2015



WWW




Paper

# Lesson 3 : 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**
3. **Semi-supervised approaches; supervised approaches +transfer learning**
4. **Supervised approaches**

# Lesson 4: 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

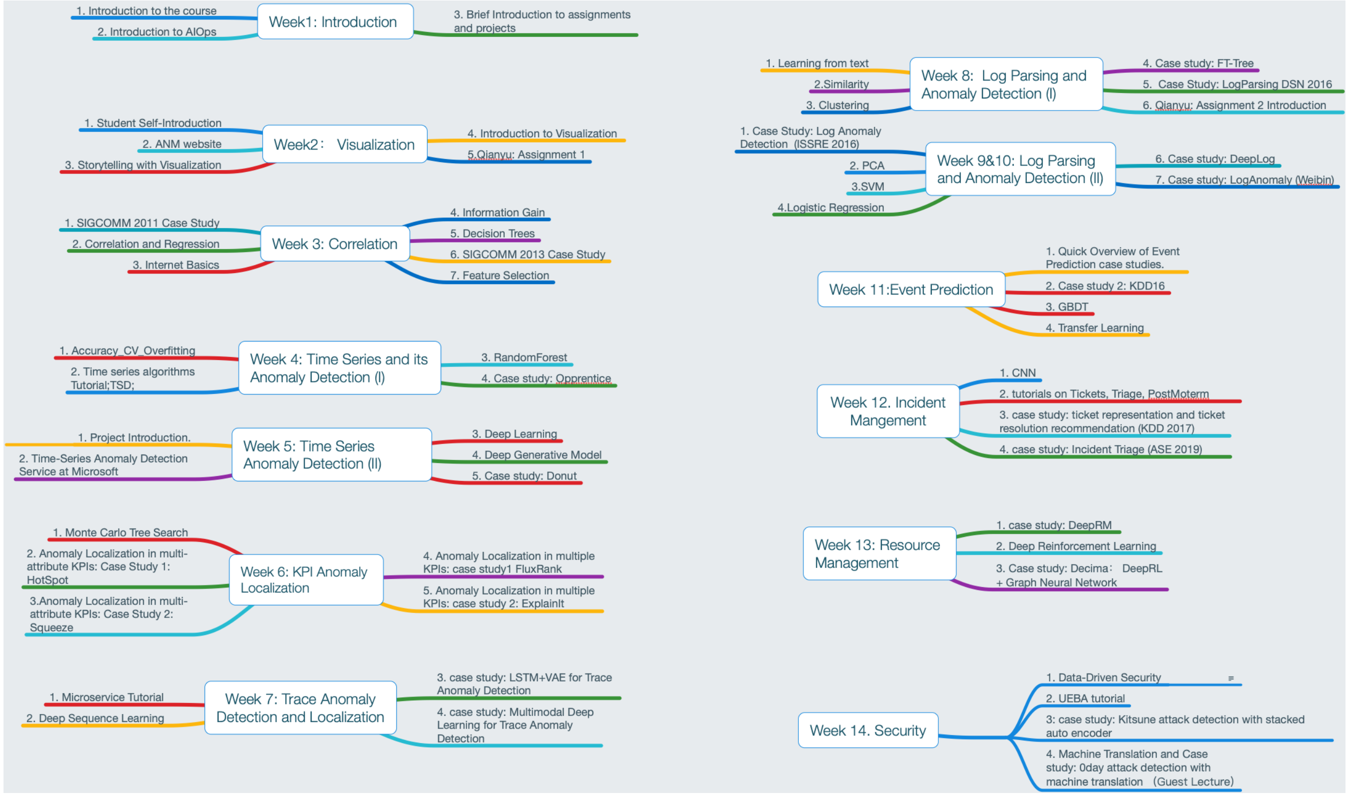
1st AIOps Challenge: time series anomaly detection. Published labeled data from 5 Internet companies. More than 50 teams participated. Papers based on these data were published in KDD, IWQoS, etc.

2nd AIOps Challenge: multi-attribute time series anomaly localization. Published data from an Internet company. More than 60 teams participated.

3rd AIOps Challenge: Realtime anomaly detection and localization on a large-scale testbed with replayed real data.


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

***with literature collected and sorted by AIOps topics***



# Summary

- AI for IT Operations (AIOps) is an interdisciplinary research field between Machine Learning and Systems/Networking/Software Engineering/Security
- 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
  - From practice, into practice
  - As little labeling as possible
- Community efforts are needed to solve AIOps problems

# Thanks！

清華大學 | NetMan Tsinghua