# LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs

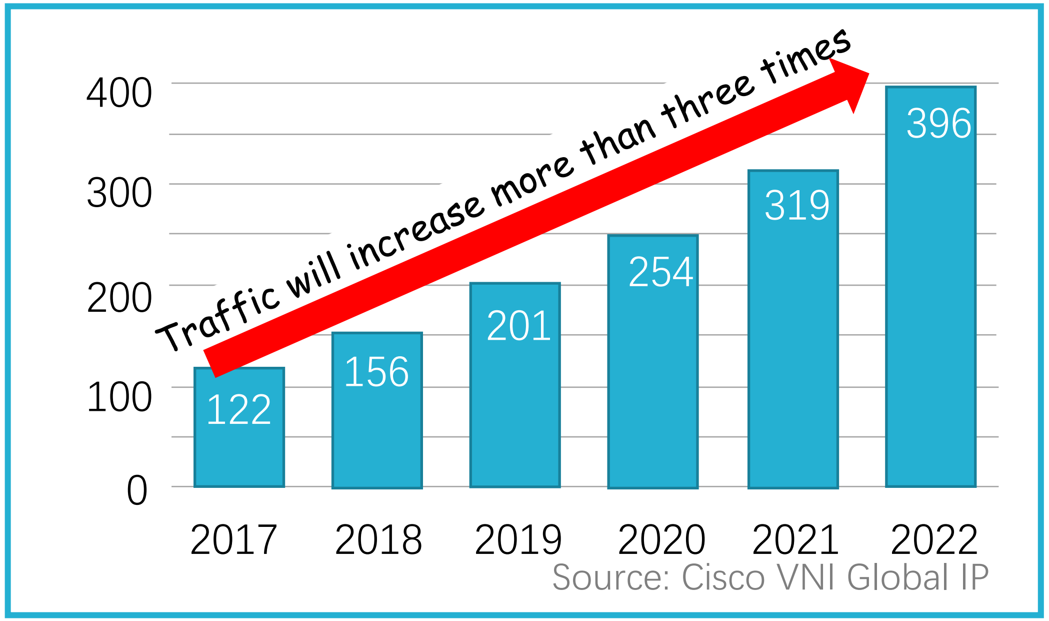
**Weibin Meng**, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei
Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun and Rong Zhou

# Internet Services

Internet provide various types of services

The number of services is growing rapidly

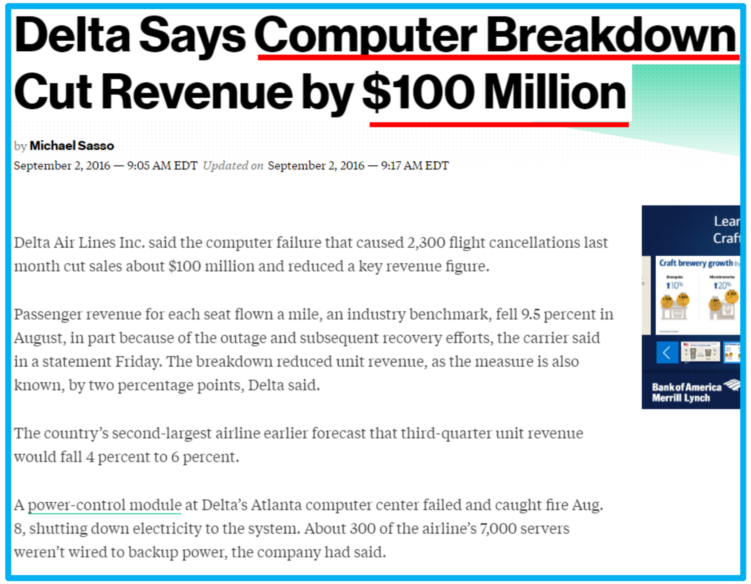
Stability of services are becoming more important



Source: Cisco VNI Global IP

# Anomaly Detection

- Anomalies will impact revenue and user experience.
- Anomaly detection plays an important role in service management.



## Delta Says Computer Breakdown Cut Revenue by $100 Million

by **Michael Sasso**
September 2, 2016 — 9:05 AM EDT *Updated on* September 2, 2016 — 9:17 AM EDT

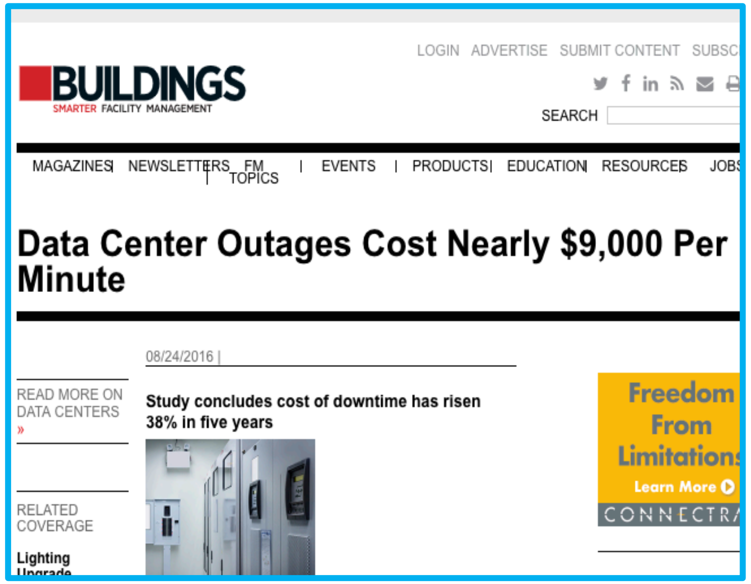
Delta Air Lines Inc. said the computer failure that caused 2,300 flight cancellations last month cut sales about $100 million and reduced a key revenue figure.

Passenger revenue for each seat flown a mile, an industry benchmark, fell 9.5 percent in August, in part because of the outage and subsequent recovery efforts, the carrier said in a statement Friday. The breakdown reduced unit revenue, as the measure is also known, by two percentage points, Delta said.

The country’s second-largest airline earlier forecast that third-quarter unit revenue would fall 4 percent to 6 percent.

A power-control module at Delta’s Atlanta computer center failed and caught fire Aug. 8, shutting down electricity to the system. About 300 of the airline’s 7,000 servers weren’t wired to backup power, the company had said.

Lear
Craf
Craft brewery growth
Bank of America Merrill Lynch



BUILDINGS
SMARTER FACILITY MANAGEMENT

LOGIN ADVERTISE SUBMIT CONTENT SUBSC

SEARCH

MAGAZINES | NEWSLETTERS | FM TOPICS | EVENTS | PRODUCTS | EDUCATION | RESOURCES | JOBS

## Data Center Outages Cost Nearly $9,000 Per Minute

08/24/2016 |

READ MORE ON DATA CENTERS »

RELATED COVERAGE

Lighting Upgrade

**Study concludes cost of downtime has risen 38% in five years**

Freedom From Limitations
Learn More
CONNECTRA

# Logs for Anomaly Detection

- **Logs** are one of **the most valuable** data for anomaly detection

## Diverse

- Logs record a vast range of runtime information

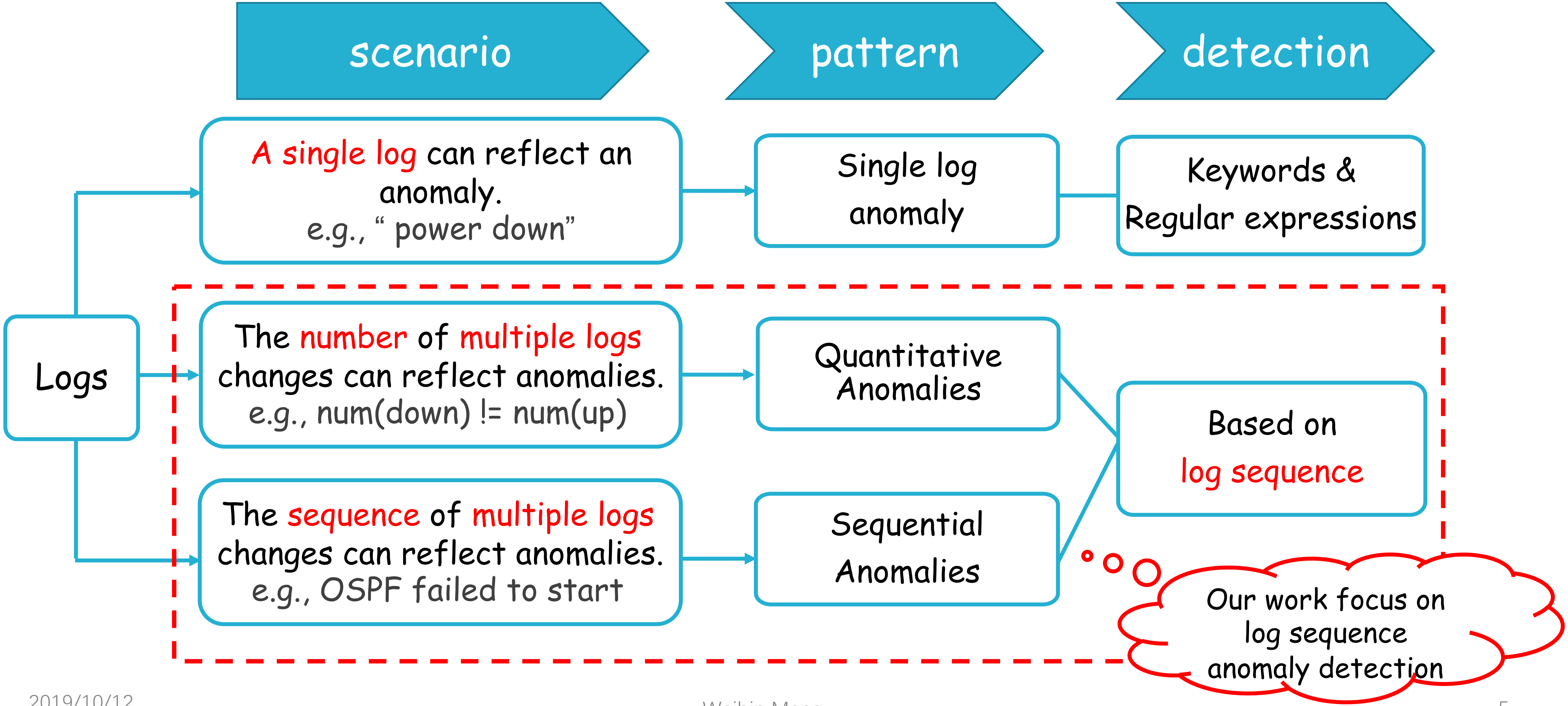
## General

- Every service and device generates logs

| Types | Timestamps | Detailed messages |
|---|---|---|
| Switch | Jul 10 19:03:03 | Interface te-1/1/59, changed state to down |
| Supercomputer | Jun 4 6:45:50 | RAS KERNEL INFO 87 L3 EDRAM error(s) (dcr 0x0157) detected and corrected over 27362 seconds |
| HDFS | Jun 8 13:42:26 | INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_-1608999687919862906 terminating |
| Router | Jul 11 11:05:07 | Neighbour(rid:10.231.0.43, addr:10.231.39.61) on vlan23, changed state from Exchange to Loading |

Unstructured logs

# Logs for Anomaly Detection



scenario

pattern

detection

Logs

A single log can reflect an anomaly.
e.g., " power down"

Single log anomaly

Keywords & Regular expressions

The number of multiple logs changes can reflect anomalies.
e.g., num(down) != num(up)

Quantitative Anomalies

The sequence of multiple logs changes can reflect anomalies.
e.g., OSPF failed to start

Sequential Anomalies

Based on log sequence

Our work focus on log sequence anomaly detection

# Manual Detection

The explosion of logs

- e.g., 10T/day in Huawei

An operator has incomplete information of the overall system

Not all anomalies are explicitly displayed

- Some anomalies hide in log

Automatically detect anomalies based on unstructured logs

**Workflow o**
Down → A

**Runtime logs**:

```
OSPF ADJCHG, Nbr 1.1.1.1 on FastEthernet0/0 from Attempt to Init
OSPF ADJCHG, Nbr 1.1.1.1 on FastEthernet0/0 from Init to Two-way
OSPF ADJCHG, Nbr 1.1.1.1 on FastEthernet0/0 from Two-way to Exstart
OSPF ADJCHG, Nbr 1.1.1.1 on FastEthernet0/0 from Two-way to Exstart
```

Every log is normal,
but OSPF failed to start

**Runtime logs:**

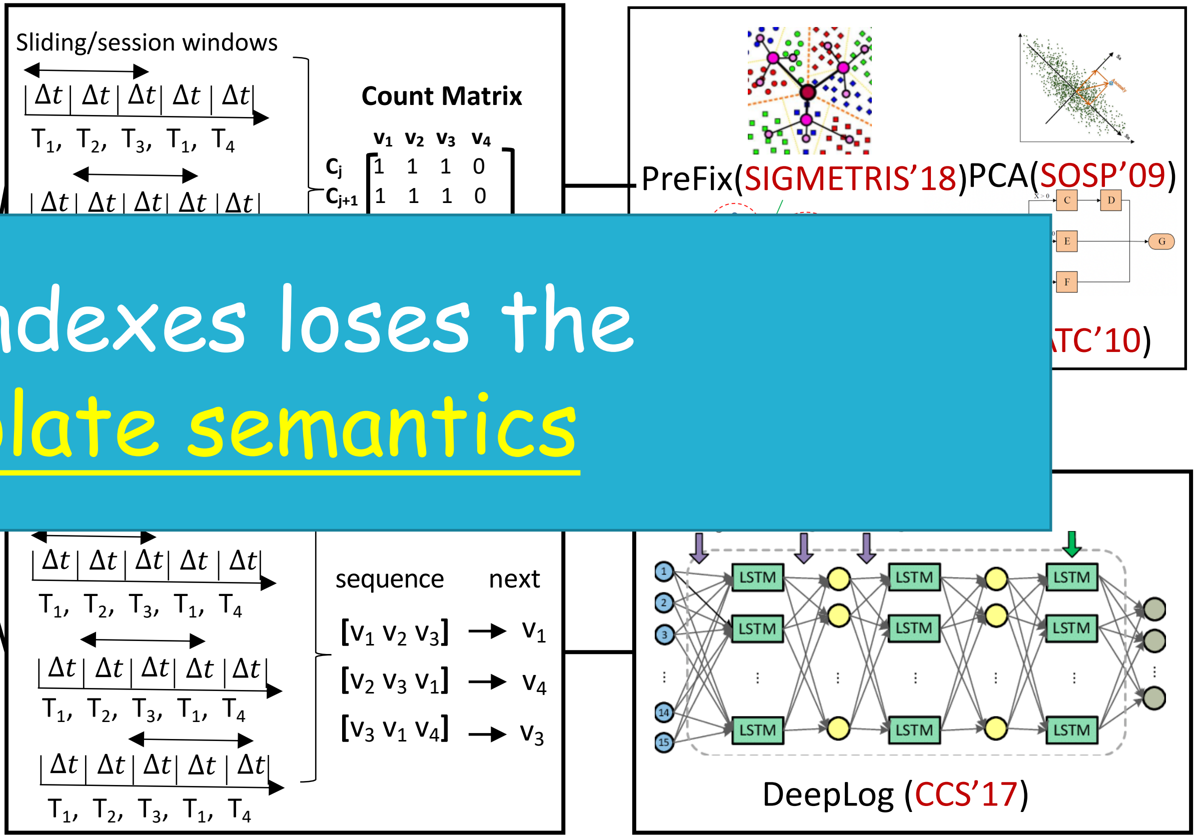
```
Line protocol on Interface ae3, changed state to down
Interface ae3, changed state to down
Interface ae3, changed state to up
```

An interface down event occurs

# Previous studies

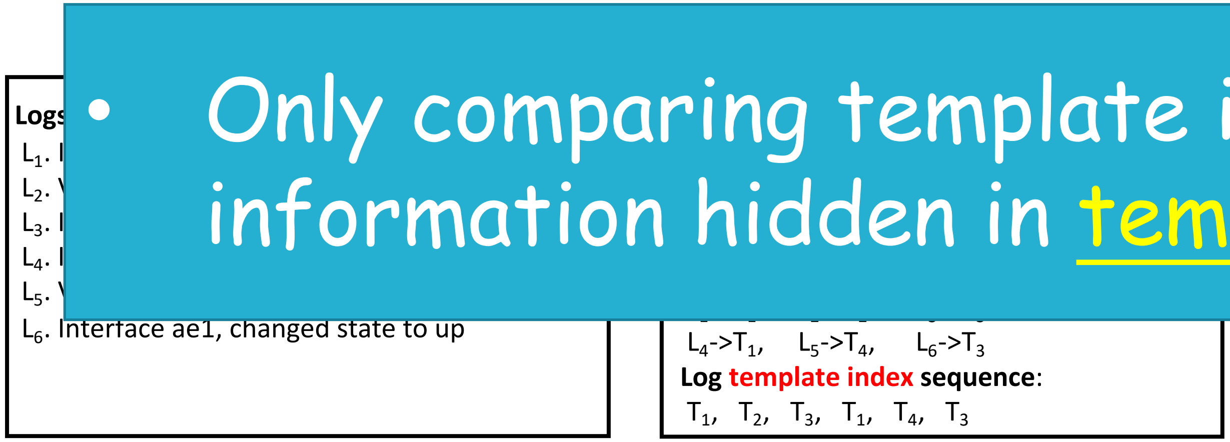
- ■Existing log anomaly detection:
  - ■Quantitative pattern based methods
  - ■Sequential pattern based methods

Quantitative anomalies detection methods



Sliding/session windows
$|\Delta t|\Delta t|\Delta t|\Delta t|\Delta t|$
$T_1, T_2, T_3, T_1, T_4$
$|\Delta t|\Delta t|\Delta t|\Delta t|\Delta t|$
Count Matrix
$v_1$ $v_2$ $v_3$ $v_4$
$C_j$ 1 1 1 0
$C_{j+1}$ 1 1 1 0
PreFix(SIGMETRIS'18)PCA(SOSP'09)
C
D
E
F
G
TC'10)
$|\Delta t|\Delta t|\Delta t|\Delta t|\Delta t|$
$T_1, T_2, T_3, T_1, T_4$
$|\Delta t|\Delta t|\Delta t|\Delta t|\Delta t|$
$T_1, T_2, T_3, T_1, T_4$
$|\Delta t|\Delta t|\Delta t|\Delta t|\Delta t|$
$T_1, T_2, T_3, T_1, T_4$
sequence next
$[v_1\ v_2\ v_3] \rightarrow v_1$
$[v_2\ v_3\ v_1] \rightarrow v_4$
$[v_3\ v_1\ v_4] \rightarrow v_3$
1
2
3
14
15
LSTM
DeepLog (CCS'17)

Sequential anomalies detection methods



Logs

$L_1$. I

$L_2$. V

$L_3$. I

$L_4$. I

$L_5$. V

$L_6$. Interface ae1, changed state to up

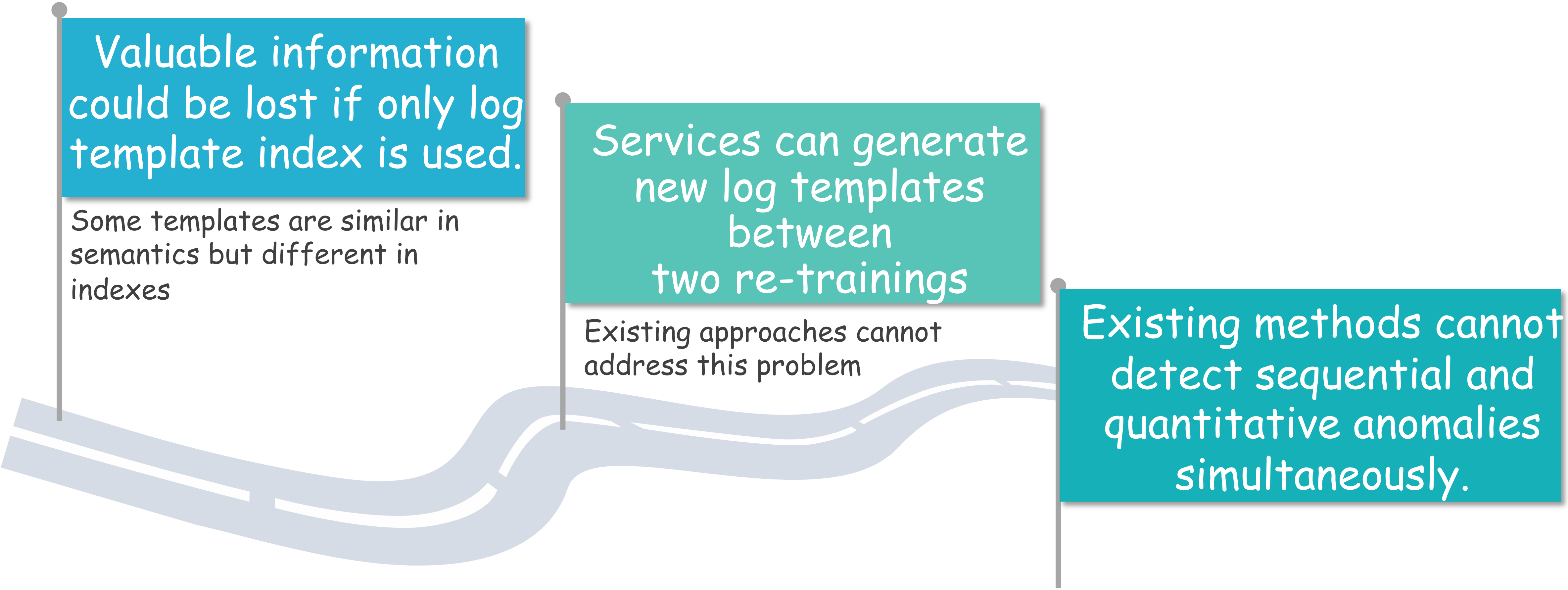
$L_4$->$T_1$, $L_5$->$T_4$, $L_6$->$T_3$

Log template index sequence:

$T_1$, $T_2$, $T_3$, $T_1$, $T_4$, $T_3$

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

# Challenges



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

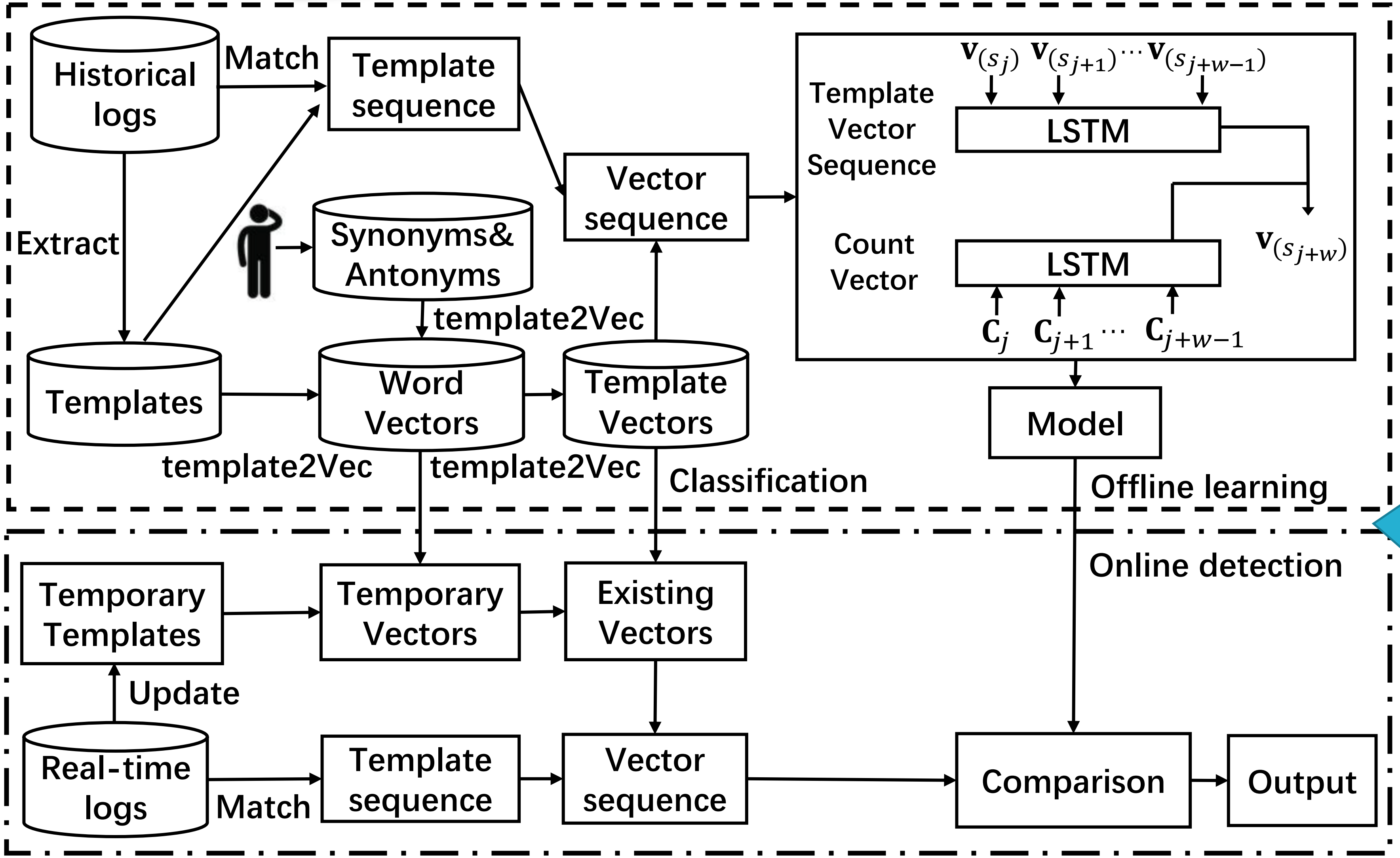
Some templates are similar in semantics but different in indexes

**Services can generate new log templates between two re-trainings**

Existing approaches cannot address this problem

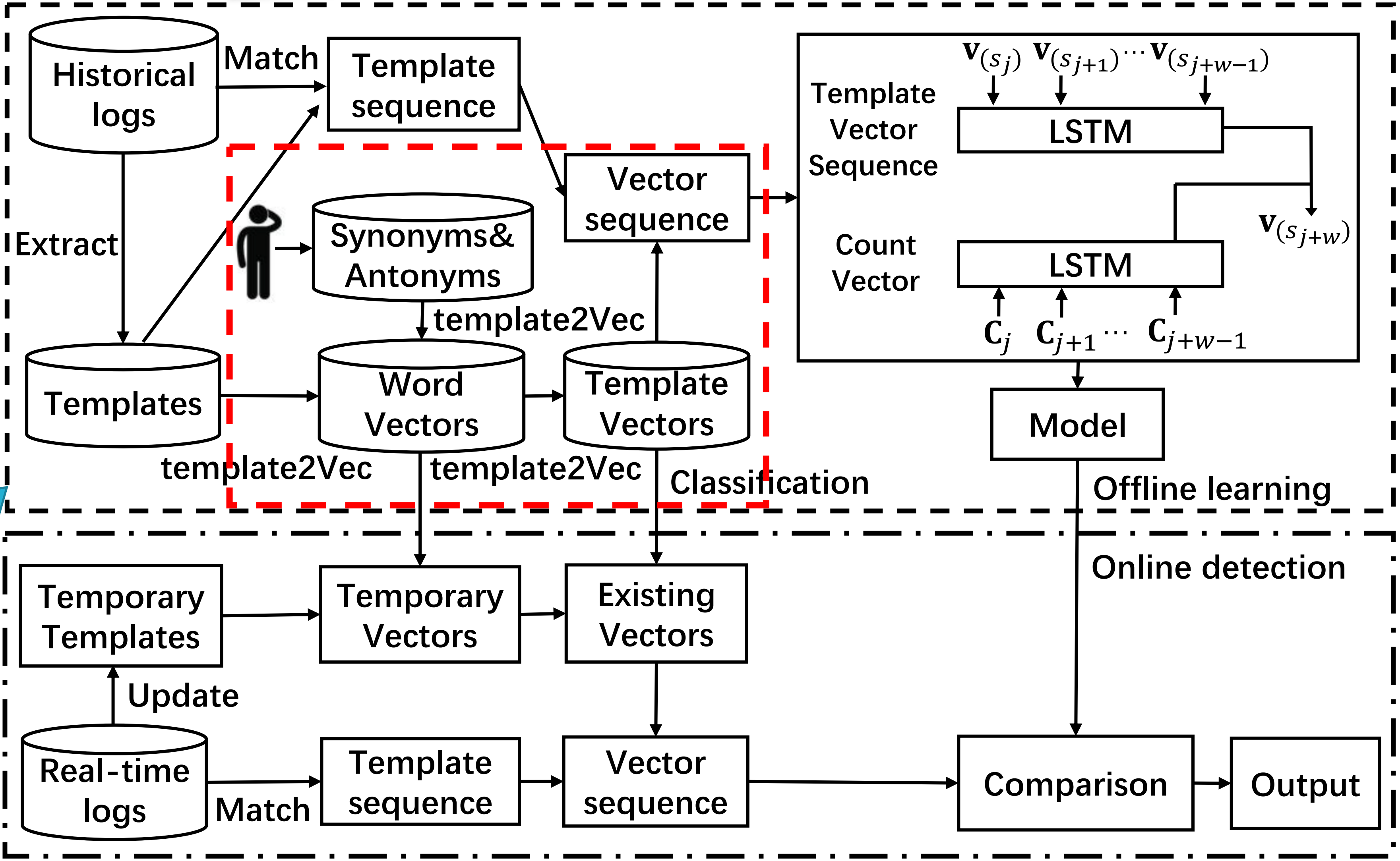
**Existing methods cannot detect sequential and quantitative anomalies simultaneously.**

# Overview of LogAnomaly



An anomaly detection system based on unstructured logs

# Template Representation



Address the first challenge and save template semantics.

# Template Representations

## Insights

- Some existing templates have similar semantics
- Some logs containing antonyms look similar but have opposite semantics

## Goals

- Convert log templates to "soft" representations
- Takes antonyms and synonyms into consideration

**Logs:**

```
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
```

**Templates :**

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**

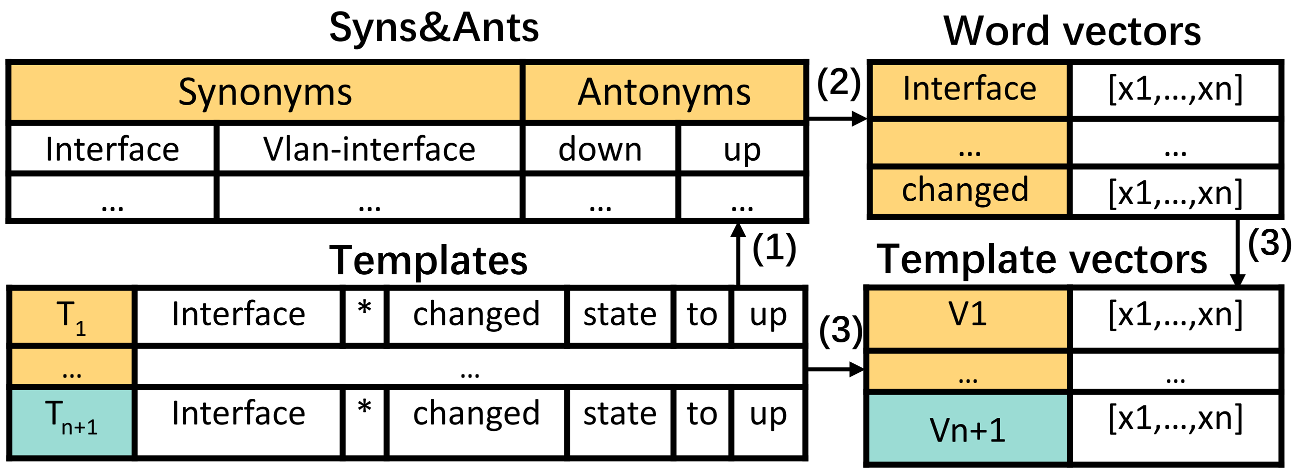
**Logs>Templates:**

L1->T1 L2->T2 L3->T3 L4->T4
L5->T1 L6->T2 L7->T3 L8->T4

# Template2Vec

- template2Vec： (template representation method)
  1. Construct the set of synonyms and antonyms
     - Combine domain knowledge and WordNet
  2. Generate word vectors by using dLCE[1] algorithm
     - dLCE is a distributional lexical-contrast embedding model
  3. Calculate template vectors.

| Relations | Word pairs | | Adding methods |
|---|---|---|---|
| Synonyms | down | low | WordNet |
| | Interface | port | Operators |
| Antonyms | DOWN | UP | WordNet |
| | powerDown | powerOn | Operators |



Syns&Ants

| Synonyms | | Antonyms | |
|---|---|---|---|
| Interface | Vlan-interface | down | up |
| … | … | … | … |

Word vectors

| | |
|---|---|
| Interface | [x1,…,xn] |
| … | … |
| changed | [x1,…,xn] |

Templates

| | | | | | | |
|---|---|---|---|---|---|---|
| $T_1$ | Interface | * | changed | state | to | up |
| … | … | | | | | |
| $T_{n+1}$ | Interface | * | changed | state | to | up |

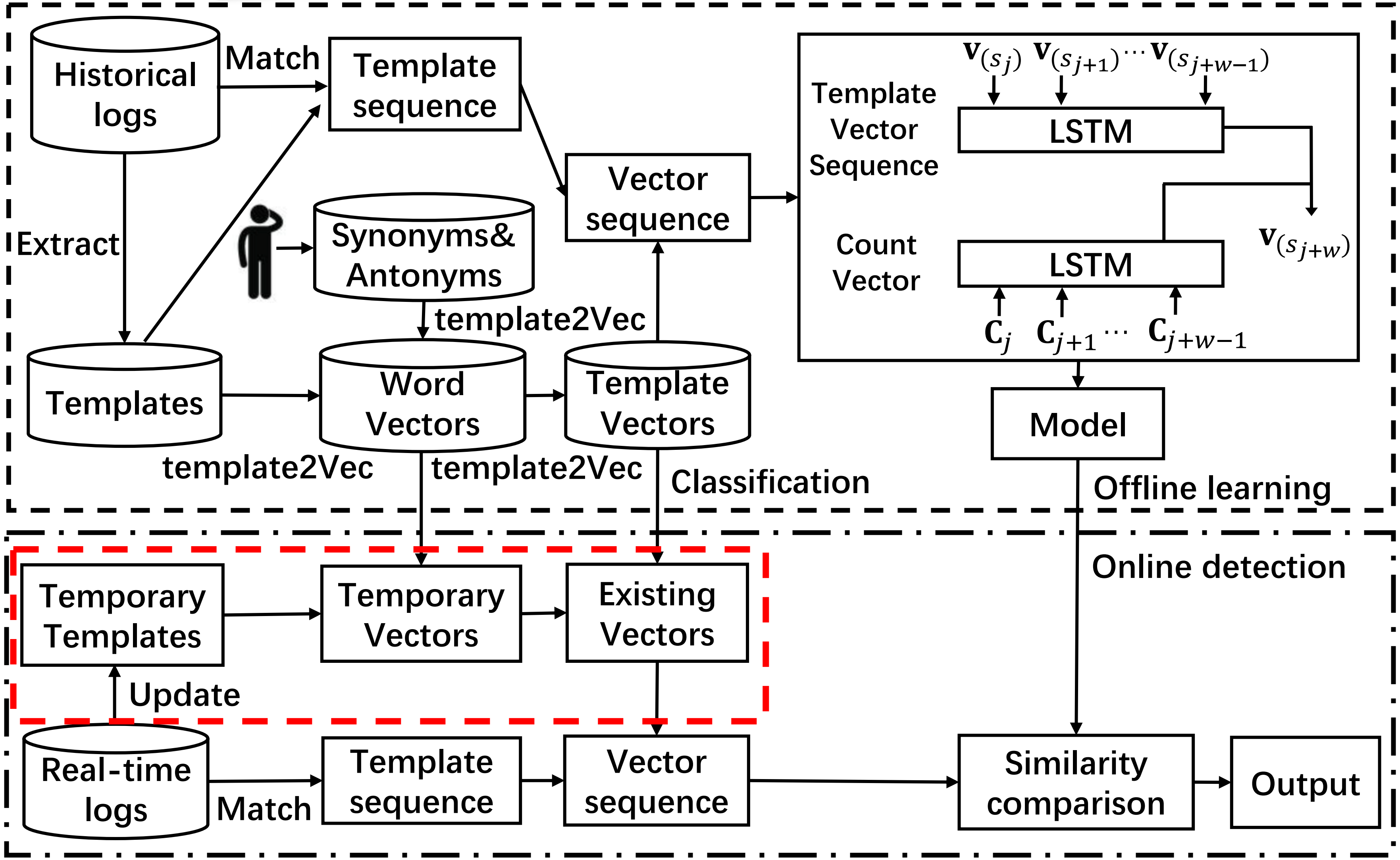
Template vectors

| | |
|---|---|
| V1 | [x1,…,xn] |
| … | … |
| Vn+1 | [x1,…,xn] |

[1] Kim Anh Nguyen, Sabine Schulte, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. *arXiv preprint arXiv:1605.07766*, 2016.

# Template Approximation

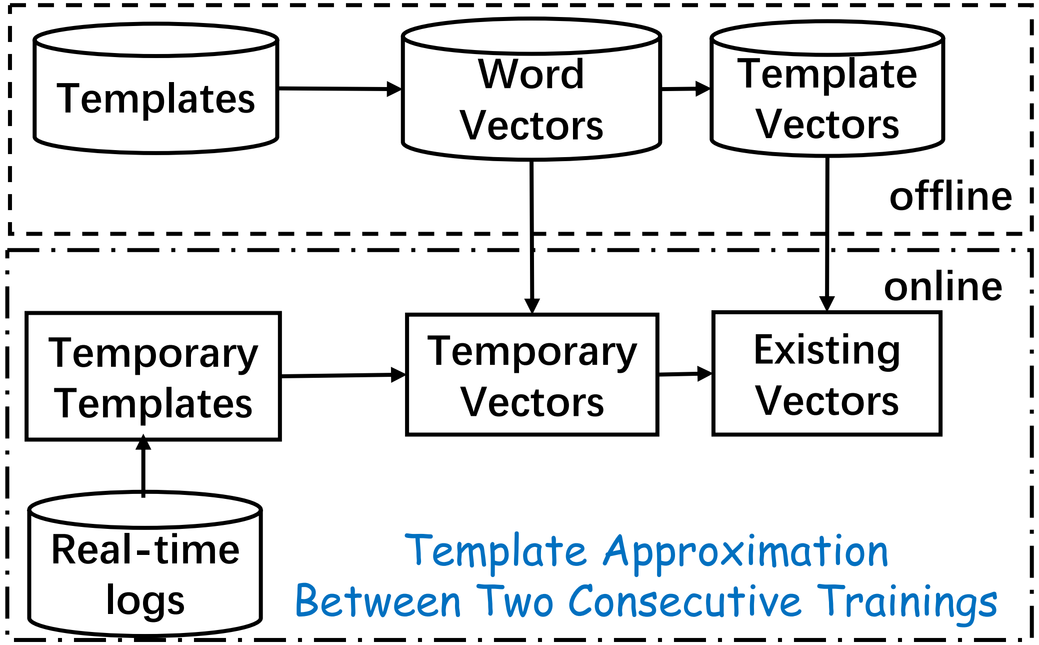
A mechanism to address new templates at runtime



# Template Approximation

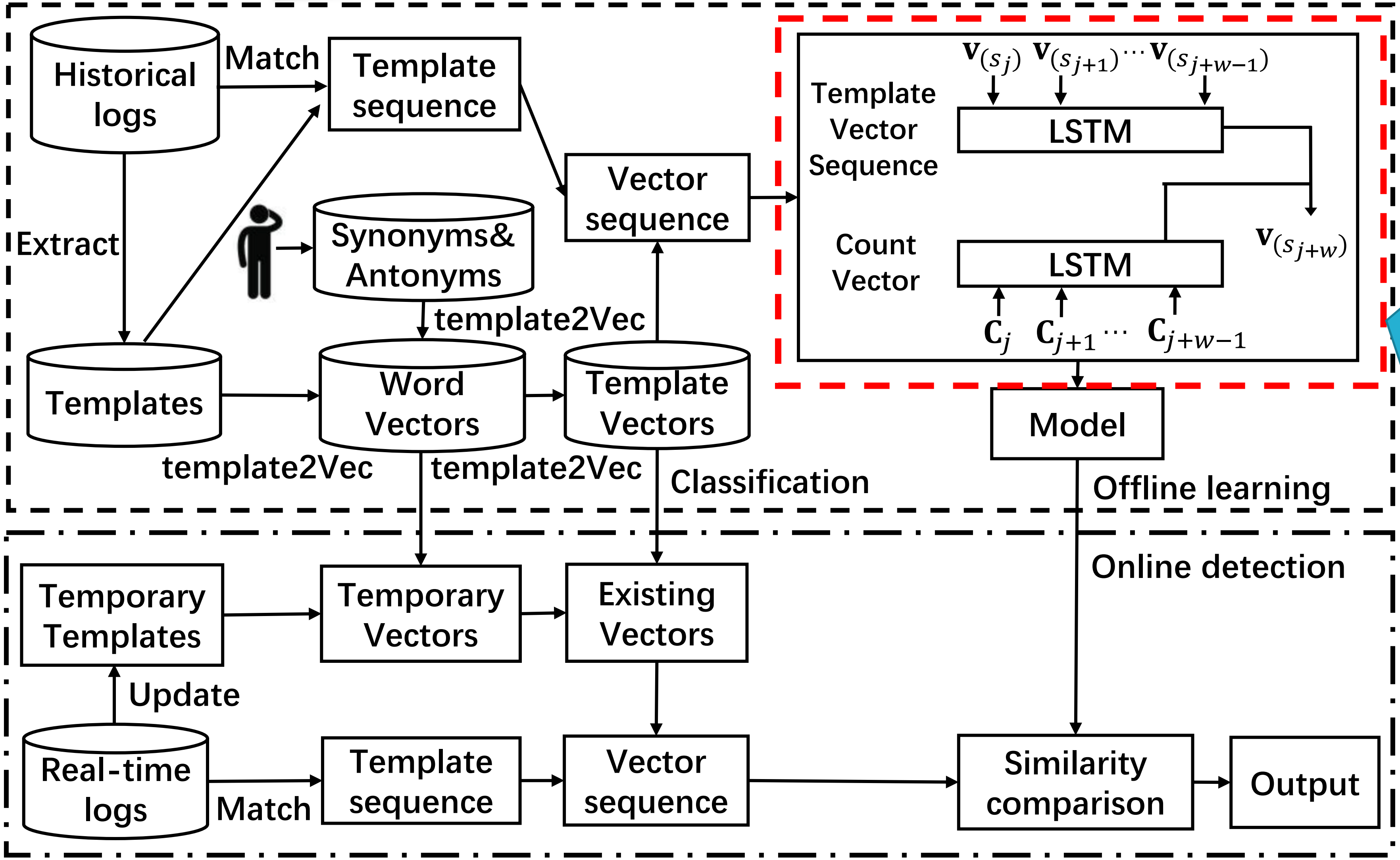
## Between two re-trainings

- Extract a temporary template for the log of a new type
- Map the temporary template vector into one of the existing vector



Template Approximation Between Two Consecutive Trainings

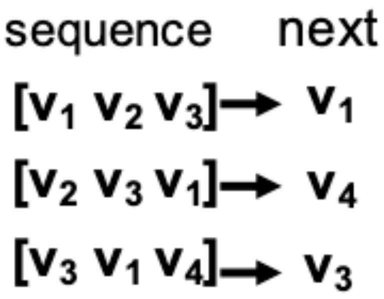
# Anomaly Detection



Address the third challenge and detect two anomalies simultaneously.

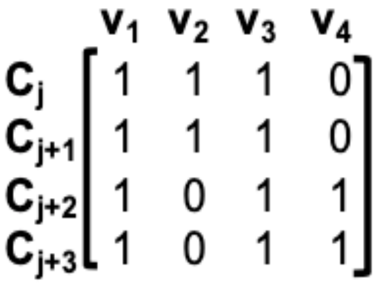
# Anomaly detection

## Sequential pattern (e.g, OSPF starting)



| sequence | next |
|---|---|
| $[v_1\ v_2\ v_3] \rightarrow$ | $v_1$ |
| $[v_2\ v_3\ v_1] \rightarrow$ | $v_4$ |
| $[v_3\ v_1\ v_4] \rightarrow$ | $v_3$ |

## Quantitative pattern (e.g., up = down)



$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline C_j & 1 & 1 & 1 & 0 \\ C_{j+1} & 1 & 1 & 1 & 0 \\ C_{j+2} & 1 & 0 & 1 & 1 \\ C_{j+3} & 1 & 0 & 1 & 1 \end{array}$$

**Logs:**

$L_1$ Interface ae3, changed state to down
$L_2$ Vlan-interface v2, changed state to down
$L_3$ Interface ae3, changed state to up.
$L_4$ Interface ae1, changed state to down
$L_5$ Vlan-interface v2, changed state to up
$L_6$ Interface ae1, changed state to up

**Templates (log keys):**

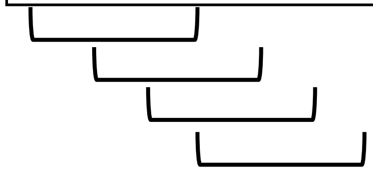
$T_1$ Interface *, changed state to down
$T_2$ Vlan-interface *, changed state to down
$T_3$ Interface *, changed state to up
$T_4$ Vlan-interface *, changed state to up

**Templates index sequence:**

$T_1\ T_2\ T_3\ T_1\ T_4\ T_3$

**Templates vector sequence:**

$v_1\ v_2\ v_3\ v_1\ v_4\ v_3$

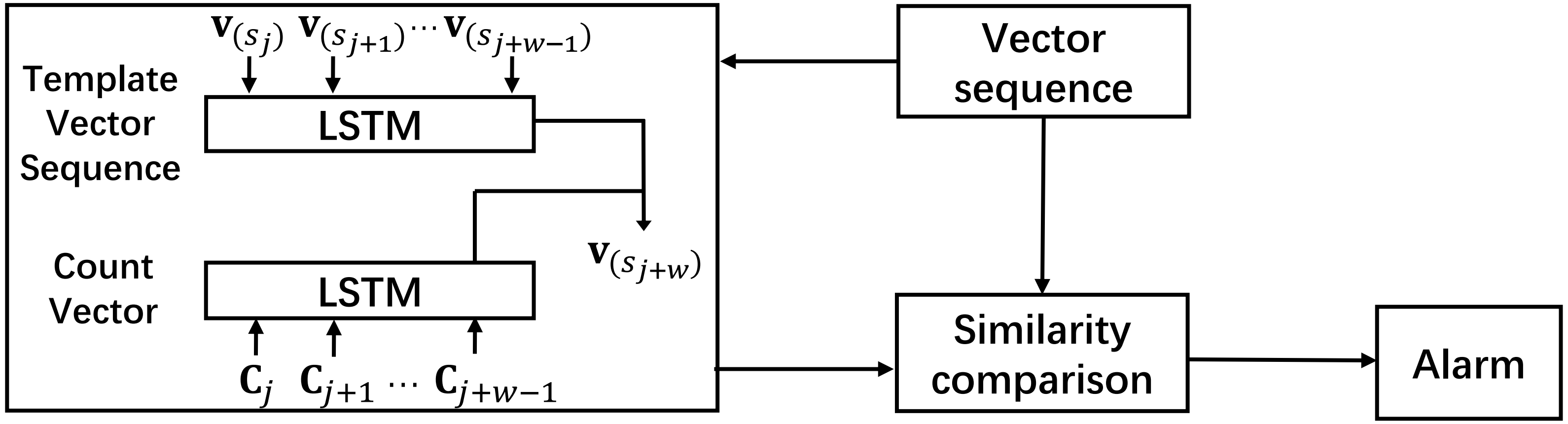


Sliding windows

# Anomaly Detection

## Combine sequential and quantitative relationship

- Sort probabilities:
  - For a log sequence, we sort the possible next template vector based on their probabilities (of appear in the next log).
- Top k candidates :
  - If the observed next template vector is included in the top k candidates (or similar enough with them), we regard it as normal.



Template Vector Sequence

$\mathbf{v}(s_j)$ $\mathbf{v}(s_{j+1})\cdots\mathbf{v}(s_{j+w-1})$

LSTM

$\mathbf{v}(s_{j+w})$

Count Vector

LSTM

$\mathbf{C}_j$ $\mathbf{C}_{j+1}\cdots\mathbf{C}_{j+w-1}$

Vector sequence

Similarity comparison

Alarm

# Evaluation Datasets & Baselines

## Datasets:

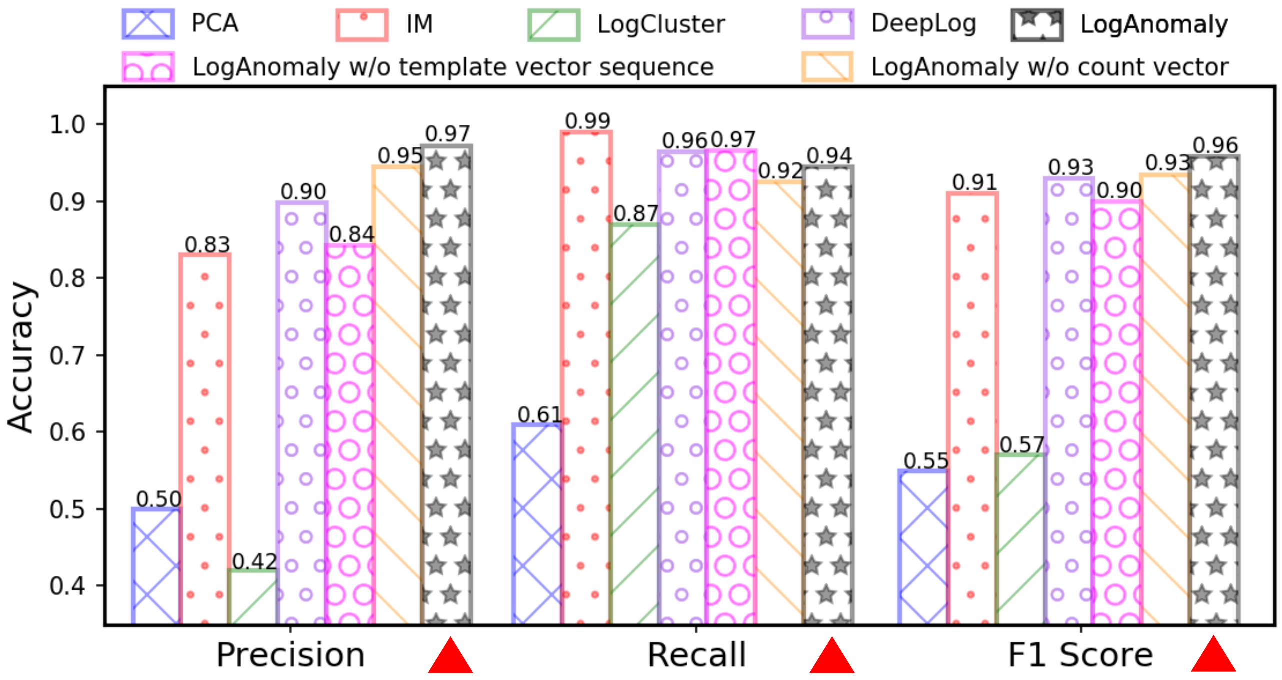
- BGL:
  - Generated by the Blue Gene/L supercomputer.
- HDFS:
  - Collected from more than 200 Amazon nodes.

## Baselines:

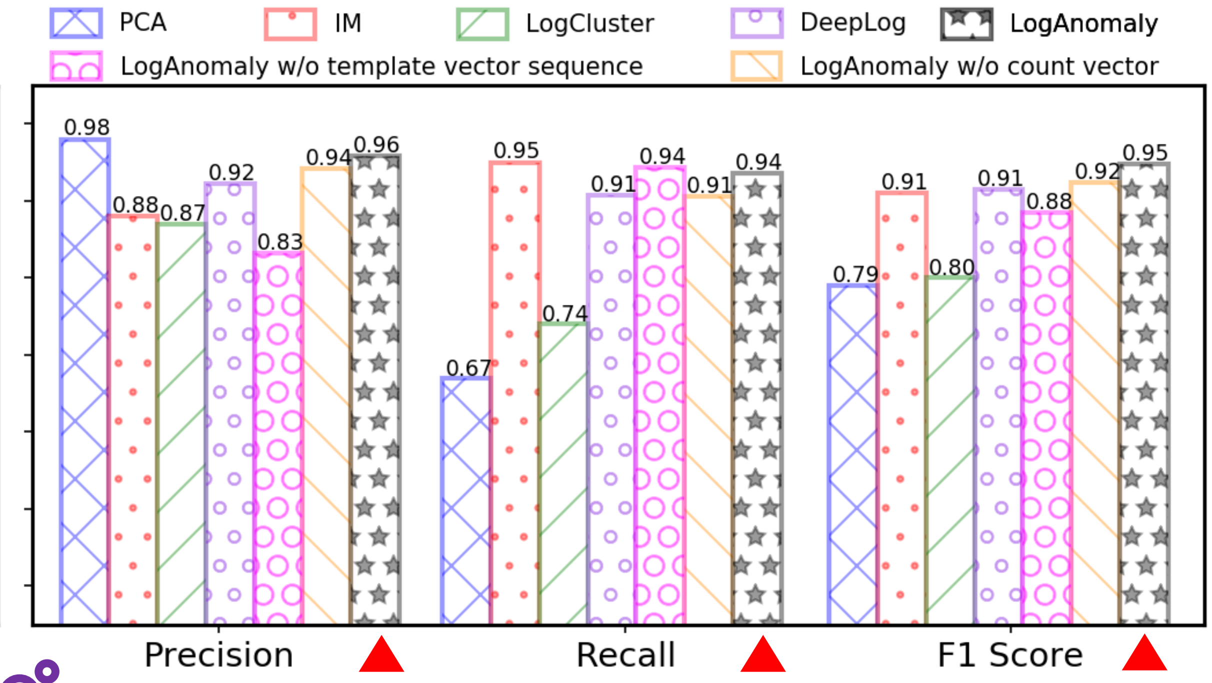
- LogCluster (ICSE'16)
- Invariants Mining (ATC'10)
- PCA (SOSP'09)
- Deeplog (CCS'17)

| Datasets | Duration | # of logs | # of anomalies |
|---|---|---|---|
| BGL | 7 months | 4,747,963 | 348,460 (logs) |
| HDFS | 38.7 hours | 11,175,629 | 16,838 (blocks) |

# Evaluation of LogAnomaly



BGL dataset



HDFS dataset

LogAnomaly achieves
the best performance

# Case Study

## Dataset

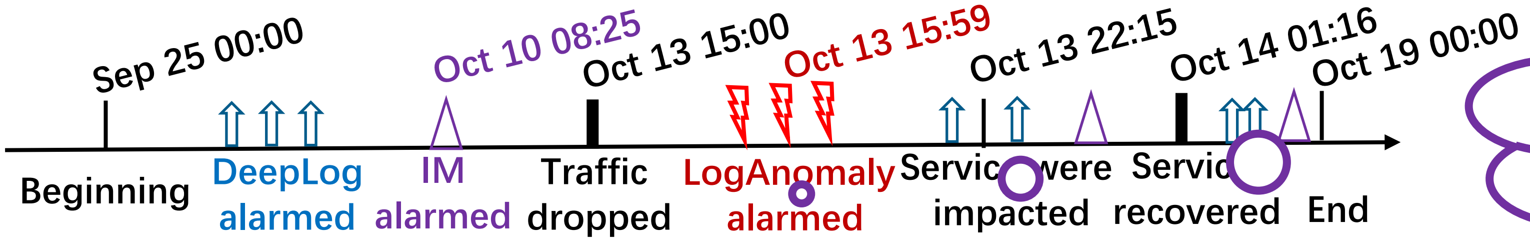
- Logs form an aggregation switch deployed in a top cloud service provider.

## Anomaly description

- The traffic forwarded by this switch dropped from 15:00, Oct 13
- The services provided by this switch were impacted from 22:15, Oct 13
- The switch recovered at 1:16, Oct 14.

## Results

- All of LogAnomaly's alarms were during 15:59 ~ 1:16



Sep 25 00:00 — Beginning

DeepLog alarmed

Oct 10 08:25 — IM alarmed

Oct 13 15:00 — Traffic dropped

Oct 13 15:59 — LogAnomaly alarmed

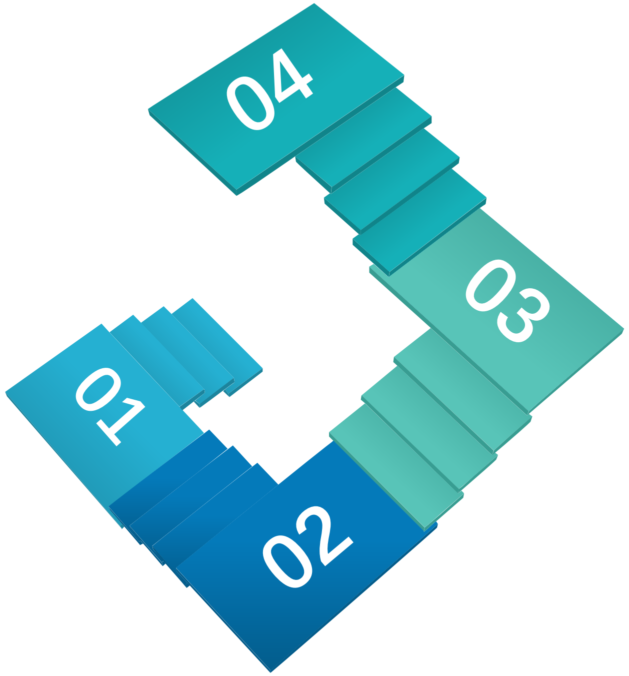
Oct 13 22:15 — Services were impacted

Oct 14 01:16 — Services recovered

Oct 19 00:00 — End

LogAnomaly successfully detected anomalies and generated no false alarm.

# Conclusion



## LogAnomaly

- An anomaly detection system based on unstructured logs.

## template2Vec

- Represent template without losing semantic information.

## Template Approximation

- Merge templates of new types automatically

## Evaluation

- Best results on public datasets and real-world switch logs

# Thanks

mwb16@mails.tsinghua.edu.cn