An Empirical Investigation of Practical Log Anomaly Detection for Online Service Systems

Nengwen Zhao, Honglin Wang, Zeyan Li, Xiao Peng, Gang Wang, Zhu Pan, Yong Wu, Zhen Feng, Xidao Wen, Wenchi Zhang, Kaixin Sui, Dan Pei

ESEC/FSE 2021 (Industry track)

Log Anomaly Detection

Log data is a valuable data source in online service systems, which records detailed information of system running status and user behavior.

```
09:37:53 INFO AllocateBlock
09:37:54 INFO Receiving block
09:37:54 INFO Receiving block
09:37:54 INFO Receiving block
….
09:38:49 WARN Redundant addStoredBlock
```

- Log anomaly detection: identify abnormal system behavior
- Assist engineers in identifying incidents promptly and diagnose incidents rapidly

Traditional Log Anomaly Detection

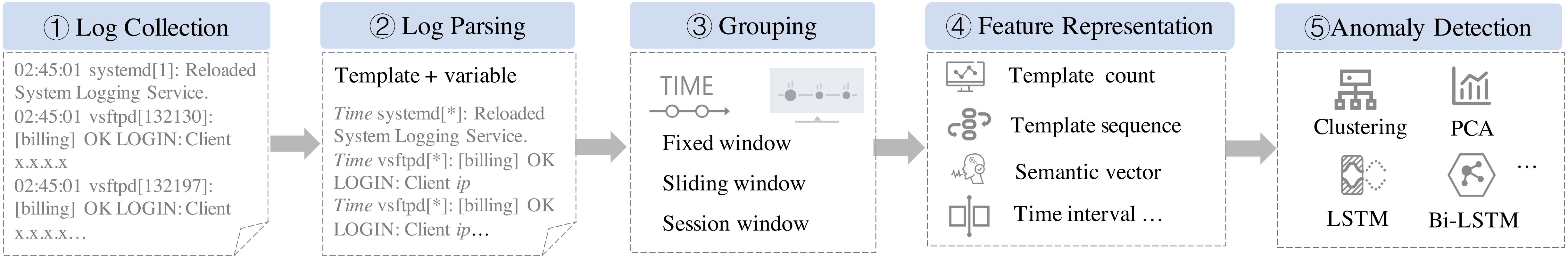
- Keywords and regular expressions

1. It is tedious to set manual rules for such numerous and various logs

2. Setting rules requires intensive domain knowledge, while the manpower of experienced engineers is limited.

3. Services are usually under frequent software changes. Thus the manual rules should be constantly updated and maintained.

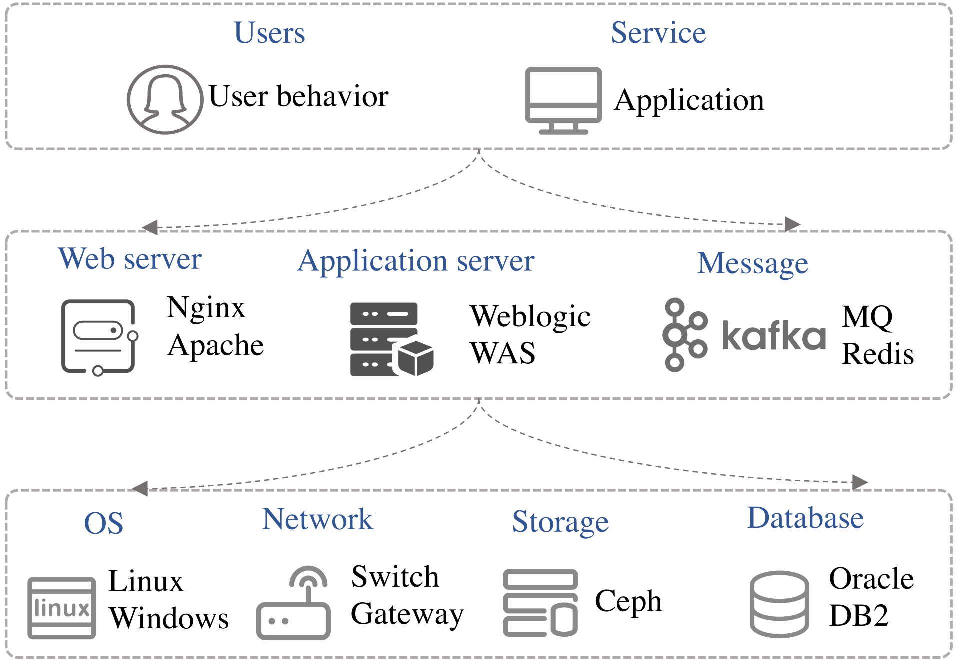
Pipeline of Log Anomaly Detection

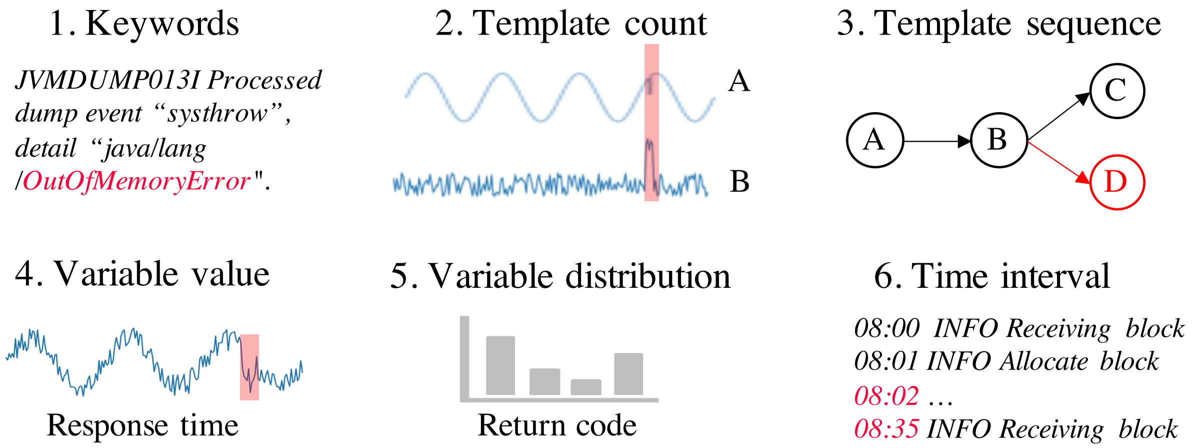


Pipeline of existing log anomaly detection approaches

Practical Challenges

1. Various logs and complex abnormal patterns

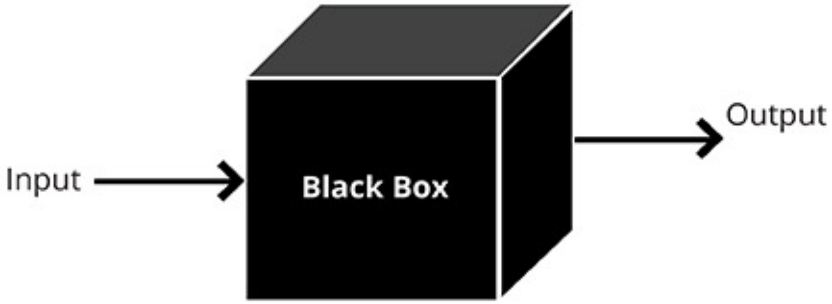




Practical Challenges

2. Poor interpretability

- Most of existing algorithms work as a “black box”.
- Engineers cannot gain any intuitive and actionable insights from the abstract results



3. Lack of domain knowledge

- Incorporating domain knowledge is necessary due to the variety of logs and abnormal patterns.
- Handle some special situations and ensure human-computer interactive log analysis, to enhance the performance and interpretability.

Empirical Study

Log anomaly detection in incident management

RQ1

What role do logs play in incident management?

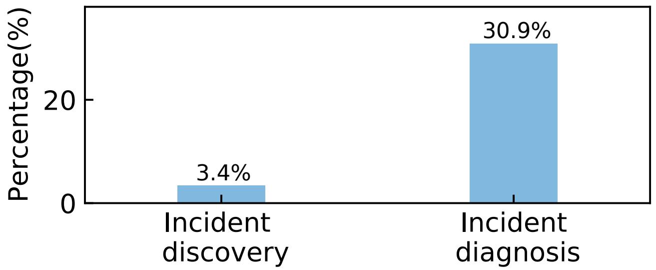
RQ2

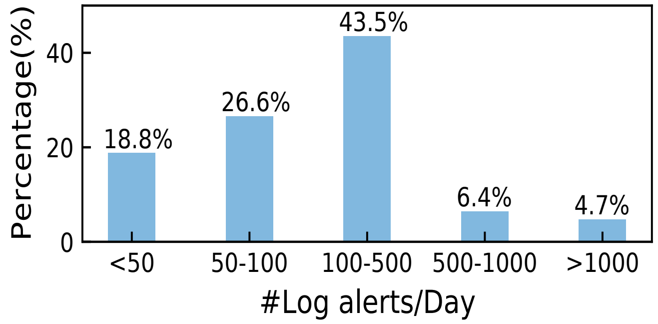
How about the performance of the current practice of log anomaly detection in the real world?

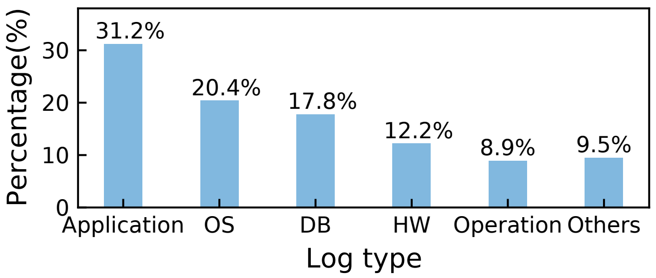
RQ3

What logs engineers frequently use in daily maintenance,
and what abnormal patterns may exist in log data?

Empirical Study







Type	**Example logs**	**Abnormal patterns**	**Incident scenarios**
Application	Application error	Template count (e.g., #“connection refused” increases), new log pattern (unknown error messages)	System unavailability, application error
Operation system	Linux, Windows, AIX	Keywords (e.g., “soft lockup” and “page allocation failure”), #total logs, template count (e.g., #“creating session” decreases)	OOM, node hang, hardware failures
Database	Oracle, DB2, MySQL	Keywords (e.g, “ORA-”, “Level: Error”), new log pattern	Database server down, performance degradation, SQL defect
Hardware (Network devices)	Switch, F5	Keywords (e.g., illegal state transition, address conflict detected), new log pattern, template count	IP address conflicted
Hardware (Storage)	Ceph	Keywords (e.g., “no reply”, and “time out”), new log pattern	Disk failure, space exhaustion
Distributed system	HDFS, OpenStack	Keywords (e.g., “IOException”), template sequence (wrong order of execution), new log pattern	Task failed, instance failure
Operation	User behavior	Template count, template sequence	Illegal operation, security issue
Middleware (Web server)	Nginx access, Apache access	Variable distribution (e.g., return code, and IP address), #total logs	System unavailability, illegal user access, security threat
Middleware (JVM GC)	CMS, G1, parallel	Keywords (“OutOfMemory”), template count (e.g.,#“FullGC” increases), variable value (GC time cost)	GC overhead limit exceeded, full heap space, memory leak
Middleware (Message queue)	IBM MQ	Keywords (e.g., “error occur”, “ended abnormally”, and “exception”), new log pattern	Message queue stuck

Empirical Study

1. Log anomaly detection is valuable for incident management.

2. Engineers tend to use keyword-based method in current practice due to better interpretability, while the performance is far from satisfying.

3. Log types and abnormal patterns are indeed diverse in practice. Thus a practical, effective and generic log anomaly detection approach is in urgent demand.

Experimental Study

The performance of existing approaches in practice

RQ1

How about the effectiveness of existing approaches on various real-world logs?

RQ2

How about the efficiency of existing approaches?

Dataset	Type	#Logs	#Pos/#Neg
*D*1	Application error	26,918	3/720
*D*2	Application error	84,139	7/1446
*D*3	User operation	9,080	2/38
*D*4	Nginx access	2,856,793	32/3036
*D*5	JVM GC (CMS)	217,613	13/398
*D*6	JVM GC (Parallel)	64,208	27/469
*D*7	DB2	16,133	2/74
*D*8	Linux system	771,083	4/768
*D*9	Linux system	3,227,843	5/1459
*D*10	Linux system	1,087,956	6/1288

Deatails of our experimental datasets

Effectiveness and Efficiency

Approach	PCA			LogCluster			IM			DeepLog			LogAnomaly		
Dataset	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
*D*1	0.40	1.00	0.57	0.38	1.00	0.55	0.51	1.00	**0.68**	0.64	0.33	0.44	0.45	0.67	0.54
*D*2	0.56	0.43	0.48	0.50	0.57	0.53	0.34	0.43	0.38	0.71	0.86	**0.78**	0.71	0.86	**0.78**
*D*3	0.67	0.50	0.57	0.72	0.50	0.59	0.43	1.00	**0.60**	0.43	0.50	0.46	0.62	0.50	0.55
*D*4	0.20	0.44	0.28	0.24	0.53	**0.33**	0.32	0.21	0.26	0.17	0.38	0.23	0.22	0.34	0.27
*D*5	0.92	1.00	**0.96**	0.80	1.00	0.89	0.87	1.00	0.93	0.91	1.00	0.95	0.91	1.00	0.95
*D*6	0.64	1.00	0.78	0.86	0.81	0.83	0.82	1.00	0.90	0.93	0.96	0.94	0.92	1.00	**0.96**
*D*7	0.42	0.50	0.46	0.44	0.50	0.47	0.58	0.25	0.35	0.57	1.00	**0.73**	0.53	1.00	0.69
*D*8	0.10	0.25	0.14	0.35	0.25	0.29	0.14	0.25	0.18	1.00	0.50	**0.66**	1.00	0.50	**0.66**
*D*9	0.18	0.60	0.28	0.28	1.00	0.44	0.44	0.40	0.42	0.32	1.00	0.48	0.54	0.60	**0.57**
*D*10	0.29	0.33	0.31	0.43	0.33	0.37	0.29	1.00	0.45	0.74	0.50	0.60	1.00	0.50	**0.67**
Mean F1 (Std)	0.48 (0.24)			0.53 (0.19)			0.52 (0.24)			0.63 (0.22)			0.66 (0.19)		

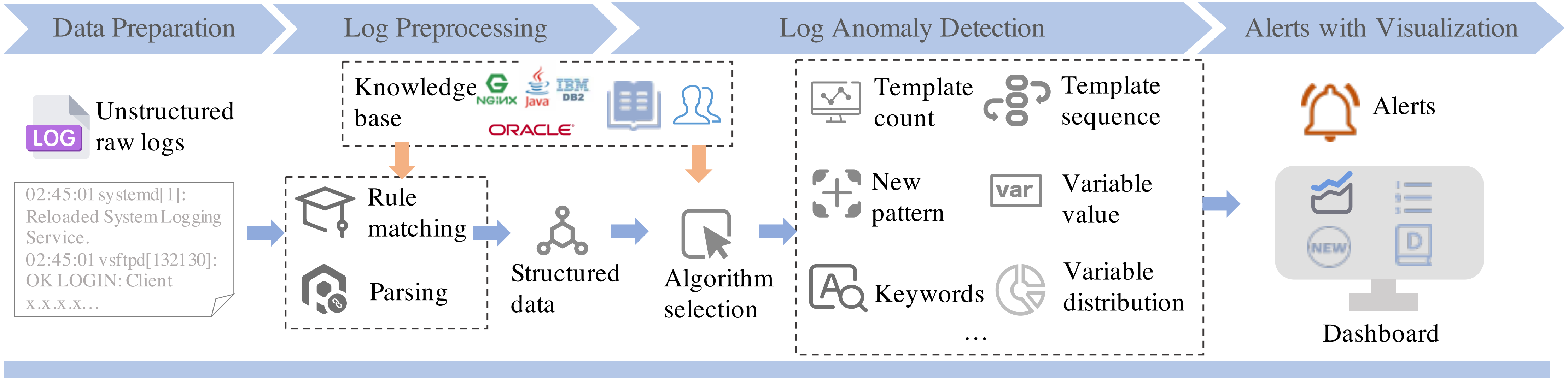
Approach	PCA	LogCluster	IM	DeepLog	LogAnomaly
Training (min)	0.89	5.40	4.60	34.67	48.54
Detection (s)	0.23	0.56	0.38	1.02	1.78

Summary

1. About effectiveness, existing approaches perform unstably on different datasets, mainly due to the variety of log abnormal patterns and the limitation of each approach.

2. About efficiency, all studied approaches could achieve satisfactory detection time, while deep learning based approaches require higher training time than statistical approaches.

LogAD



Data Preparation
Log Preprocessing
Log Anomaly Detection
Alerts with Visualization
LOG
Unstructured raw logs
02:45:01 systemd[1]: Reloaded System Logging Service.
02:45:01 vsftpd[13213 0]: OK LOGIN: Client x.x.x.x…
Knowledge base
NGINX
Java
IBM DB2
ORACLE
Rule matching
Parsing
Structured data
Algorithm selection
Template count
Template sequence
New pattern
var
Variable value
Keywords
Variable distribution
…
Alerts
NEW
D
Dashboard

Core Ideas

Various logs and abnormal patterns	Integrate multiple anomaly detection techniques
Poor interpretability	An intuitive alert report with good interpretability
Lack of domain knowledge	Import a knowledge base fusing expert experience

Template count
Template sequence
New pattern
Variable value
Keywords
Variable distribution
…

Alerts

Dashboard

Knowledge base
NGINX Java IBM DB2 ORACLE

Lessons Learned

There exists a gap between the research of algorithms and real application scenarios.

A single algorithm is usually not a panacea in practice.

Choose appropriate logs for anomaly detection.

Human-computer interactive log analysis.

Conclusion

We propose several significant practical challenges when applying log anomaly detection approaches in academic to practice.

We conduct the first empirical study and an experimental study based on real-world data and obtain several key observations, supporting these challenges.

We propose a generic log anomaly detection system named LogAD to tackle these challenges together.

Hope our work can provide some inspiration and guidance for practitioners and researchers to apply log anomaly detection to practice.

Thank you!

znw17@mails.tsinghua.edu.cn