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

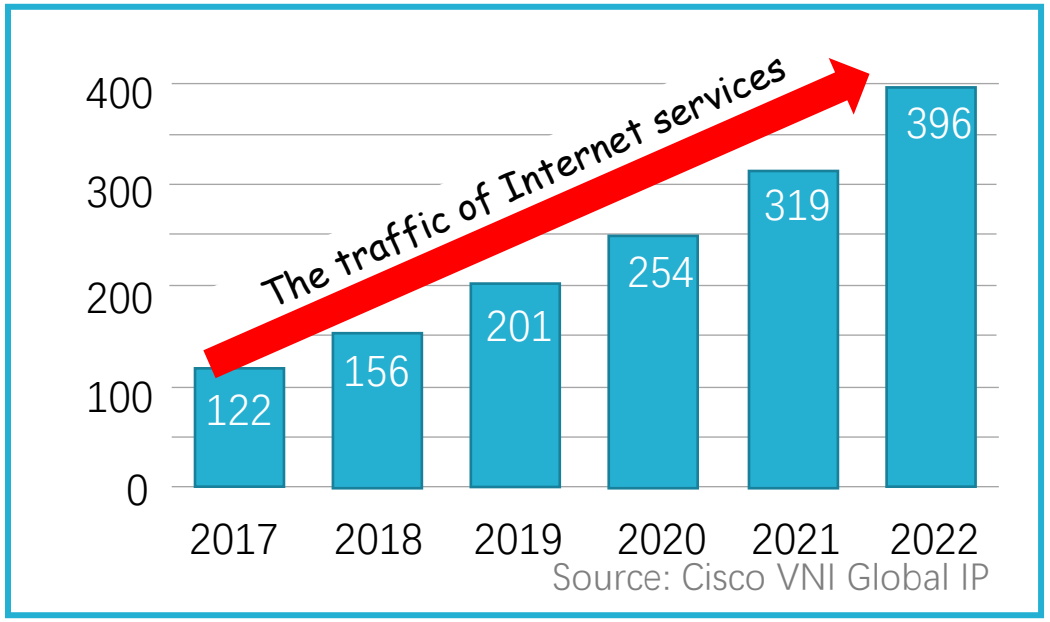
Accepted by IJCAI 19

Internet Services

Internet services provide various types of services

The traffic are growing rapidly

Stability of services are becoming increasingly more important.



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.

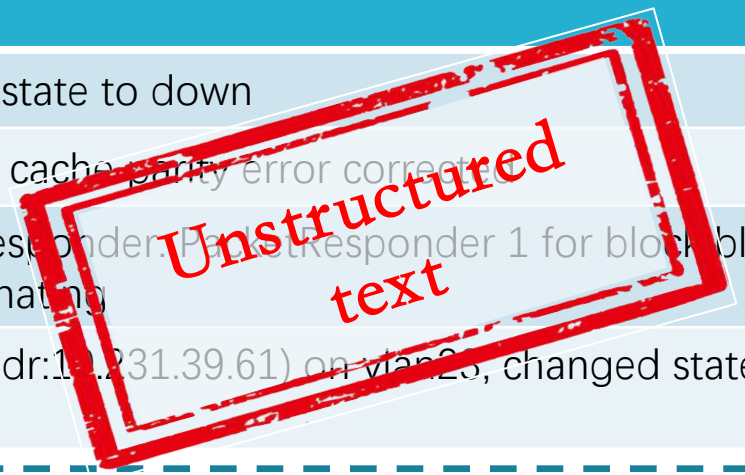
A power-control module at Delta's Atlanta computer center failed and caught fire Aug. 8, shutting down electricity to the system. About 300 of the airline's 7,000 servers weren't wired to backup power, the company had said.

Logs for Anomaly Detection

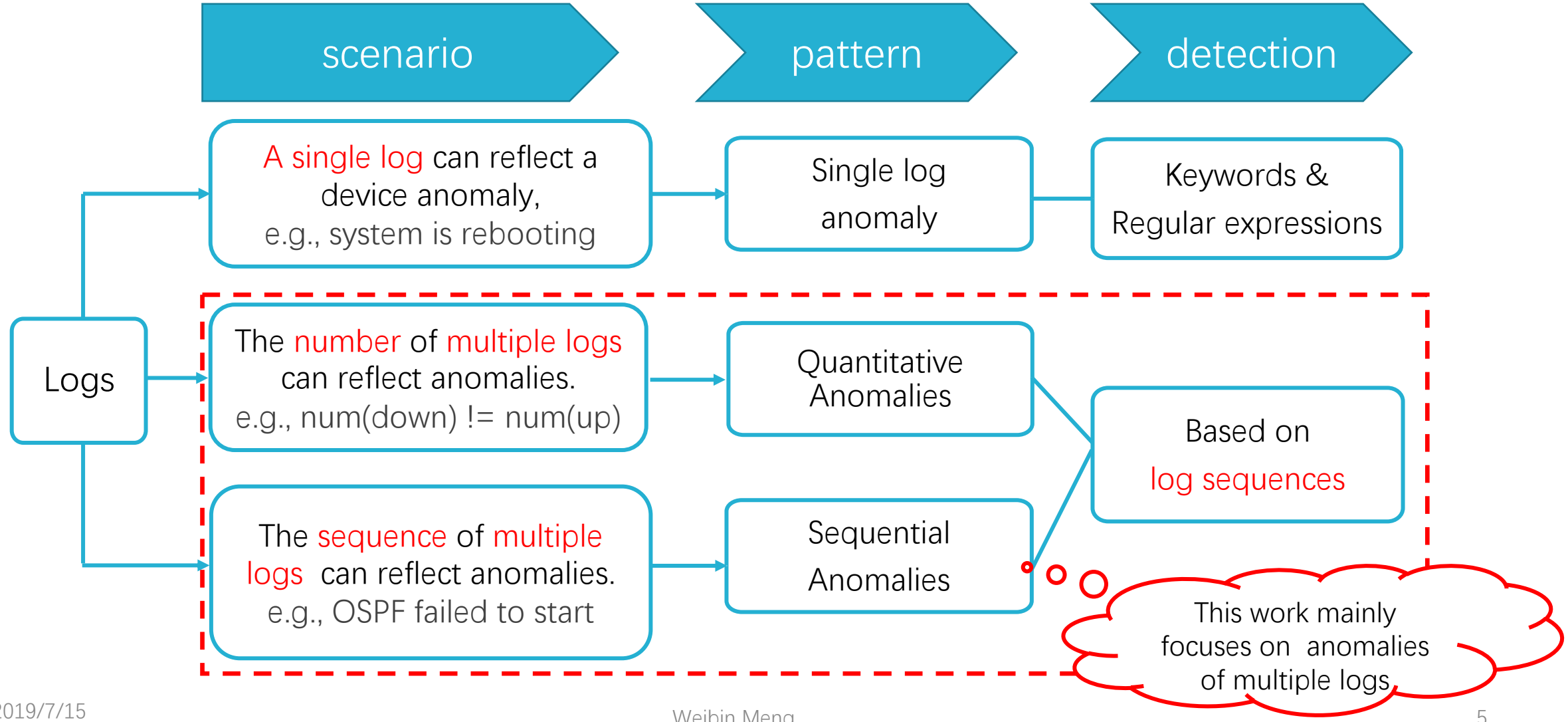
Logs are one of the most valuable data sources for anomaly detection

- Diverse: record a vast range of runtime information
- General: every IT system or 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 instruction cache parity error corrected
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



Logs for Anomaly Detection



Manual Inspection

Incomplete information of the overall system

The explosion of logs

Not all anomalies are explicitly displayed

- Some anomalies hide in log sequence.

Normal OSPF Startup process:
Down → Attempt → Init → Two-way→ Exstart → Exchange → Loading → Full

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**

Each log is normal, but OSPF failed to start

Line protocol on Interface ae3, changed state to **down**
Interface ae3, changed state to **down**
Interface ae3, changed state to **up**
Line protocol on Interface ae3, changed state to **up**
Line protocol on Interface ae3, changed state to **down**
Interface ae3, changed state to **down**
Interface ae3, changed state to **up**.

Line protocol down (the number of up/down is not equal)

Manual Inspection

Incomplete information of the overall system

The explosion of logs

Not all anomalies are explicitly displayed

Automatically and accurately detect anomalies based on **unstructured** logs

N

Down → Attempt → Init → Two-way→ Exstart → Exchange → Loading → Full

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**

n

Interface ae3, changed state to **down**
Interface ae3, changed state to **up**
Line protocol on Interface ae3, changed state to **up**
Line protocol on Interface ae3, changed state to **down**
Interface ae3, changed state to **down**
Interface ae3, changed state to **up**.

Each log is normal, but OSPF failed to start

Line protocol down (the number of up/down is not equal)

Previous Studies

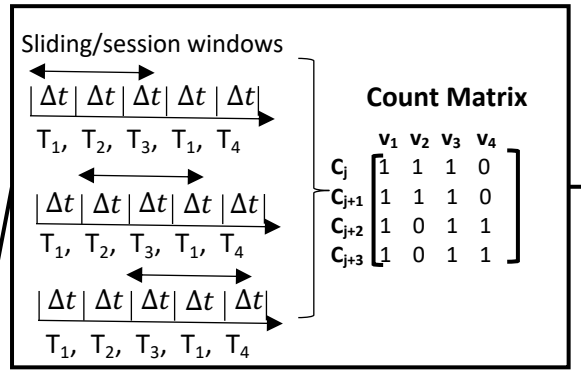
Two categories

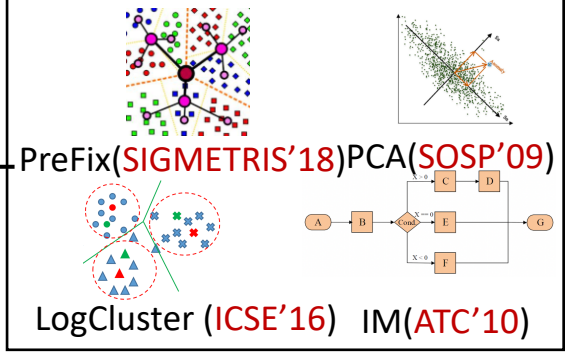
- Quantitative pattern based approaches
- Sequential pattern based approaches

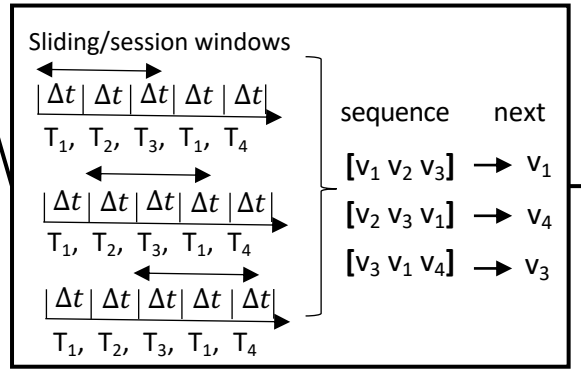
Logs:
L_1. Interface ae3, changed state to down
L_2. Vlan-interface vlan22, changed state to down
L_3. Interface ae3, changed state to up.
L_4. Interface ae1, changed state to down
L_5. Vlan-interface vlan22, changed state to up
L_6. Interface ae1, changed state to up

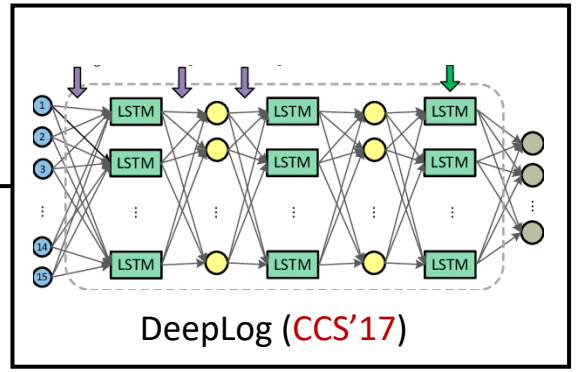
Templates (log keys):
T_1. Interface *, changed state to down
T_2. Vlan-interface *, changed state to down
T_3. Interface *, changed state to up
T_4. Vlan-interface *, changed state to up
Logs -> Template keys:
L_1->T_1, L_2->T_2, L_3->T_3
L_4->T_1, L_5->T_4, L_6->T_3
Log template key sequence:
T_1, T_2, T_3, T_1, T_4, T_3

Quantitative anomaly detection methods









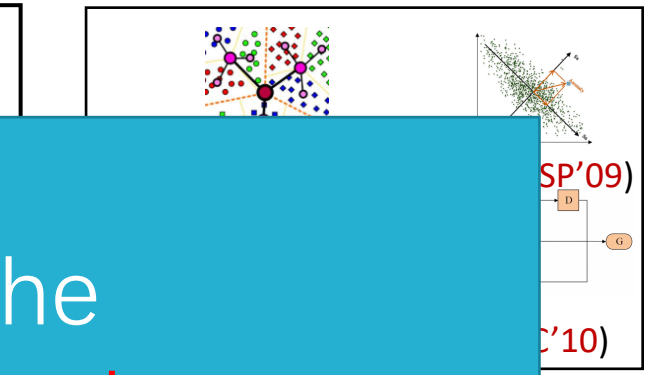
Sequential anomaly detection methods

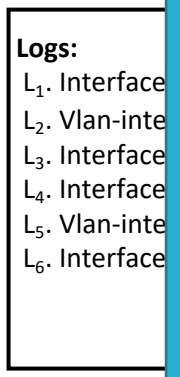
Previous Studies

Two categories:

- Quantitative pattern based approaches
- S

Quantitative anomaly detection methods





- Only comparing template key loses the information hidden in template semantics
- Cannot detect quantitative and sequential anomalies simultaneously

Sequential anomaly detection methods

Challenges

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

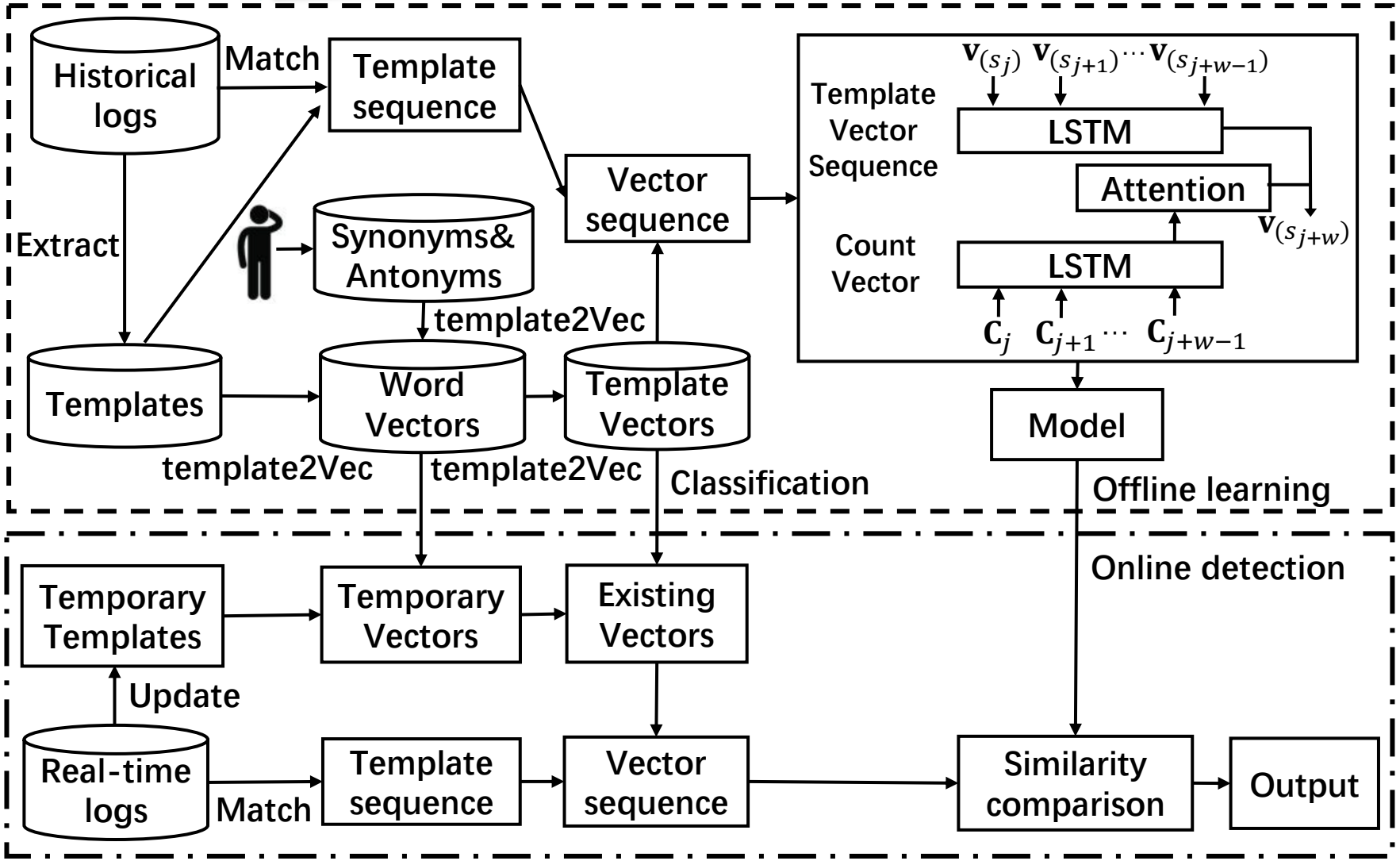
- Some templates are similar in semantics but different in keys

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

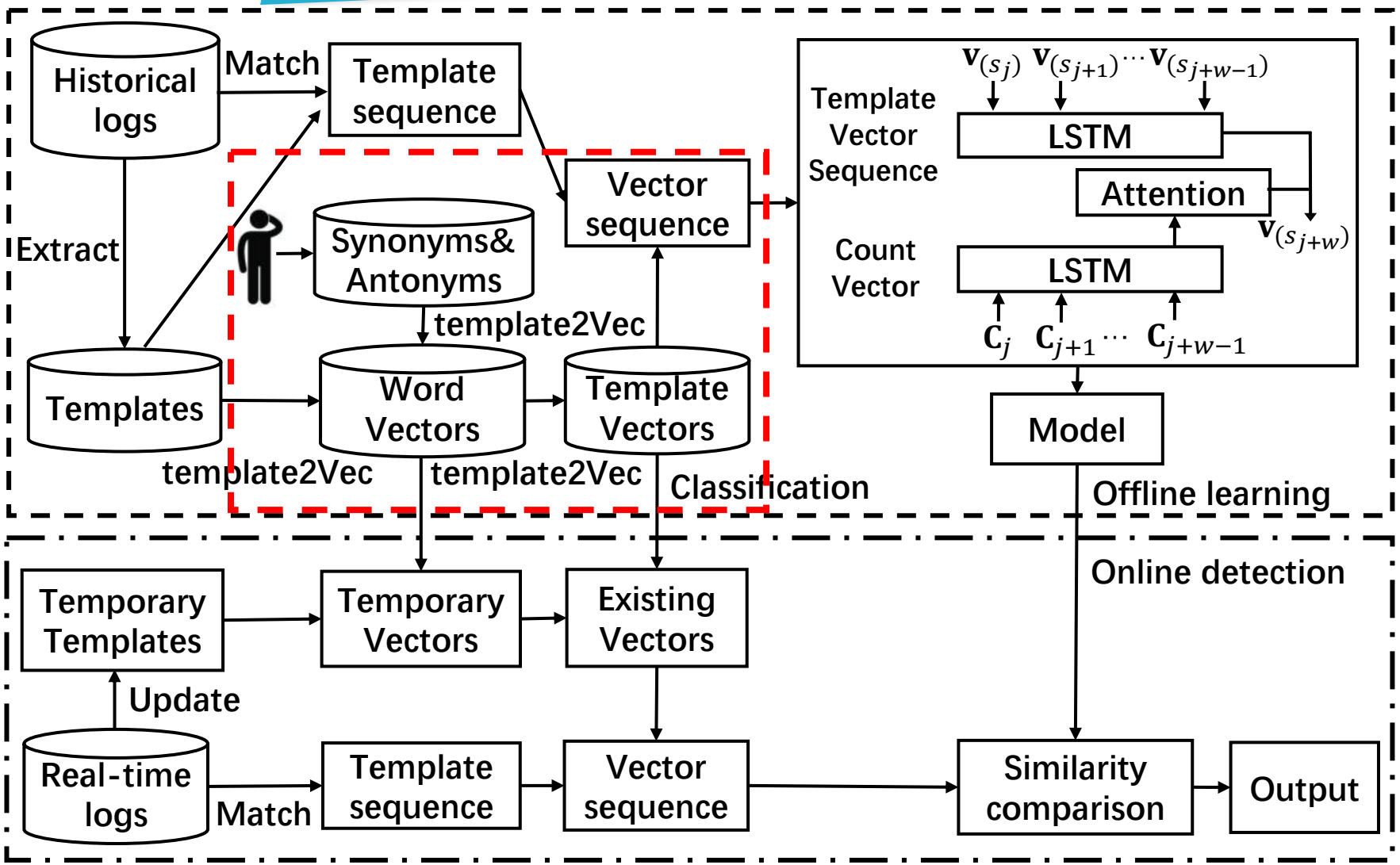
- Existing approaches cannot address this problem

Existing methods cannot detect sequential and quantitative anomalies simultaneously.

Overview of LogAnomaly



Template Representation



Template Representation

Insights

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

Goals

- Convert log templates to “soft” representations (vectors)
- 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

Construct the set of synonyms and antonyms

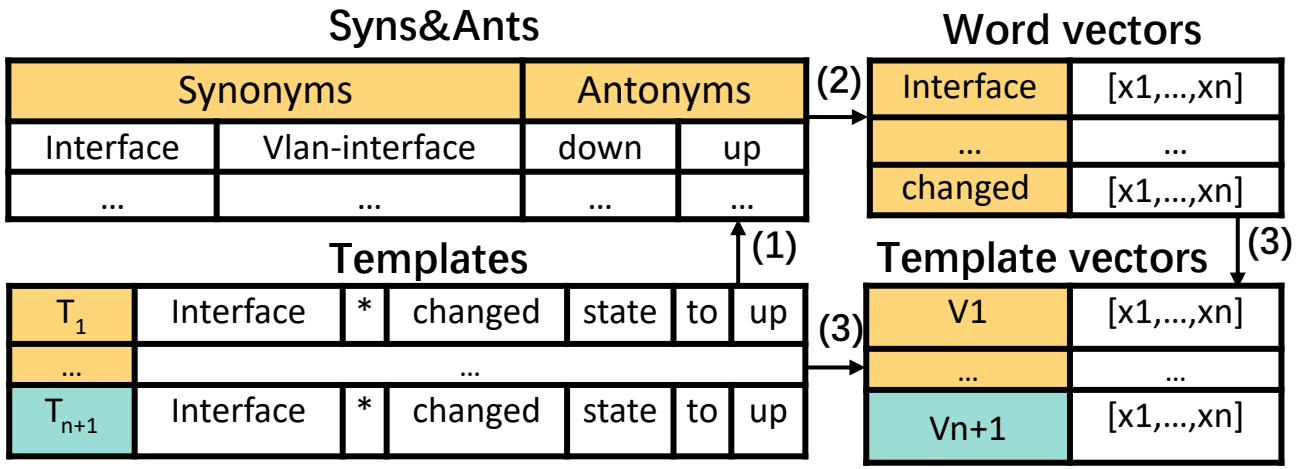
- Combine domain knowledge and WordNet dataset

Generate word vectors by using dLCE[1]

- dLCE is a distributional lexical-contrast embedding model

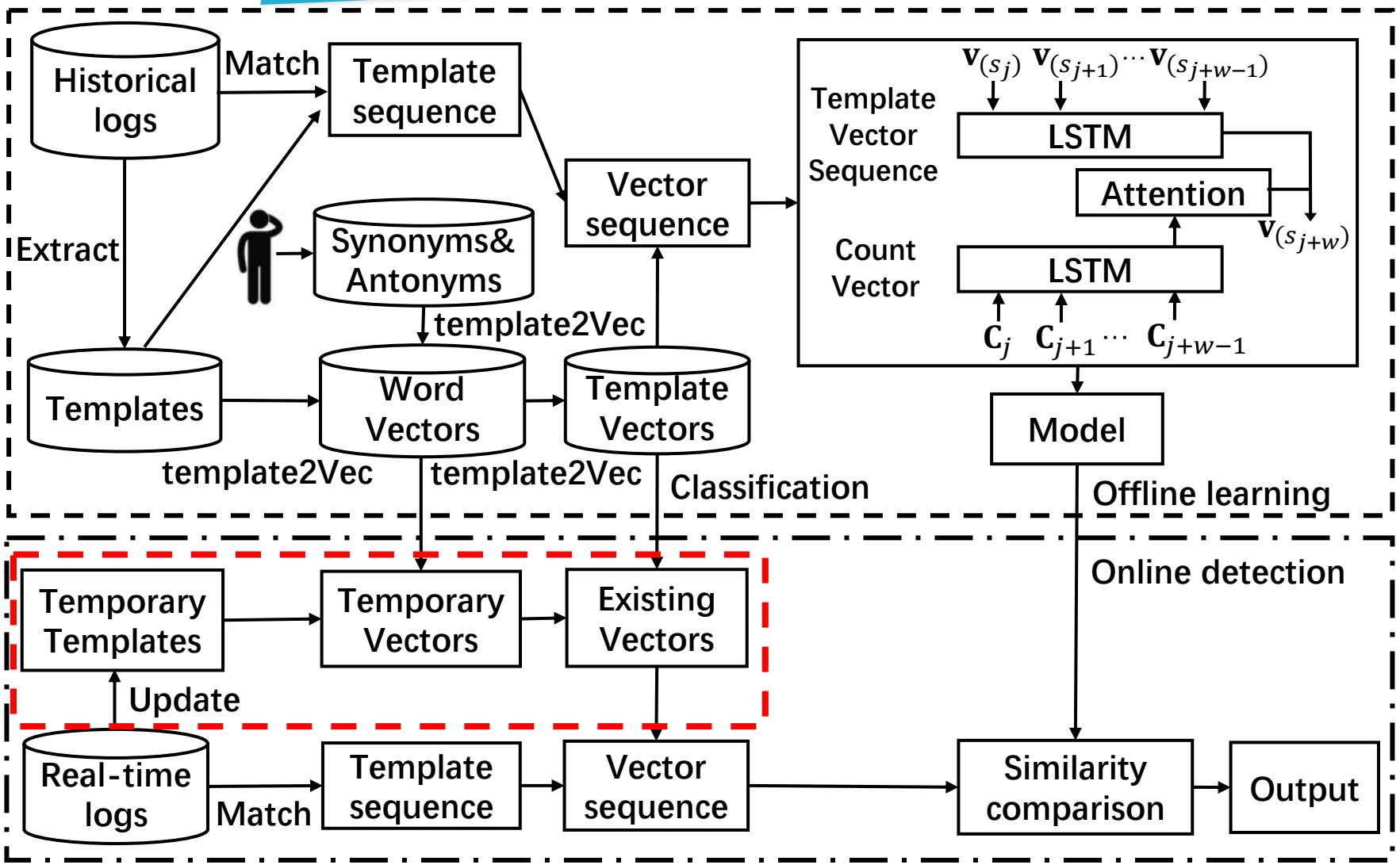
Calculate template vectors

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



[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.

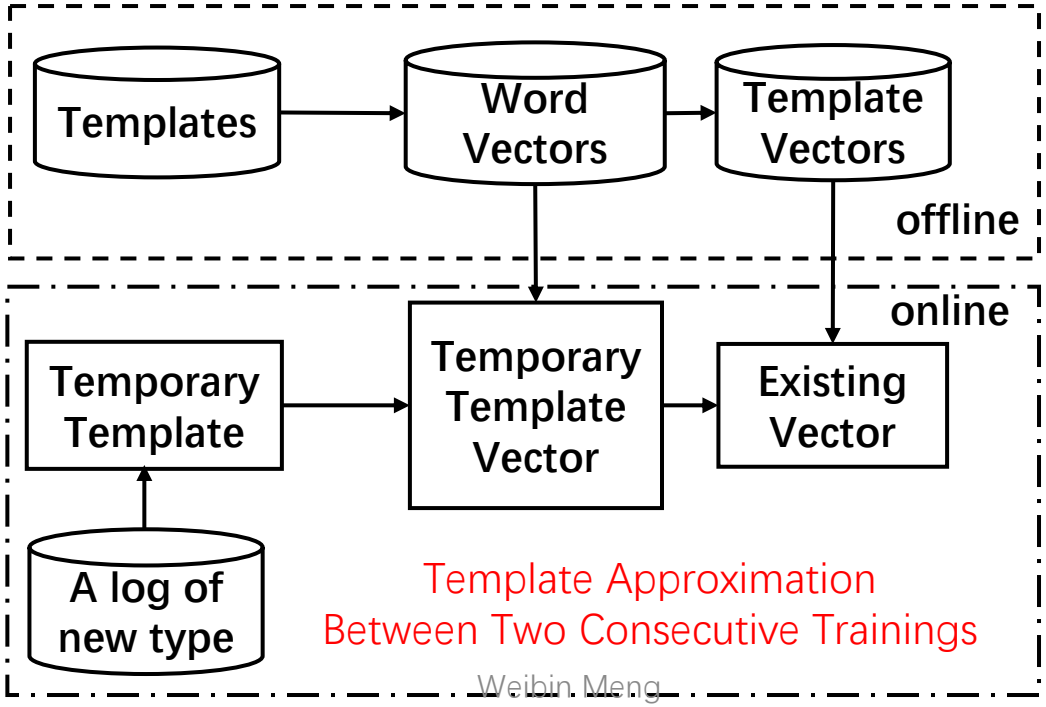
Template Approximation



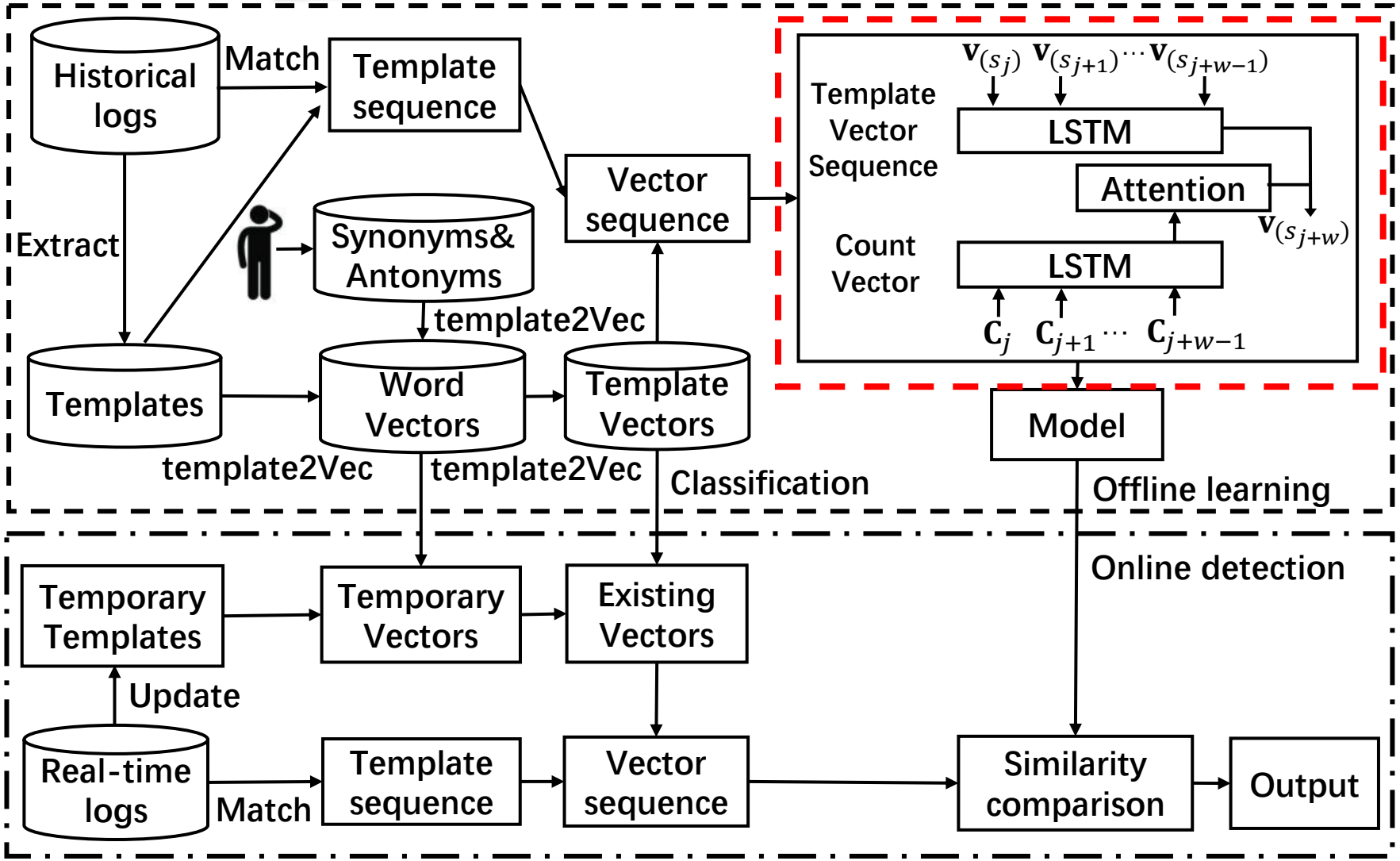
Template Approximation

Automatically merge new templates into existing templates

- Extract a template for the log of new type
- Generate a temporary template vector for the new template
- Mapping the temporary template vector into one of the existing vectors



Anomaly Detection



Anomaly Detection

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

	v_1	v_2	v_3	v_4
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

- Sequential pattern

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

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:

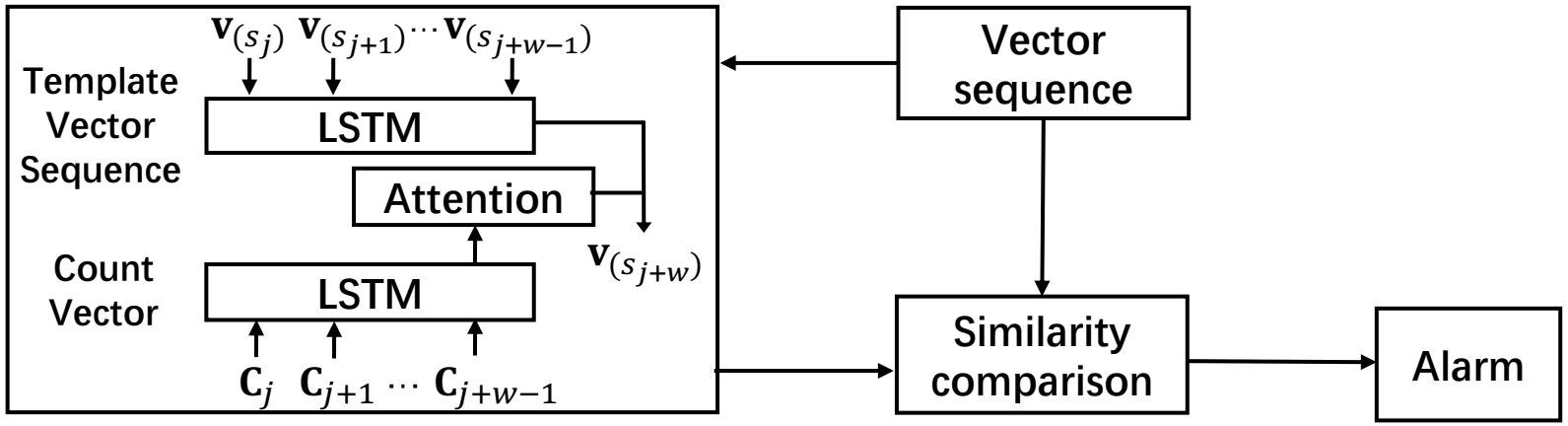
T_1 T_2 T_3 T_1 T_4 T_3

Conversions:

$L_1 \rightarrow T_1$
$L_2 \rightarrow T_2$
$L_3 \rightarrow T_3$
$L_4 \rightarrow T_1$
$L_5 \rightarrow T_4$
$L_6 \rightarrow T_3$

Combine Sequential and Quantitative relationship

- For a log sequence, we sort the possible next template vector based on their probabilities, which is learned according to the LSTM model.
- If the observed next template vector is included in the top k candidates (or similar enough), we regard it as normal.



Evaluation Datasets & Baselines

Datasets:

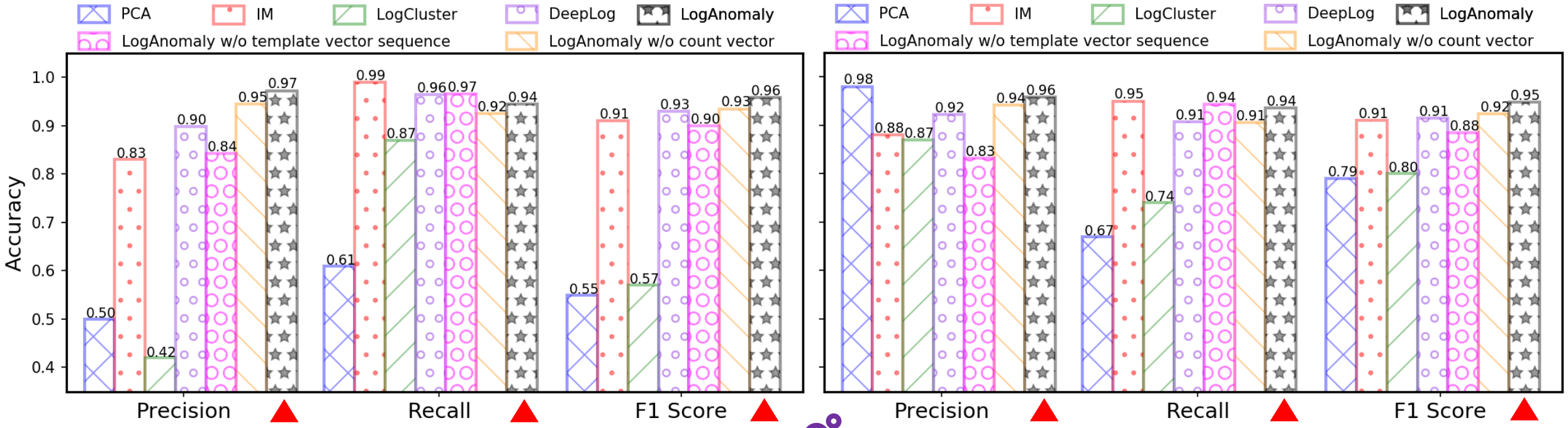
- BGL:
  - Generated by the Blue Gene/L supercomputer.
- HDFS:
  - Collected from more than 200 Amazon nodes.

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 Results



LogAnomaly achieves the best performance

Case Study

Dataset

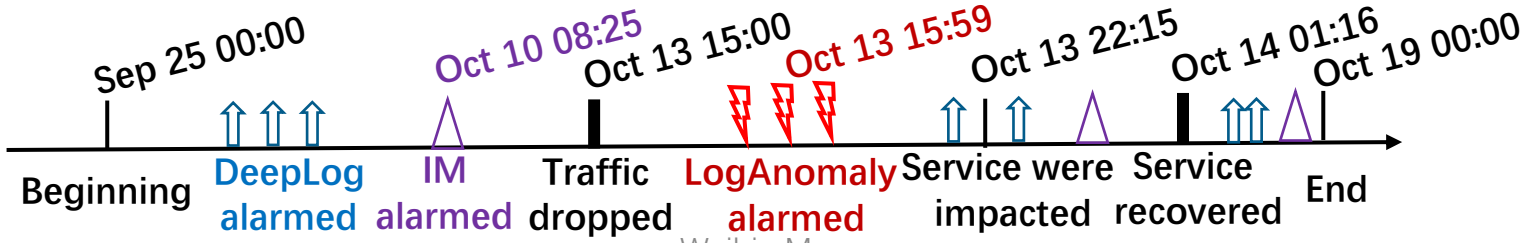
- Logs form an aggregation switch deployed in a top cloud service provider.

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
- LogAnomaly successfully detected it and generated no false alarm.



Conclusion

LogAnomaly

- Unsupervisedly detect quantitative and sequential log anomalies simultaneously

template2Vec

- Represent template without losing semantic information.

Merge templates of new types automatically

Evaluation and case study

Thanks

zhangsl@nankai.edu.cn

Evaluation of Online Detection

# template in training logs	# template in detection logs	# unmatched logs by training templates
251	523	299,174

Table 3: BGL dataset for online detection

Methods	Precision	Recall	F1 score
DeepLog	0.3817	0.9768	0.5489
LogAnomaly	0.8039	0.9319	0.8632

Table 4: Accuracy on online detection

Case in Intro

L_1. 1537885119 IFNET/2/linkDown_active(l):CID=0x807a0405, alarmID=0x0852003; The interface status changes.
L_2. 1537885119 LACP/4/LACP_STATE_DOWN(l): CID=0x804804, PortName=40GE1/0/3; The LACP state is down. Reason = **The interface went down physically**.
L_3. 1537885130 DEVM/3/LocalFaultAlarm_clear(l): CID=0x852003, clearType= service_resume, The local fault alarm has resumed.
L_4. 1537885135 IFNET/2/linkDown_clear(l): CID=0x807a0405, alarmID=0x0852003; The interface status changes. Physical link is up, mainName=Eth-Trunk104.

L_5. 1539139152 IFNET/2/linkDown_active(l):CID=0x807a0406, alarmID=0x0852007; The interface status changes.
L_6. 1539138152 LACP/4/LACP_STATE_DOWN(l): CID=0x804807, PortName=40GE1/0/3; The LACP state is down. Reason = **No LCAPDUs were received**.
L_7. 1539138164 DEVM/3/LocalFaultAlarm_clear(l): CID=0x852004, clearType= service_resume, The local fault alarm has resumed.
L_8. 1539138164 IFNET/2/linkDown_clear(l): CID=0x807a0406, alarmID=0x0852007; The interface status changes. Physical link is up, mainName=Eth-Trunk104.

Template2Vec