



2023 CCF国际AIOps挑战赛决赛
暨“大模型时代的AIOps”研讨会

基于建行云龙舟运维平台的稳定性工具和 多维监控系统故障识别和故障分类

南开大学/王潇霏

主办单位：中国计算机学会（CCF）、清华大学、中国建设银行股份有限公司、南开大学

承办单位：中国计算机学会互联网专委会、清华大学计算机科学与技术系、中国建设银行股份有限公司运营数据中心、南开大学软件学院、北京必示科技有限公司

赞助单位：华为技术有限公司、国网宁夏电力有限公司电力科学研究院、软通动力信息技术（集团）股份有限公司

目录

CONTENTS

- 01 团队介绍
- 02 赛题解析
- 03 方案介绍
- 04 未来展望

01 团队介绍

D Parallel and Distributed
Software Technology Lab

NB Nankai-Baidu
Joint Laboratory



南开大学
Nankai University



● 参赛人员

王文蕊 (硕士研究生)

王潇霏 (博士研究生)

张家茗 (硕士研究生)

● 参赛单位

南开大学 (并行与分布式软件技术研究室)

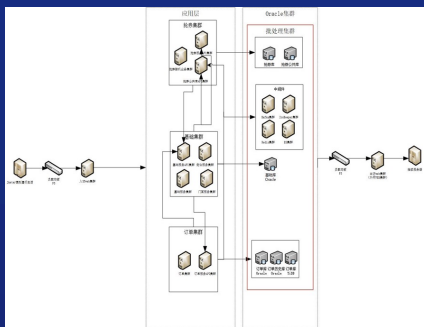
● 指导老师

王 刚 (教授、博导)

刘晓光 (教授、博导)

02 赛题解析 | 故障识别是故障根因定位和故障分类的基础

● 数据分析



部署架构图

Monitor数据

timestamp	cmd_id	log_name	value	device
169470760 ES_01		system.cpu.guest	0	
169470760 WebLogic_48		system.net.packets_out_error	0	eth0
169470760 ES_03		system.mem.free	4430	
169470760 WebLogic_48		system.io.mfile.current	4192	
169470760 ES_03		system.mem.cached	7004	
169470760 WebLogic_48		system.io.mfile.max	3267890	
169470760 WebLogic_48		system.io.mfile.used_pct	0.109	
169470760 WebLogic_48		system.processes.gc.InfoCount.PGC	1	
169470760 ES_03		system.mem.buffered	3	
169470760 ES_03		system.load.avg.5	0.01	
169470760 WebLogic_48		system.processes.gc.InfoPercent.E	23.92	
169470760 ES_03		system.load.avg.15	0.01	
169470760 ES_03		system.load.avg.1	0.01	
169470760 ES_03		system.mem.pct_usage	24.95	
169470760 WebLogic_48		system.processes.gc.InfoPercent.O	5.46	
169470760 WebLogic_48		system.processes.gc.count	1	
169470760 ES_03		system.load.1	0.01	
169470760 ES_03		system.load.15	0.04	
169470760 ES_03		system.load.1	0.04	

Log数据

```

4474:s 1694534438 * 10000 changes in 60 seconds. Saving...
4474:s 1694534438 * Background saving started by pid 18761
18761:c 1694534438 * DB saved on disk
18761:c 1694534438 * RDB: 7 MB of memory used by copy-on-write
4474:s 1694534438 * Background saving terminated with success
4474:s 1694534504 * 10000 changes in 60 seconds. Saving...
4474:s 1694534504 * Background saving started by pid 19023
19023:c 1694534504 * DB saved on disk
19023:c 1694534505 * RDB: 7 MB of memory used by copy-on-write
4474:s 1694534505 * Background saving terminated with success
4474:s 1694534579 * 10000 changes in 60 seconds. Saving...
4474:s 1694534579 * Background saving started by pid 19305
19305:c 1694534579 * DB saved on disk
19305:c 1694534579 * RDB: 7 MB of memory used by copy-on-write
    
```



异常检测

如何借助Monitor和Log数据
高效识别异常信息并进行分类?

TC数据

tran_code	timestamp	amount	bus_success_rate	sys_success_rate	avg_rsp_time	stall	amount_avg	proc_time	stall_rate	apdex
A0018	1694620800	206	100	100	10	0	10	0	100	
A0005	1694620800	202	100	100	10	0	2	0	100	
A0004	1694620800	204	100	100	10	0	4	0	100	
A0010	1694620800	99	100	100	27	0	8	0	100	
A0007	1694620800	23	100	100	2	0	2	0	100	
A0001	1694620800	50	100	100	4	0	2	0	100	
A0022	1694620800	158	100	100	1	0	1	0	100	
A0023	1694620800	2	100	100	30	0	30	0	100	
A0017	1694620800	205	100	100	3	0	3	0	100	
A0014	1694620800	201	100	100	2	0	2	0	100	
A0008	1694620800	400	100	100	2	0	2	0	100	
P0002	1694620800	149	100	100	0	0	2	0	100	
P0015	1694620800	205	100	100	1	0	1	0	100	
P0005	1694620800	17	100	100	1	0	1	0	100	
P0008	1694620800	1	100	100	2	0	2	0	100	
P0013	1694620800	14	100	100	1	0	1	0	100	

Apptesting数据

vsId	biId	appId	ipId	appId	deployEnv	traceId	cost	status	reason	biId	parentFlowName	tenantId	attr	thrt	svfCost	startCost	timestamp	duration	cmd_id	
0x000	4444444444444444	0x000	0	0x00000000	0	INTERNAL	99174847	0	UNSET	99174847	Handler	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_21
0x000	4000e00040004000	App0008	46680e341	2	CLIENT	644477521877	POST	3735ac67c1	code	dur	2	OK	1	1698-09	2	WebLogic_23				
0x000	5ac1871091091091	App0008	2300070e6	26	INTERNAL	58611475101	ContentRoll	3735ac67c1	code	dur	20	UNSET	1	1698-09	20	WebLogic_25				
0x000	1010600040004000	App0008	2300070e6	0	CLIENT	5ac1871091	GET	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_23				
0x000	4e4a391481481481	App0008	1d7010e4e	0	CLIENT	3897196671	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_19				
0x000	8a7b7a7a7a7a7a7a	App0008	44910e4e	0	CLIENT	4700c0c0c0	GET	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_19				
0x000	8a7b7a7a7a7a7a7a	App0011	e680a631a	0	INTERNAL	676170061	Handler	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_11				
0x000	App0011	0d970e103	1	CLIENT	6300600000	SELECT	3735ac67c1	code	dur	1	UNSET	1	1698-09	1	WebLogic_11					
0x000	4000000000000000	App0008	0e7c3ba7c	0	INTERNAL	605497200	Handler	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_20				
0x000	380a31404242189	App0008	0e7c3ba7c	0	CLIENT	544544888	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_20				
0x000	37c16a6b6b6b6b6b	App0008	0e7c3ba7c	0	CLIENT	544544888	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_20				
0x000	644b3548391004	App0008	0e7c3ba7c	12	CLIENT	544544888	TEXT	3735ac67c1	code	dur	12	OK	1	1698-09	12	WebLogic_20				
0x000	19117343712605	App0008	44910e4e	0	CLIENT	4660000000	SELECT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_20				
0x000	3413b36868686868	App0008	44910e4e	0	CLIENT	4733179718	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_20				
0x000	33a3a50409c71a	App0008	44910e4e	2	CLIENT	4733179718	TEXT	3735ac67c1	code	dur	2	OK	1	1698-09	2	WebLogic_20				
0x000	09c0ee17a0a0a0	App0008	44910e4e	11	INTERNAL	4733179718	TEXT	3735ac67c1	code	dur	11	OK	1	1698-09	11	WebLogic_20				
0x000	639012000a13	App0008	44910e4e	0	CLIENT	4733179718	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_20				
0x000	4f40000040107e	App0008	50e1145c	0	CLIENT	4301000000	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_25				
0x000	47541088700044	App0008	40520a771	0	CLIENT	605545248	PING	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_19				
0x000	6730ceef4921028	App0008	60702a771	0	CLIENT	801c1f04c	TEXT	3735ac67c1	code	dur	0	UNSET	1	1698-09	0	WebLogic_19				



根因定位

结合TC和Apptesting数据进行故障根因定位?

难点1

- 数据量大，数据高维稀疏，数据类型差异大。

应对措施

- 特征筛选，特征转换等方式降低数据维度。
- 不同类型的数据采取不同的处理方式。
- 通过数据拆分降低数据稀疏程度。
- 查看数据分布，采用分布一致的数据进行建模。

难点2

- 单指标故障识别容易受数据波动影响。

应对措施

- 方案针对多指标联合识别。

基于建行云龙舟运维平台的稳定性工具和多维监控系统故障识别和故障分类



难点3

- 损耗性硬件故障的发生并不属于突发事件。

应对措施

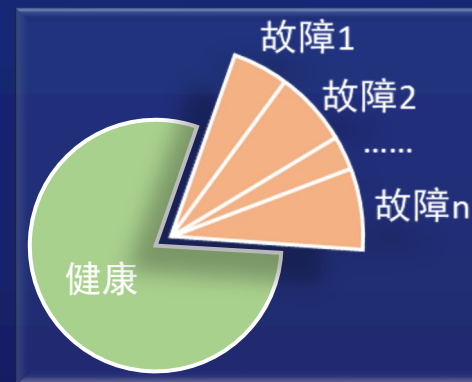
- 方案设置时间窗口进行故障识别。

难点4

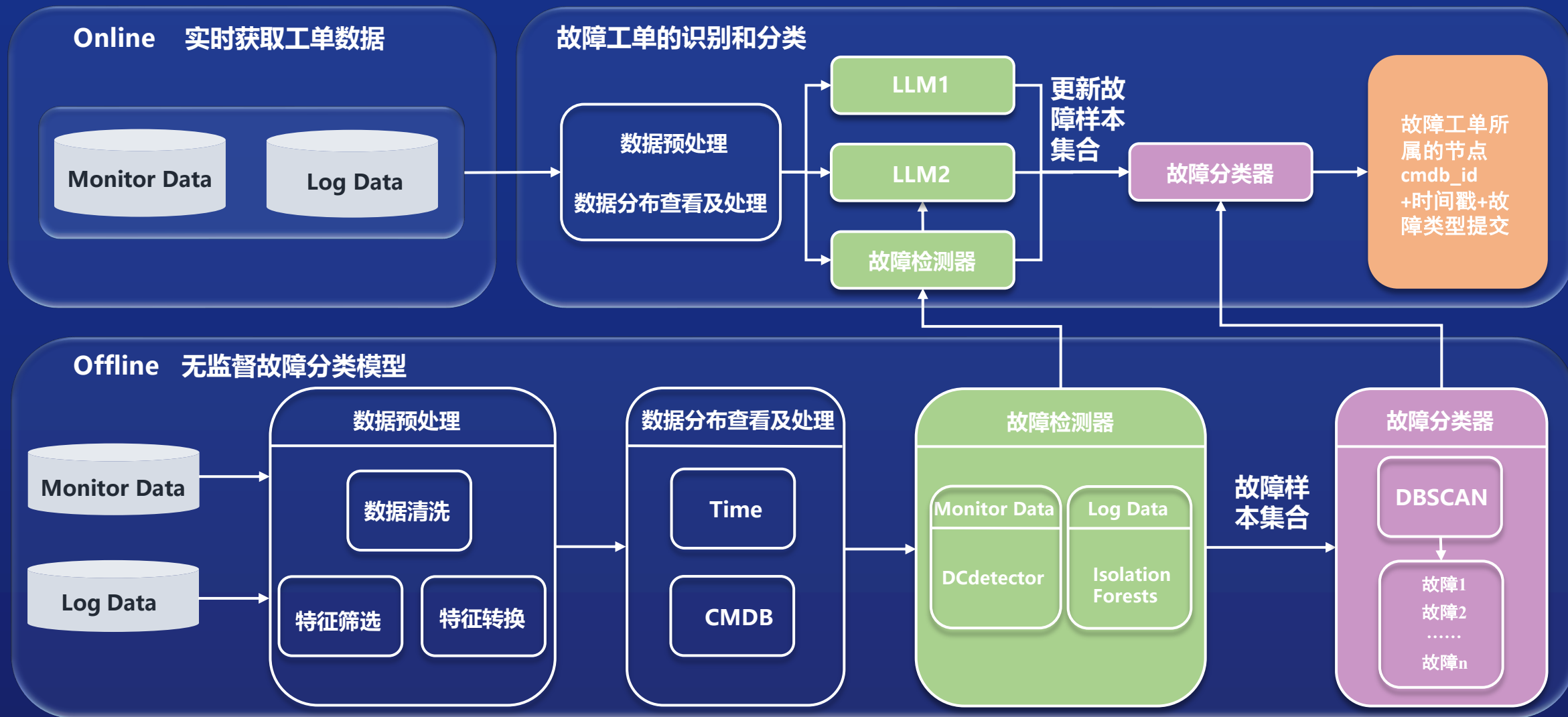
- 同类型故障区分难度较大。

应对措施

- 借助聚类方法进行故障分类。



03 方案介绍 | 基于LLM，采用ID²进行故障检测和故障多分类

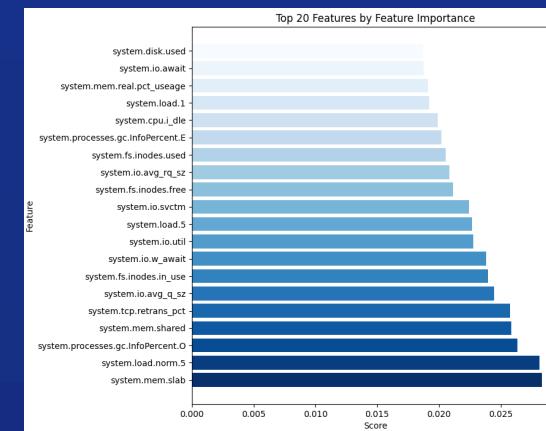
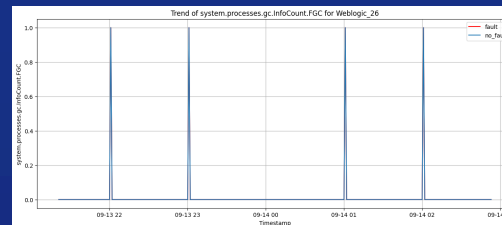
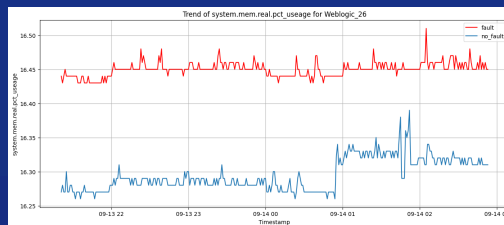


ID²_LLM (Isolation Forests_DCdetector_DBSCAN_LLM) 总体框架图

03 方案介绍 | 通过数据筛选和归一化操作保证数据质量

数据分析

- 特征对于异常数据识别的敏感性不同。

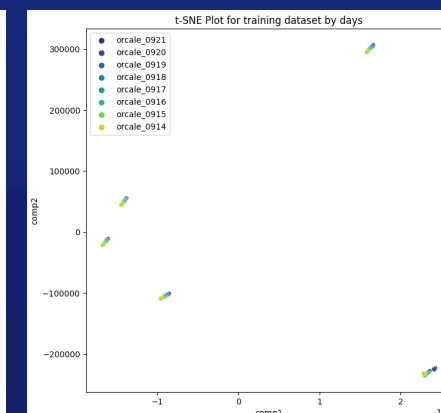
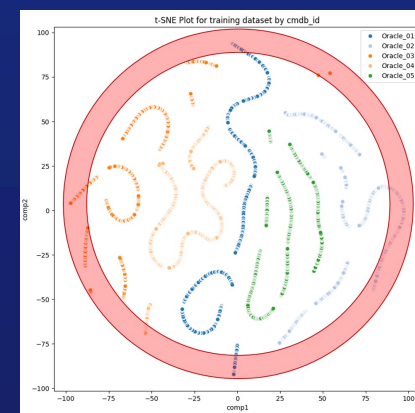
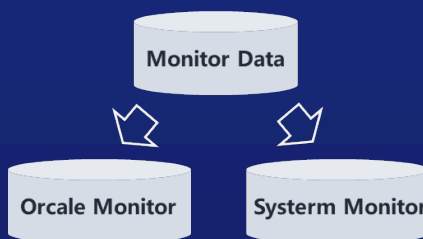
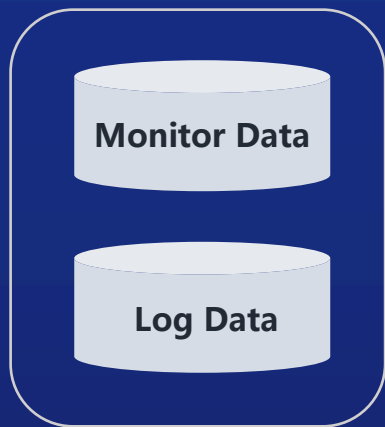


数据筛选

- 数据拆分：将Monitor数据进行拆分，降低数据稀疏性。
- 数据筛选：分别以Time和CMDB为单位筛掉分布不一致的数据。

按列归一化

- 指标值按列归一化：指标值之间具有数量级的差异。



特征筛选

- 去除冗余特征：针对常量指标。

```
4474:S 1694534438 * 10000 changes in 60 seconds. Saving...
4474:S 1694534438 * Background saving started by pid 18761
18761:C 1694534438 * DB saved on disk
18761:C 1694534438 * RDB: 7 MB of memory used by copy-on-write
4474:S 1694534438 * Background saving terminated with success
4474:S 1694534504 * 10000 changes in 60 seconds. Saving...
4474:S 1694534504 * Background saving started by pid 19023
19023:C 1694534505 * DB saved on disk
19023:C 1694534505 * RDB: 7 MB of memory used by copy-on-write
4474:S 1694534505 * Background saving terminated with success
4474:S 1694534579 * 10000 changes in 60 seconds. Saving...
4474:S 1694534579 * Background saving started by pid 19305
19305:C 1694534579 * DB saved on disk
19305:C 1694534579 * RDB: 7 MB of memory used by copy-on-write
```



Timestamp	Ten Thousand Changes Times	Growth Times	RDB Times
1694534450	1	0	1
1694534612	2	0	2
1694534773	3	0	3
1694534884	3	1	3
1694534985	4	1	4

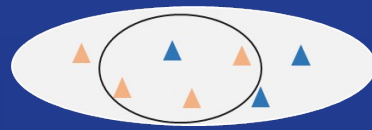
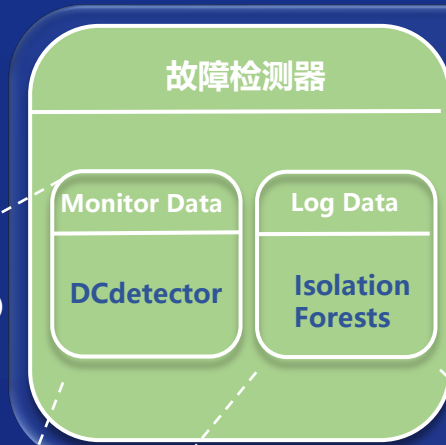
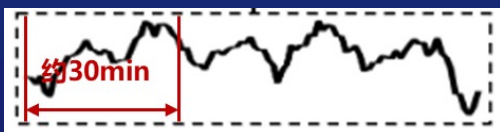
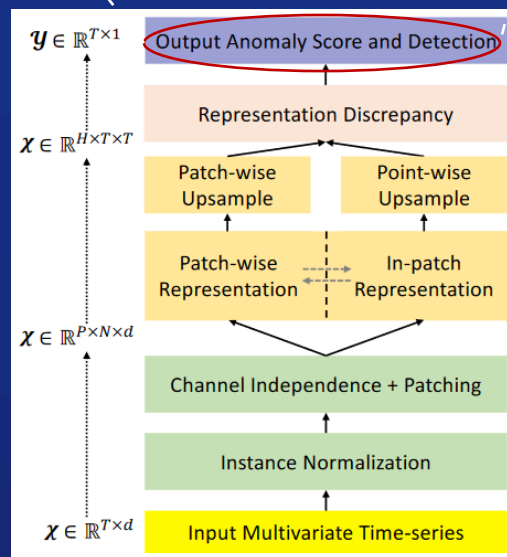
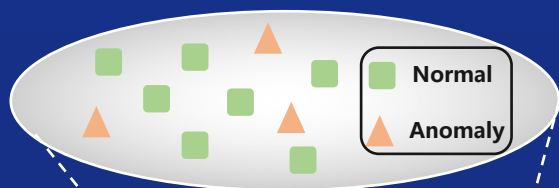
特征构造

- 正则匹配：利用正则表达式解析Log数据。
- 特征构造：基于统计学的方法构造差分特征。

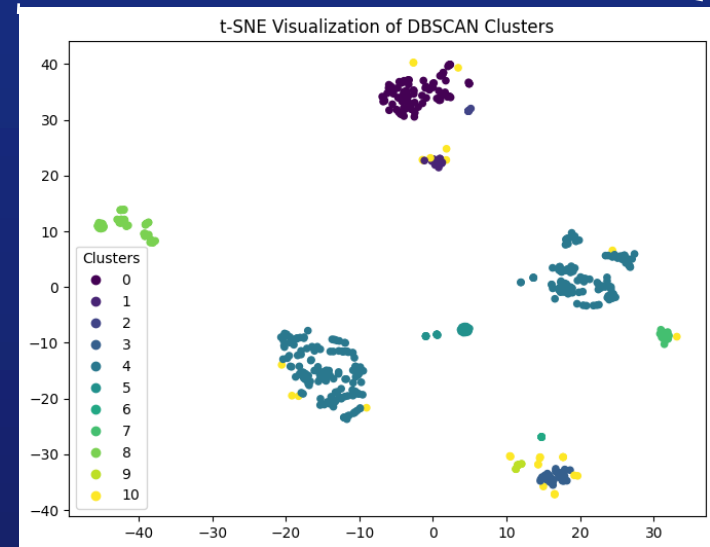
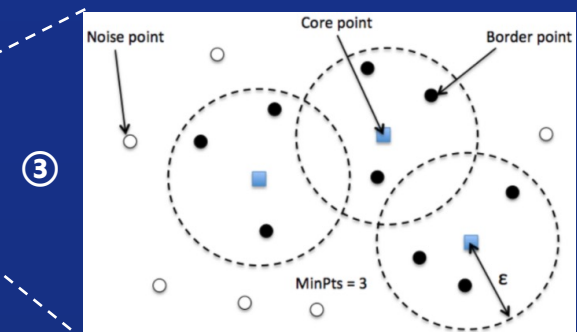
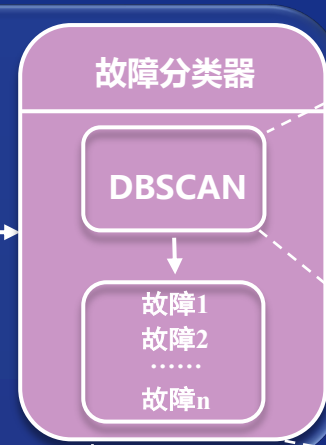
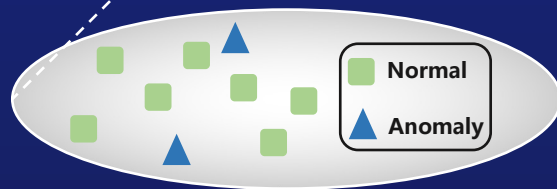
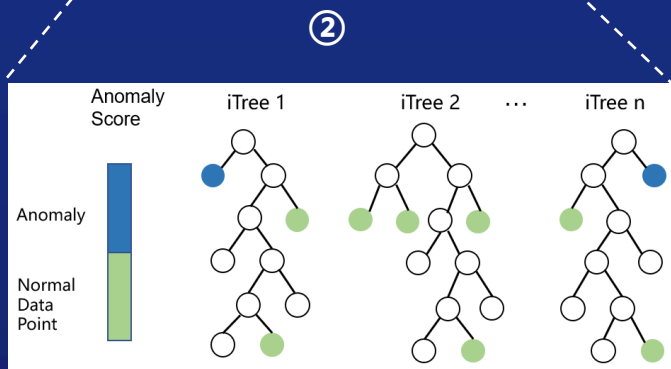


Timestamp	AOF Rewrite Memory Times	AOF Diff Memory Times	AOF Flush Memory Times	Timestamp_Diff
1694534450	0	0	0	0.0
1694534612	0	0	0	162.0
1694534773	0	0	0	161.0
1694534884	1	1	1	111.0
1694534985	1	1	1	101.0

03 方案介绍 | offline故障识别和故障分类



故障样本集合

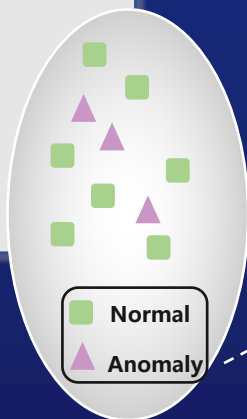


03 方案介绍 | online故障识别和故障分类

```
<|im_start|>system
You are a helpful
assistant.<|im_end|>
<|im_start|>user
You are an experienced operations
engineer. You will receive some
server data and find patterns in the
data to determine whether the
server is damaged and provide the
reasons. You can only diagnose the
status of the server, and should not
diagnose other hardware such as the
RAM and CPU. You should output
them in the required format as JSON.
```

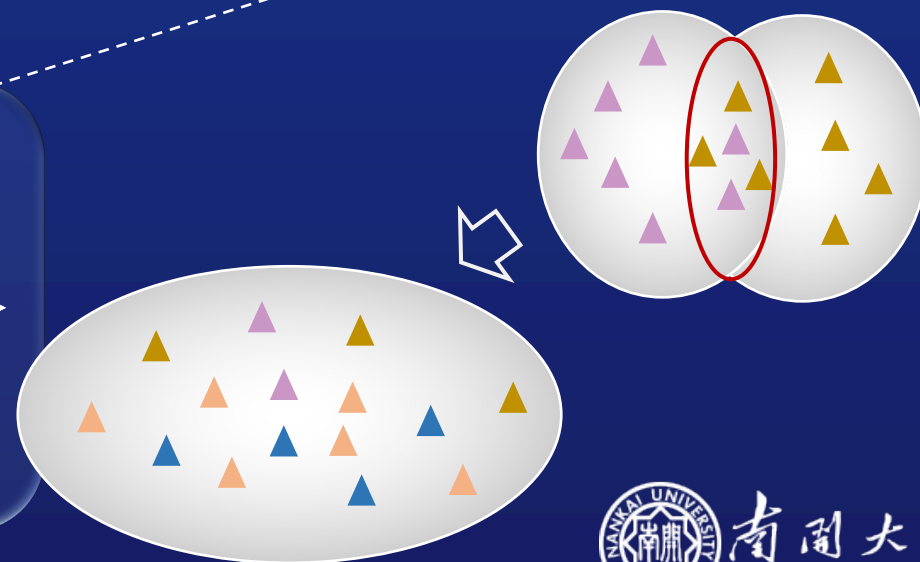
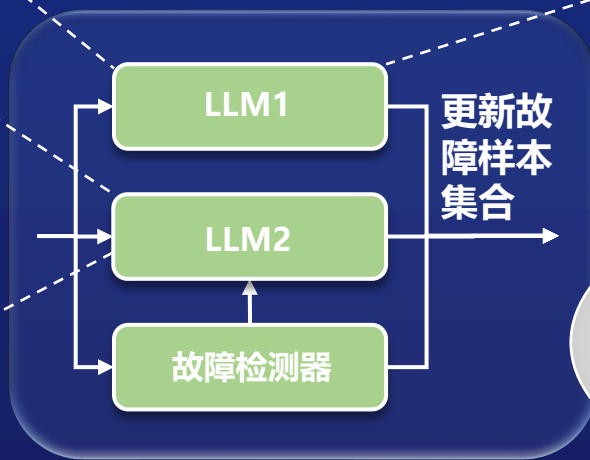
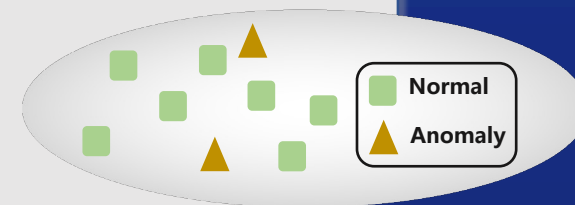
```
-- A MALFUNCTION NODE EXAMPLE
{malfunctional example}
-- A FUNCTIONAL NODE DATA
EXAMPLE
{functional example}
```

```
-- OUTPUT REQUIREMENT
{format_instructions}
-- OP DATA:
{disk_data}
-- YOUR JSON RESPONSE:
<|im_end|>
<|im_start|>assistant
```



```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
You are an experienced operations engineer. You will receive some server data and
find patterns in the data to determine whether the server is damaged and provide the
reasons. You can only diagnose the status of the server, and should not diagnose
other hardware such as the RAM and CPU. You should output them in the required
format as JSON.
```

```
-- OUTPUT REQUIREMENT
{format_instructions}
-- NODE DATA:
{disk_data}
-- YOUR JSON RESPONSE:
<|im_end|>
<|im_start|>assistant
```



● 方案的创新性和实用性

数据

- 本方案使用的Monitor和Log数据相对易于获取，则思路更容易被其他生产商借鉴去进行故障的识别和分类。
- 本方案对考虑了数据分布对模型性能的影响，该思路为后续故障预测类任务提供了新的方向。

模型

- 本方案故障检测和分类均为无监督算法，完全节省了人工标注数据的成本。
- 本方案在测试阶段借助了大模型的优势更新故障样本集合，实现了智能化的故障检测。
- 本方案的DCdetector创新性维护了一个故障检测窗口，考虑了硬件故障发生不属于突发事件，可做到提前预测。
- 本方案针对不同业务场景、不同数据只需要微调便可快速适应。

● 改进思路

数据

- 数据处理及特征工程阶段存在人工干预，并未完全实现自动化。

模型

- 故障检测阶段各对象之间独立完成，后续可以考虑将对象之间的动态拓扑关系融入该阶段。
- 故障分类模型后续可进一步定位故障根因。
- 大模型后续可尝试prompt tuning。



2023 CCF国际AIOps挑战赛决赛暨“大模型时代的AIOps”研讨会

THANKS