Anomaly Detection and Troubleshooting of Large Scale Systems from Event Logs

Presented By Niloy Ganguly

Bivas Mitra, Subhendu Khatuya
Also in collaboration with NetApp

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

- Prerequisite
- Dataset
- Objective
- Challenges
- Model Development
  - Anomaly detection framework
  - Building an automated troubleshooter
- Results

- **Prerequisite**

- Dataset

- Objective

- Challenges

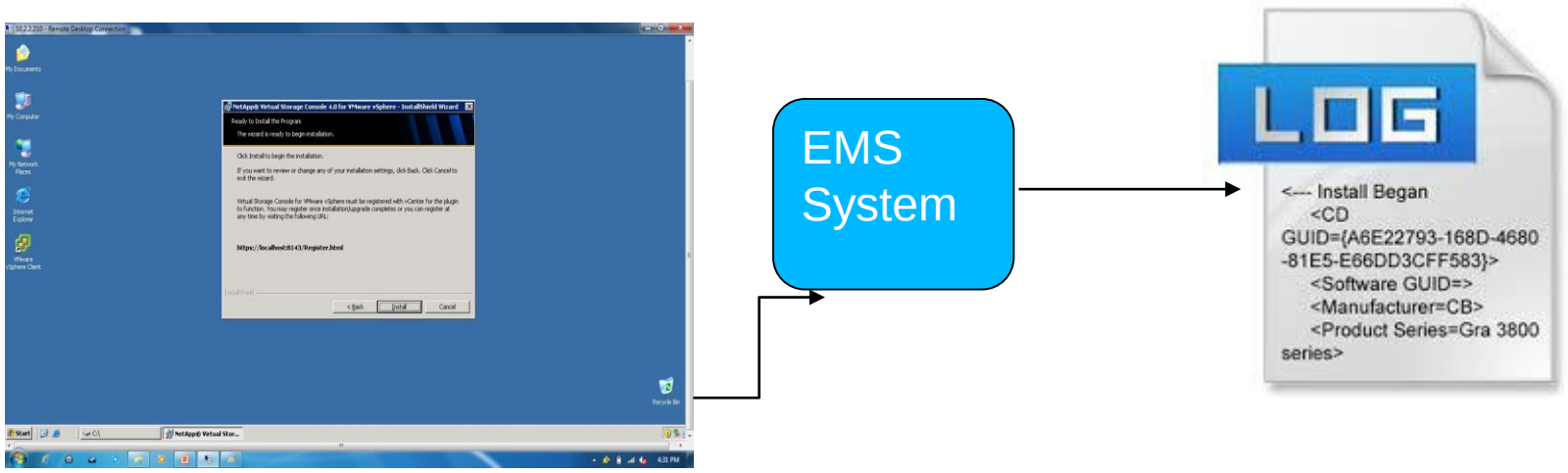
- Model Development
  - Anomaly detection framework
  - Building an automated troubleshooter

- Results

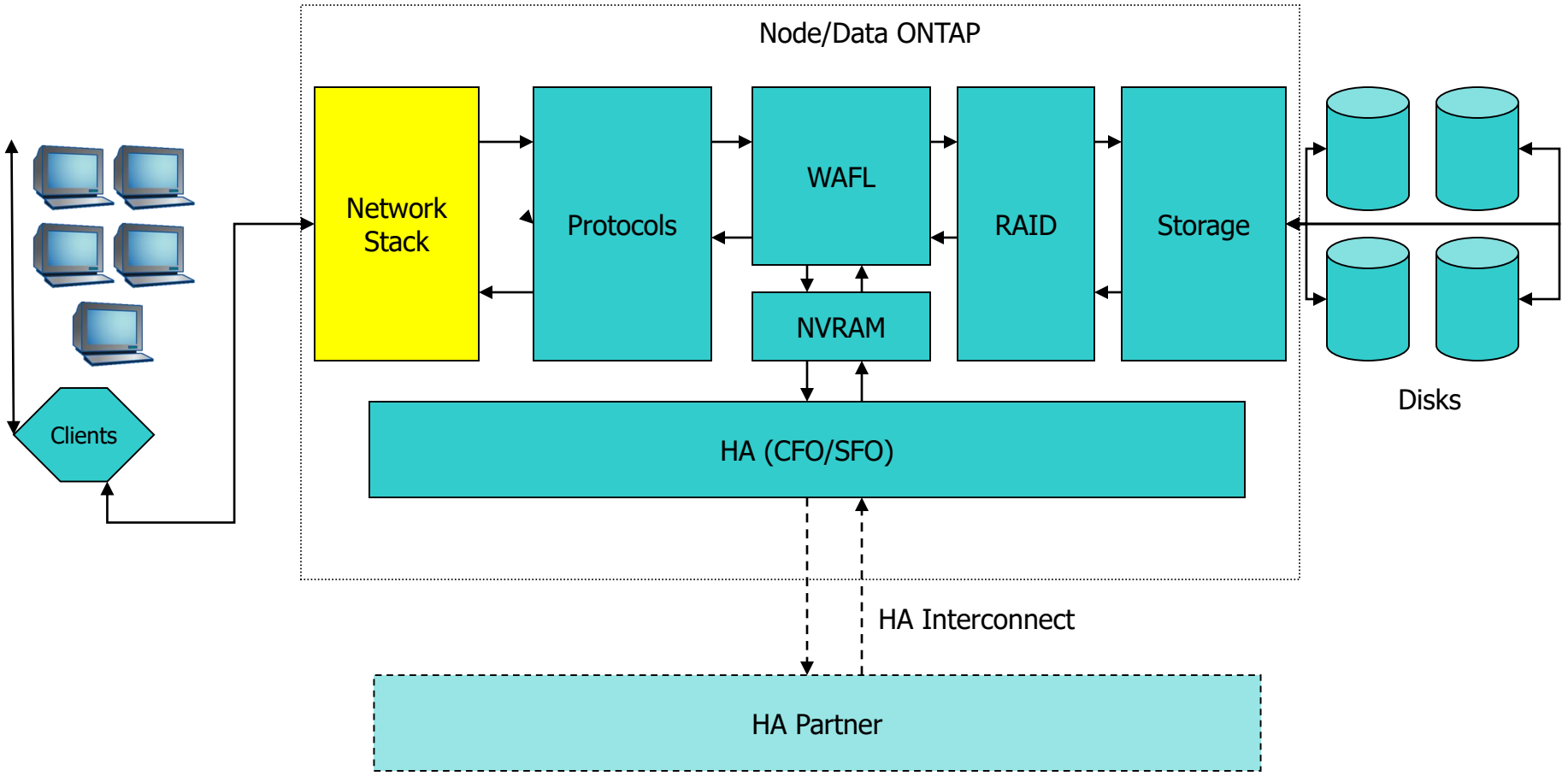
Prerequisite

EMS: Event Message System

- EMS supports a **built-in logging facility** that logs all activities on storage appliance done by customer.
- The system writes out event indication descriptions using a generic **text-based log** format.

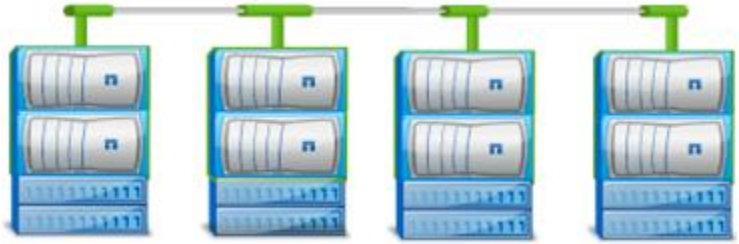


ONTAP Components

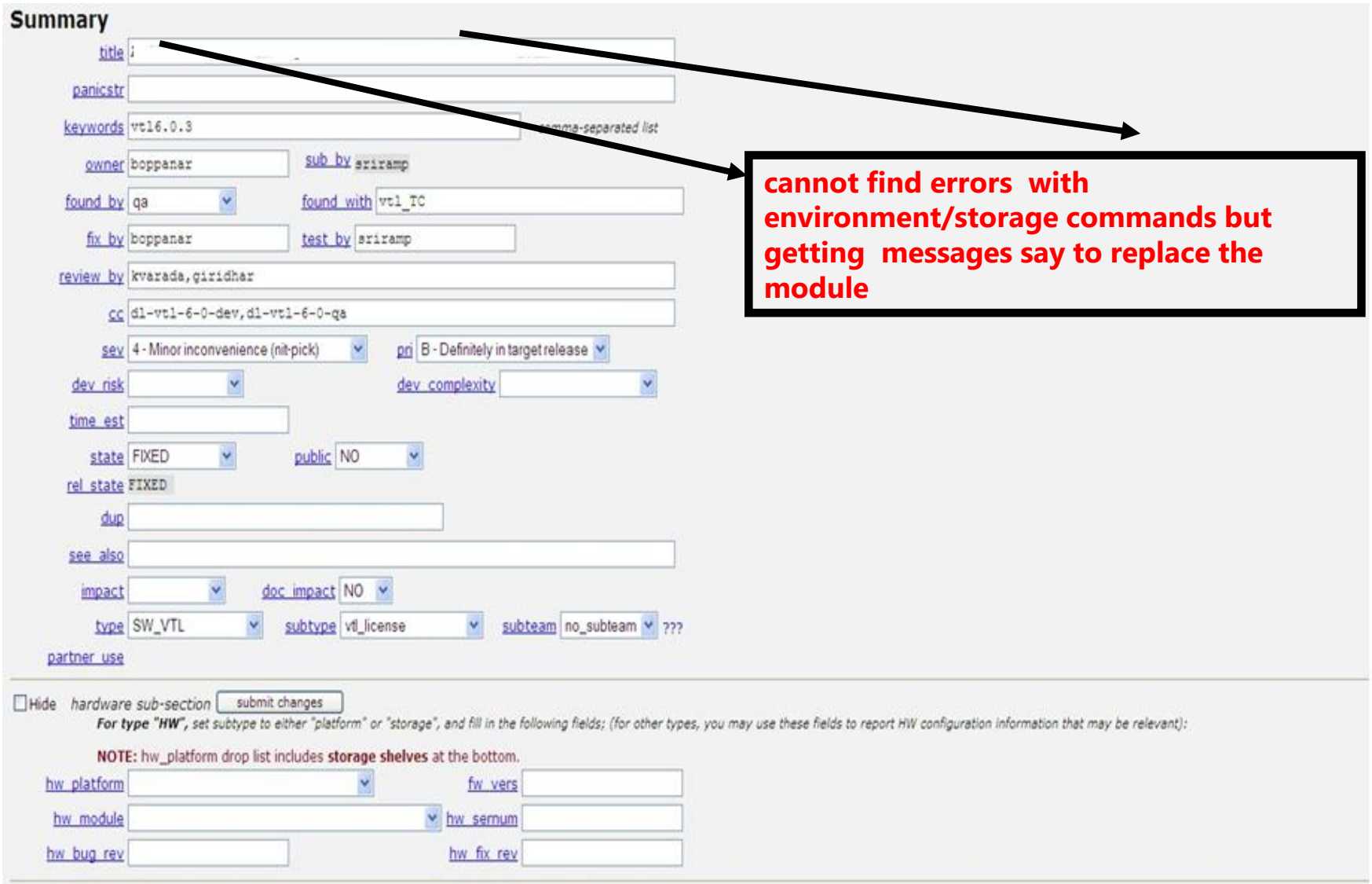


Prerequisite

Case:



Case Filed



Summary

title:

panicstr:

keywords: vtl6.0.3 *comma-separated list*

owner: boppanar sub_by: sriramp

found_by: qa found_with: vtl_TC

fix_by: boppanar test_by: sriramp

review_by: kvarada,giridhar

cc: dl-vtl-6-0-dev,dl-vtl-6-0-qa

sev: 4 - Minor inconvenience (nit-pick) pri: B - Definitely in target release

dev_risk: dev_complexity:

time_est:

state: FIXED public: NO

rel_state: FIXED

dup:

see_also:

impact: doc_impact: NO

type: SW_VTL subtype: vtl_license subteam: no_subteam ???

partner_use

Hide *hardware sub-section* submit changes

***For type "HW"**, set subtype to either "platform" or "storage", and fill in the following fields; (for other types, you may use these fields to report HW configuration information that may be relevant):*

NOTE: hw_platform drop list includes **storage shelves** at the bottom.

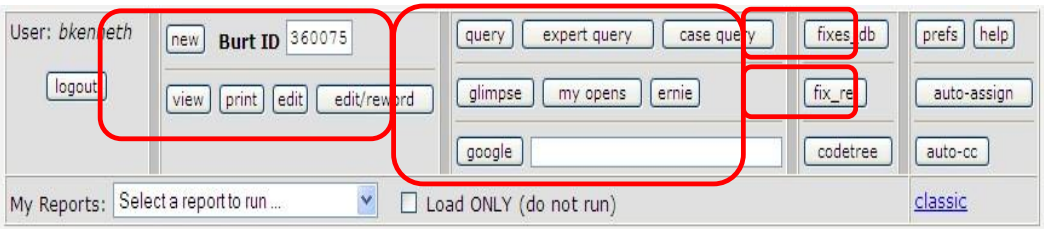
hw_platform: fw_vers:

hw_module: hw_sernum:

hw_bug_rev: hw_fix_rev:

cannot find errors with environment/storage commands but getting messages say to replace the module

Snapshot of a BURT



show/hide reload **Select:** ○ hide all ○ show all save hide settings as prefs

Hide Short_Fields: ☐ date ☐ external ☐ hardware ☐ internal ☐ mecheng ☐ partner ☐ release ☐ release_workflow

Show empty Short_Fields: ☐

[Description] [Configuration] [Bug_Signature] [Notes]
[Rel_Notes] [Attachments] [Cores] [Public_Report]
[Escalation_Status] [RCA_Notes] [Fix_Report] [Fixes_Database]
[Code_Review_Notes] [Unit_Test_Plan] [Unit_Test_Results] [Test_Plan]
[Pubs_Notes] [Change_Log]

```
===== Short_Fields ==================================================
#dbver 68
id                      360075
title                   Perfstat to collect snapmirror show and job show command output in BR
state                   UNTESTED
#rel_state              UNTESTED
public                  NO
doc_impact              NO
sev                     3
pri                     B
owner                   neetug
sub_by                  vdaga
bug_rel                 TOOLS
impact                  SUPPORT
type                    TOOLS
subtype                 perftool_perfstat2
#partner_use
subteam                 129
#info only              subteam name:  129 [BRAT]
```

Post Case Info

SmartSolve
NetApp Cluster Viewer | Try OneCollect

View | Tools | Templates | TRO Templates | Technica

Case Info

RSE	2005864084
Case Type	Technical Case (ZCAS)
Case Owner	Stephen Smitas
Priority	P2
Status	Closed
Symptom	RPANIC: 5:Associated error message c...
Category 1	File System
Category 2	Data ONTAP/WAFL
WAFL Problems	No available ASUPs
AutoSupport	View Autosupport
History of Controller Changes NEW!	0 available
Agreement Number	10915702
Support Offering	Premium/4HRPRMONS
ASUP Status	ON
Hostname	NHC3-WHT-FAS002
System Id	1892876249
Serial No	700001254159 | Order Status
Product	FAS2240A-SSA-R5
OS	8.1.1 7-MODE | FW
Warranty Product	WARR36MTH_HW_DISK
Warranty Start	Mon Jan 07 2013
Warranty End	Sun Jan 31 2016
Failure Category	NetApp_Software/Firmware
Failure Type	Configuration
Component	Data ONTAP WAFL
Component Details	Panic
Resolution	Duplicate Case
KB Solution	KB Not Needed
Date Opened	Sep 05 2015, 06:13 IST (~1 Day)

[PUBLIC] @ Sep 05 2015, 09:30 IST Stephen Smitas

---Closing---
Problem Resolution:
Notified customer of existing case 2005862896 and archived as a duplicate to that case.

Root Cause:
RPANIC

Customer or SAM Agreement to Closure:
Scott Donahue
978-442-4470
978 551-5753
scott.donahue@cobham.com

Customer-Support Engg. Communication

[PUBLIC] @ Sep 05 2015, 09:27 IST Stephen Smitas

From: Smitas, Stephen
Sent: Friday, September 04, 2015 11:57 PM
To: scott.donahue@cobham.com
Cc: NetApp Support
Subject: RE: NetApp Log # 2005864084: Assignment of Your NetApp Support Case
Importance: High

I am following up on case 2005864084. I noted that the RPANIC occurred earlier thus generating case 2005862896; which will be the active case for this RPANIC. Please let me know if you have any questions regarding these cases. For any immediate assistance please call us at 888-463-8277.

Stephen

[PUBLIC] @ Sep 05 2015, 08:32 IST Stephen Smitas

---Initial/Open---

Customer Problem Summary:
Assigned Technical Case (2005864084) via dispatch. Noted that case 2005862896 is open for the same issue. Will archive as a duplicate to this case.

Model Name: FAS2240-4
asup-serial-num: 700001254159
asup-hostname: nhc3-wht-fas002

asup-generated-on: Fri Sep 04 20:39:18 EST 2015
==========
00000028.000005a7 00000f31 Sat Sep 05 2015 01:35:09 +00:00 [mgr.boot.reason_abnormal:ALERT] System rebooted after a panic.
00000028.000005ad 00000f31 Sat Sep 05 2015 01:35:09 +00:00 [mgr.stack.module:notice] Module kernel loaded from 0x100000 - 0xa163c8.
00000028.000005b0 00000f31 Sat Sep 05 2015 01:35:09 +00:00 [mgr.stack.module:notice] Module platform.ko loaded from 0xffffffff80a17000 - 0xffffffff80c17240.
00000028.000005b1 00000f31 Sat Sep 05 2015 01:35:09 +00:00 [mgr.stack.module:notice] Module common_kmod.ko loaded from 0xffffffff81c00000 - 0xffffffff81d13000.
00000028.000005b2 00000f31 Sat Sep 05 2015 01:35:09 +00:00 [mgr.stack.module:notice] Module maytag.ko loaded from 0xffffffff81d13000 - 0xffffffff89e5b000.
00000028.000005b3 00000f31 Sat Sep 05 2015 01:35:09 +00:00 [mgr.stack.module:notice] Module dbdriver.ko loaded from 0xffffffff89e5b000 - 0xffffffff89e5d000.

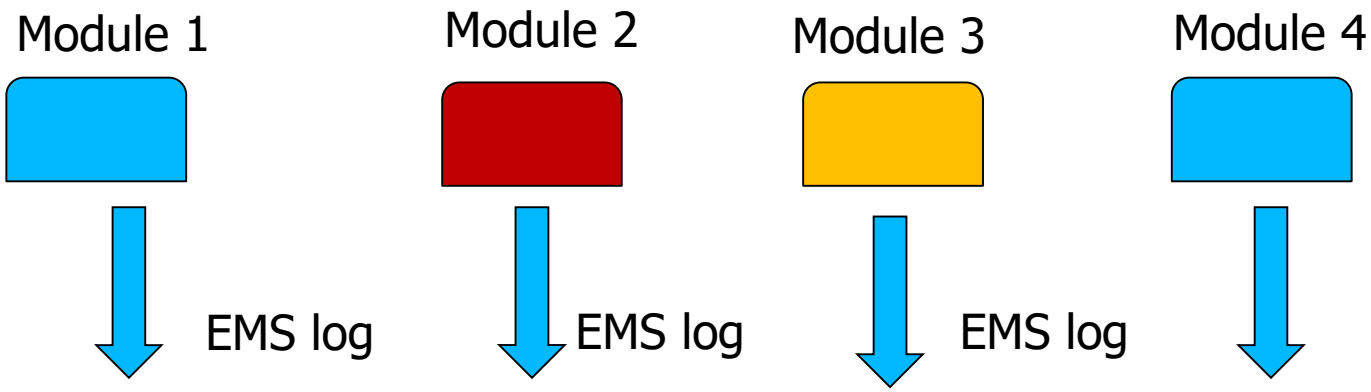
- Prerequisite
- Dataset
- Objective
- Challenges
- Model Development
  - Anomaly detection framework
  - Building an automated troubleshooter
- Results

Dataset

- Daily Event message system (EMS) log
- Customer support database
  - Customer support portal provides the platform to report cases, failures, communicate with support engineers
- Bug database
  - Internally oriented
  - Each **case** is associated with a **bug**

Dataset

- Daily Event message system (EMS) log



Dataset: A Typical EMS Log

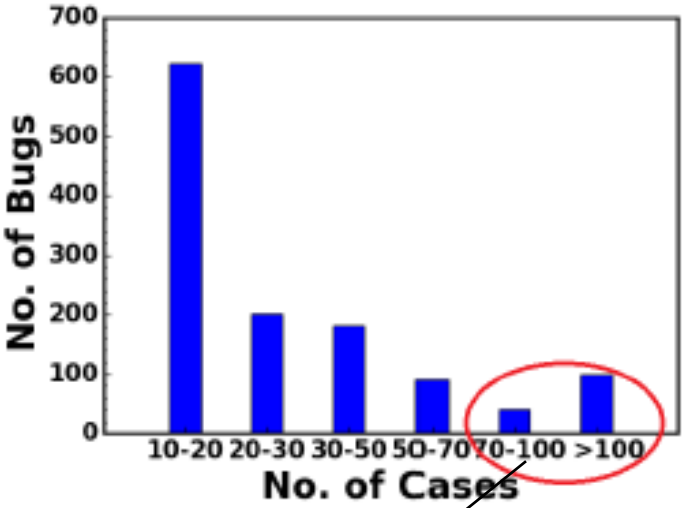
Raw EMS Data

```
<LR d="01Apr2014 00:00:00" id=[redacted] n=[redacted] o="statd" p="6" s="Ok" seq="71443" t="1396303200" type="0" vf="">
<kern_uptime_filer_1 cifsOps="5763" fcpOps="62872004381" httpOps="0" iscsiOps="0" msg=" 12:00am up 236 days,  5:51 0 NFS ops,
 5763 CIFS ops, 0 HTTP ops, 62872004381 FCP ops, 0 iSCSI ops " nfsOps="0" secs="20411519" />
</LR>
<LR d="01Apr2014 00:00:05" id=[redacted] n=[redacted] o="statd" p="6" s="Ok" seq="71444" t="1396303205" type="0" vf="">
<callhome_performance_data_1 subject="PERFORMANCE DATA" />
</LR>
...
..
..
```

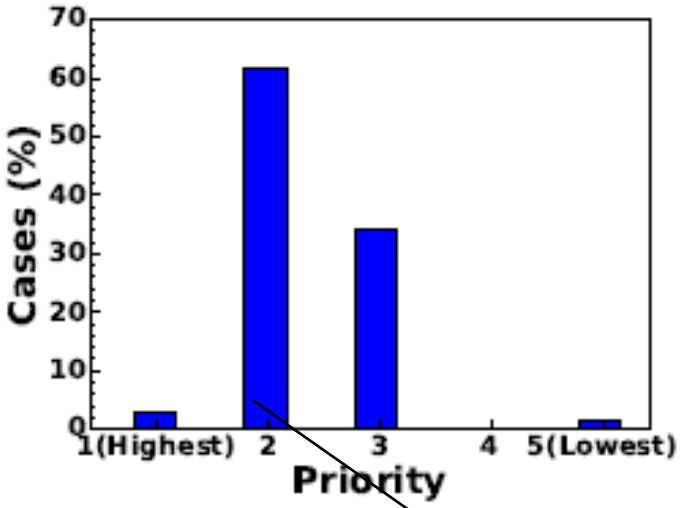
Extracted Information

Field	Log Entry Example	Description
Event Time	Apr 01 2014 09:11:12	Day, date, timestamp
System name	cc-nas1	Name of the node in cluster that generated the event
Event Message	kern.uptime.filer	Contains Subsystem name and event type
Severity	info	Severity of the event

Data filtering



(a)



(b)

1 Select the bugs with sufficient number of cases

2 Select the bugs with high priority levels

3 Eliminate the cases with missing data

Final EMS Dataset

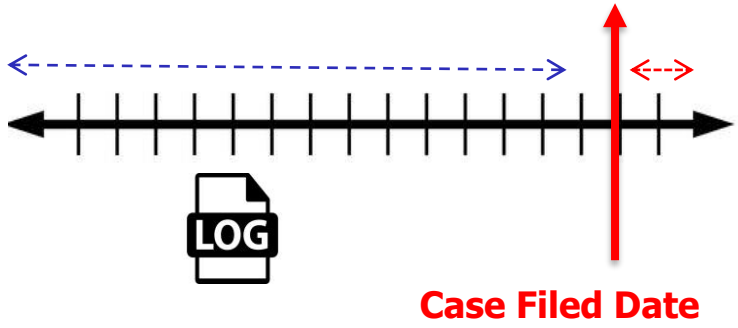
Raw EMS Data

```
<LR d="01Apr2014 00:00:00" id=[redacted] n=[redacted] o="statd" p="6" s="Ok" seq="71443" t="1396303200" type="0" vf="">
<kern_uptime_filer_1 cifsOps="5763" fcpOps="62872004381" httpOps="0" iscsiOps="0" msg=" 12:00am up 236 days,  5:51 0 NFS ops,
 5763 CIFS ops, 0 HTTP ops, 62872004381 FCP ops, 0 iSCSI ops " nfsOps="0" secs="20411519" />
</LR>
<LR d="01Apr2014 00:00:05" id=[redacted] n=[redacted] o="statd" p="6" s="Ok" seq="71444" t="1396303205" type="0" vf="">
<callhome_performance_data_1 subject="PERFORMANCE DATA" />
</LR>
...
..
..
```

Extracted Information

Apr 01 09:11:12
INFO kern_uptime_filer_1
…

Dataset-info	Number
Total No of Bugs	48
Total No of Cases	**4827**
No of Customers	2691
No of unique system	4305
No of Module	**331**
Types of Message	~8k
Timeline	January 2011 to June 2016



For each filed case we have collected around **18 weeks** prior data , and **1 weeks** log after case filed date.

How to resolve?

The support engineers use predefined rules to resolve the problem.

Resolution period:

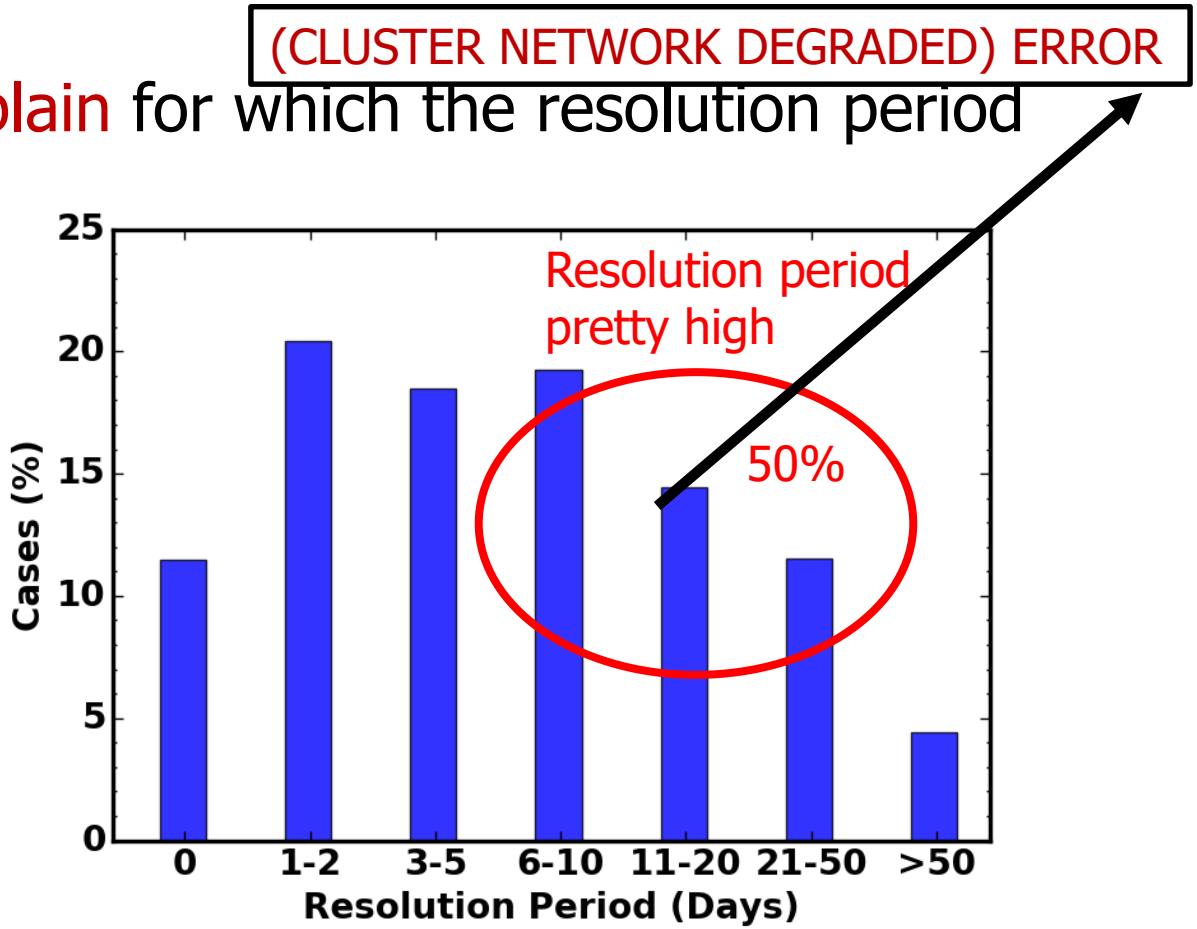
Let's assume customer filed case at To. It resolved on Tc

Resolution period = (Tc - To)

Motivation

Reliable and fast customer support service is pre-requisite to the storage industry

There are some complain for which the resolution period is very high.

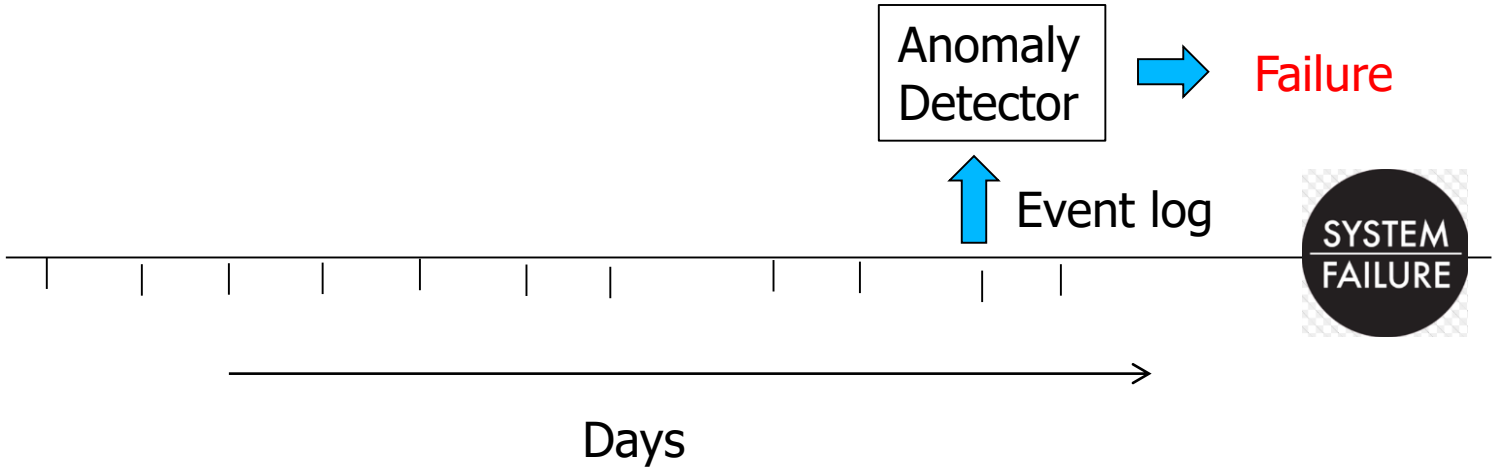


- Prerequisite
- Dataset
- **Objective**
- Challenges
- Model Development
  - Anomaly detection framework
  - Building an automated troubleshooter
- Results

Objective 1 (Anomaly detection)

- Leverage on the event logs generated by the subsystems/modules

- Development of anomaly detection framework



Objective 2 (Troubleshooting)

- Building a troubleshooter which can localize faulty components within a very short time.
- Providing a ranked list of modules to the support engineers
- Reducing the complexity of the diagnostic process

- Prerequisite
- Dataset
- Objective
- Challenges
- Model Development
  - Anomaly detection framework
  - Building an automated troubleshooter
- Results

Challenges (Anomaly detection)

- Detection of abnormality from log becomes challenging in the noisy environment
  - where the log gets colluded with the messages from system misconfiguration
- Do event log messages carry signals of anomaly?

- Do the anomaly signals eventually lead to failure?
  - File-system fragmentation may cause performance slowdown

- How many false alerts?

Challenges (Troubleshooting)

- Most of the real systems are complex as various constituent system components exhibit functional dependencies
- Each component has its own failure modes. For example, a storage system failure can be caused by disks, physical interconnects, shelves, RAID controllers etc.
- It is extremely hard for support engineer to have a updated domain knowledge in this evolving system.
- In such a large evolving complex system the prior knowledge of dependency tree between modules is not available.

- Prerequisite
- Dataset
- Objective
- Challenges
- Model Development
  - **Anomaly detection framework**
  - Building an automated troubleshooter
- Results

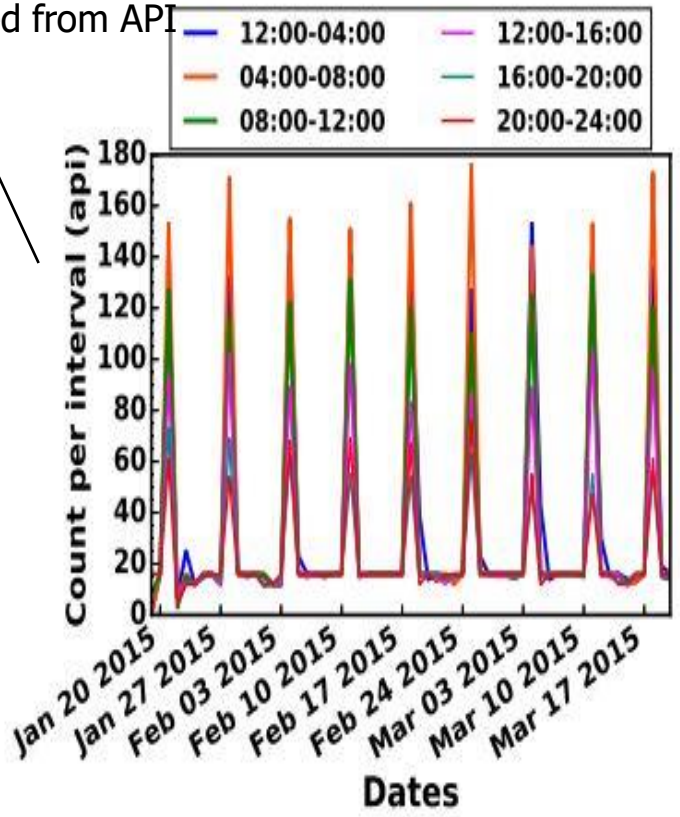
Model development: Attribute Extraction

Attributes	Description
Event Count	Total number of events generated by the subsystem
Event Ratio	Ratio of number of events generated by the subsystem to total number of messages
Mean Inter-arrival Time	Mean time between successive events generated of the particular subsystem
Mean Inter-arrival Distance	Mean number of other messages between successive events of the particular subsystem
Severity Spread	**Eight** features corresponding to event counts of each severity type for the subsystem
Time-interval Spread	**Six** features denoting event counts during six four-hour intervals of the day for the subsystem

Observation1:Periodicity

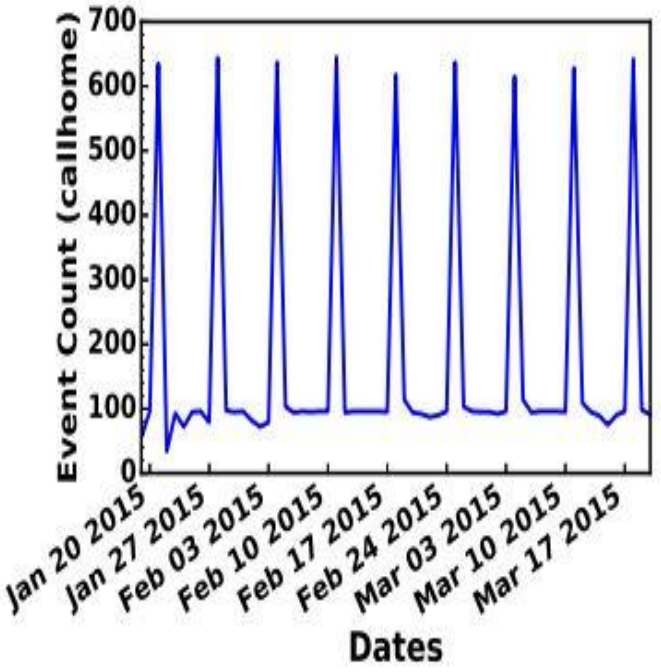
Weekly periodicity can be observed for attributes from event log

Number of messages generated from API module



(a)

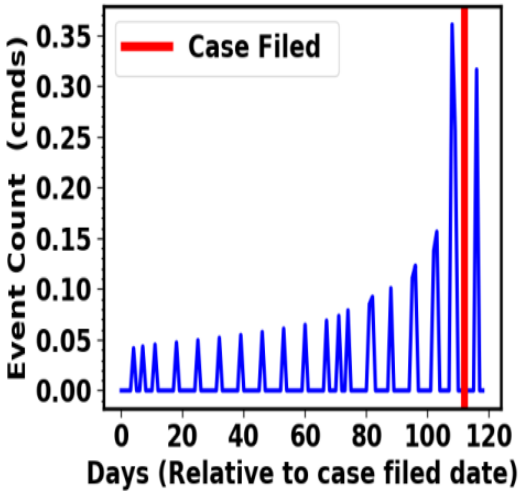
planned maintenance, scheduled backups, workload intensity changes

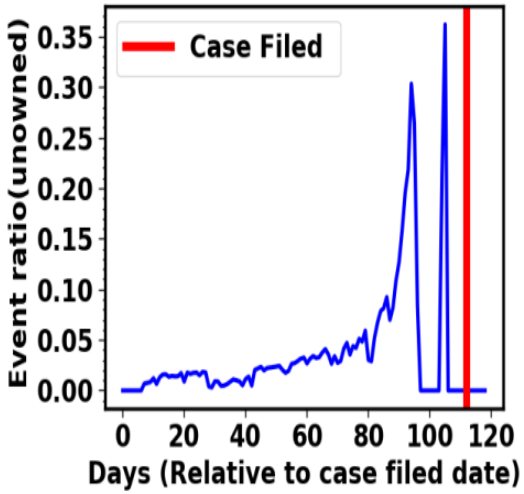


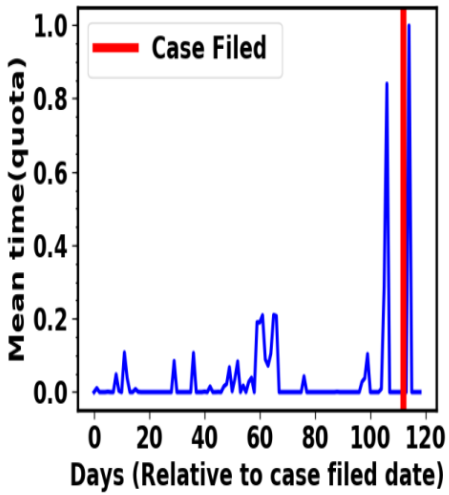
(b)

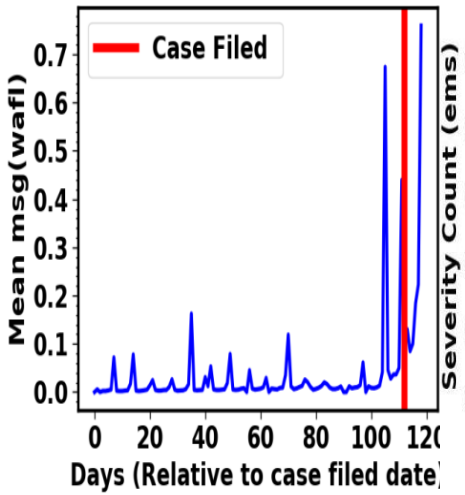
Anomaly Clues

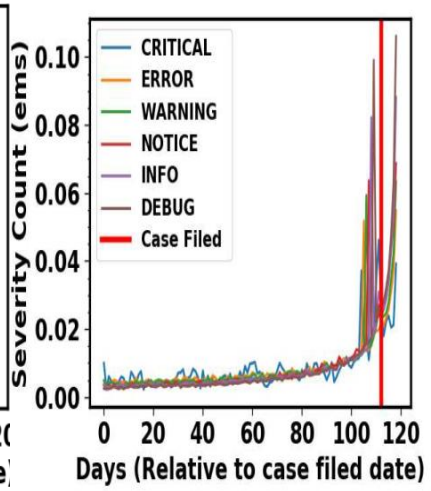
- If one or more subsystem is going through an anomalous phase
  - it gets reflected in some attributes of logs generated for those subsystems

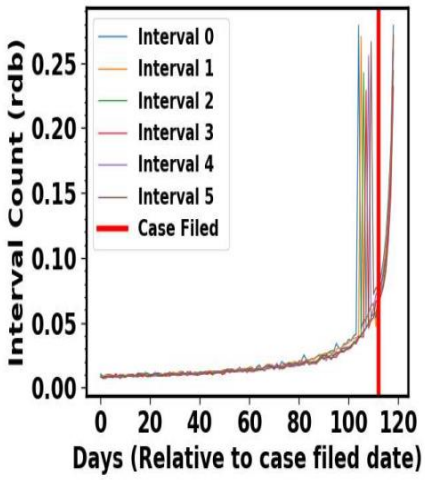




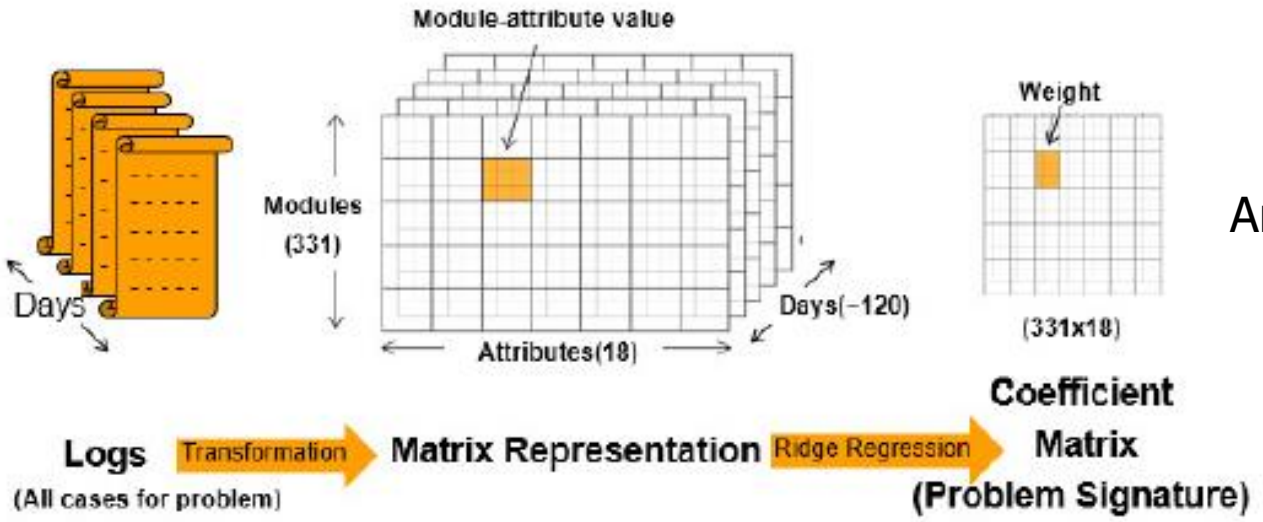








Model development: Overview



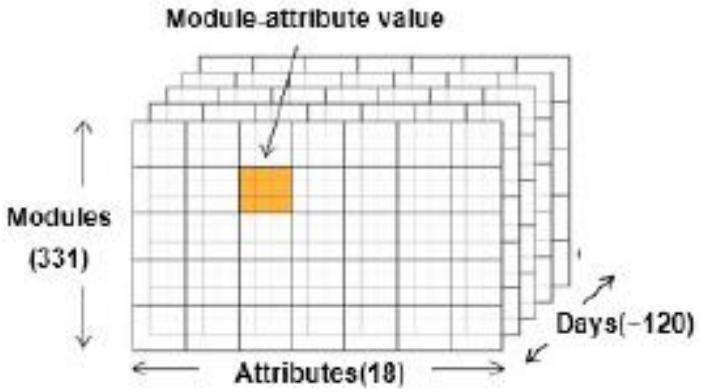
Log transformation

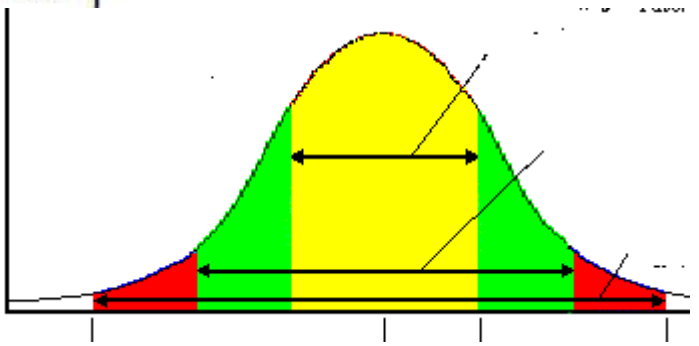
Extract 18 features from EMS log, for each module

Model development : Log Transformation

- EMS log of each day is abstracted into a matrix (X_d)

$$X_d = \left\{ X_{i,j}^{(d)} \text{ where } i \in M \text{ and } j \in A \right\}$$





Anomaly score of this observation $X_{i,j}$ is then calculated as

$$S_{i,j} = 2 * \left| 0.5 - CDF\left(X_{i,j}\right) \right| \qquad (2)$$

- We fit a normal distribution with the features of the last few weeks

$$(X_{i,j}^{(d-7)}, X_{i,j}^{(d-14)}, X_{i,j}^{(d-21)}, X_{i,j}^{(d-28)}, \ldots).$$

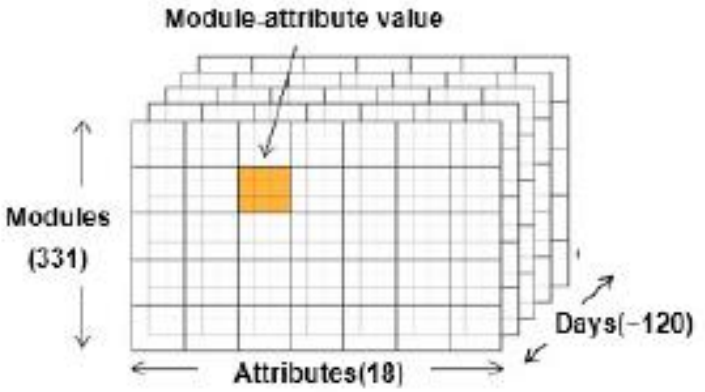
Model development: Score Matrix

- EMS log of each day is abstracted into a matrix (X_d)

$$X_d = \left\{ X_{i,j}^{(d)} \text{ where } i \in M \text{ and } j \in A \right\}$$

- We transform the raw matrix (X_d) of d^{th} day into score matrix (S_t) as follows

$$S_t = \left\{ \begin{array}{l} S_{i,j}^{(t)} \text{ where } i \in X \text{ and } j \in Y \\ \text{X is the set of subsystems and} \\ \text{Y is the set of attributes} \end{array} \right\}$$

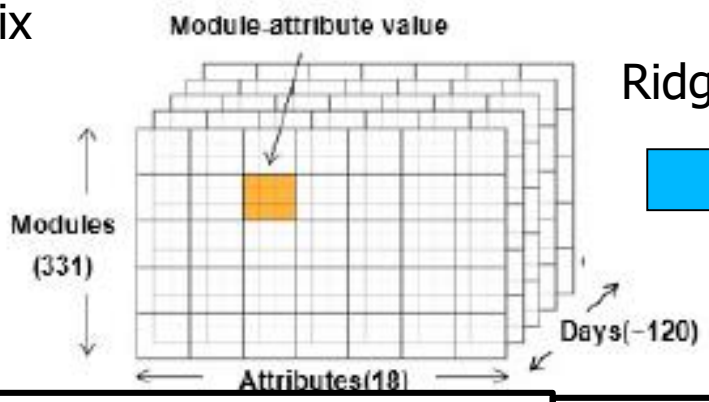


Model development: Anomaly Detect

S(i,j) contributes differently to overall anomaly of the system

Score matrix

S_f



Ridge regression

W Weight matrix

Event log of a day

Anomaly score

Above threshold

Anomaly

Below threshold

No Anomaly

$$J(w) = \lambda \left\| w^2 \right\| + \sum (w^T x_i - y_i)^2.$$

$$\frac{\partial J}{\partial w} = \lambda w + \sum_i (w^T x_i - y_i) x = 0$$

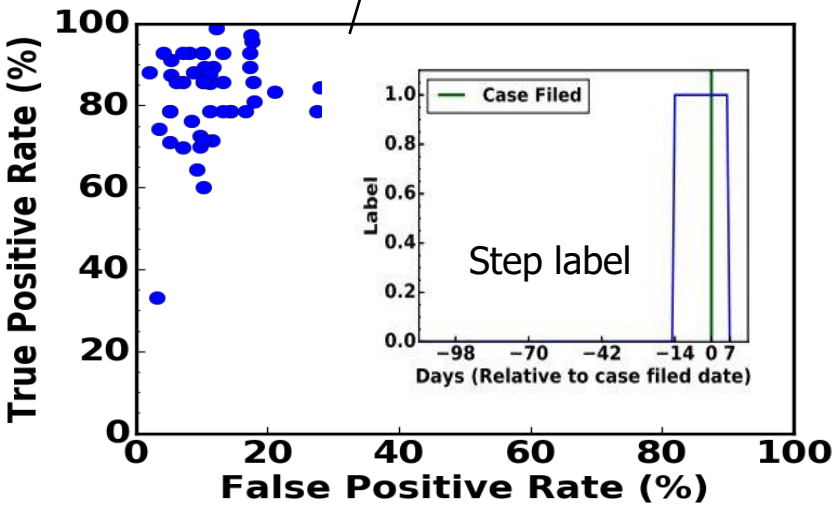
$$(X_d^T + \lambda I) w = Sy$$

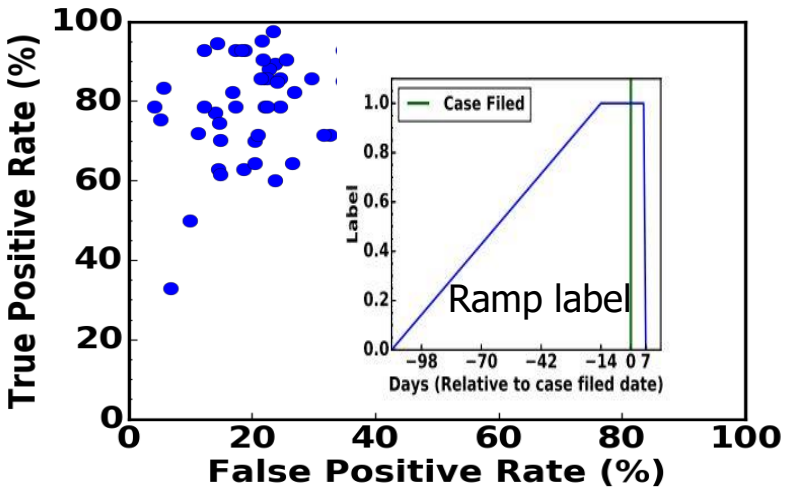
$$w = (X_d X_d^T + \lambda I)^{-1} X_d y$$

$$= X_d \alpha.$$

True positive Vs False positive

High anomaly detection rate with low false alert





Model	TPR (%)	FPR (%)	Accuracy (%)	F1 Score (%)
Liang	74	34.75	66.79	0.440
EGADS-KSigma	***76.41***	22.49	***77.31***	***0.543***
EGADS-DBScan	59.52	***20.05***	76.34	0.476
EGADS-ExtremeLowDensity	62.31	21.78	75.41	0.475
ADELE Direct	47.87	27.04	68.53	0.352
ADELE	**83.92**	**12.02**	**87.26**	**0.710**

Comparison with Baseline

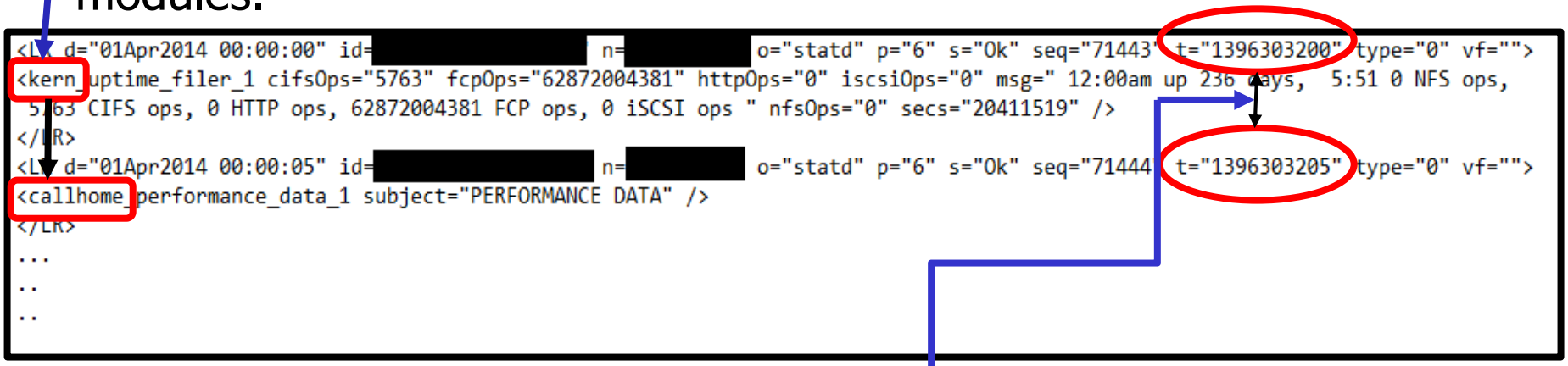
ADELE: Anomaly Detection from Event log Empiricism, accepted in **INFOCOM'18**

- Prerequisite
- Dataset
- Objective
- Challenges
- Model Development
  - Anomaly detection framework
  - **Building an automated troubleshooter**
- Results

Graph Construction

Vertex:

Each module is considered as vertex, we took all **331** possible modules.



Edge:

Edge is decided based on **timestamp difference**, if the timestamp difference between two module is less than **300 second**, one directed edge is formed between them.

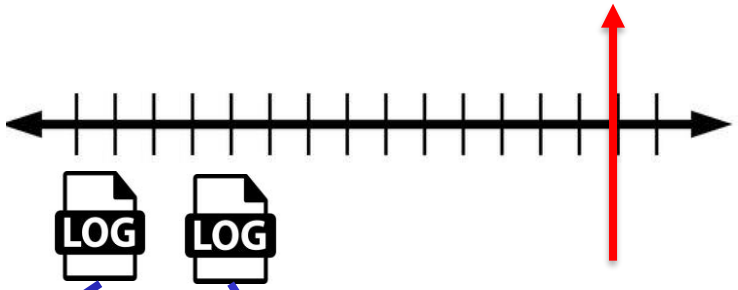
Edge weight:

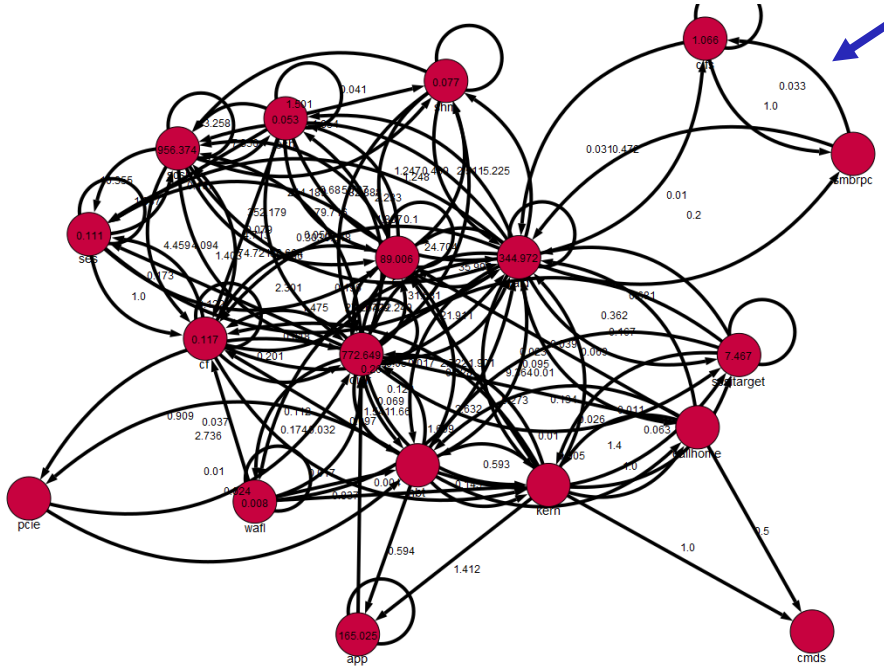
$$k * \sum_{i=1}^{k} 1/t_i$$

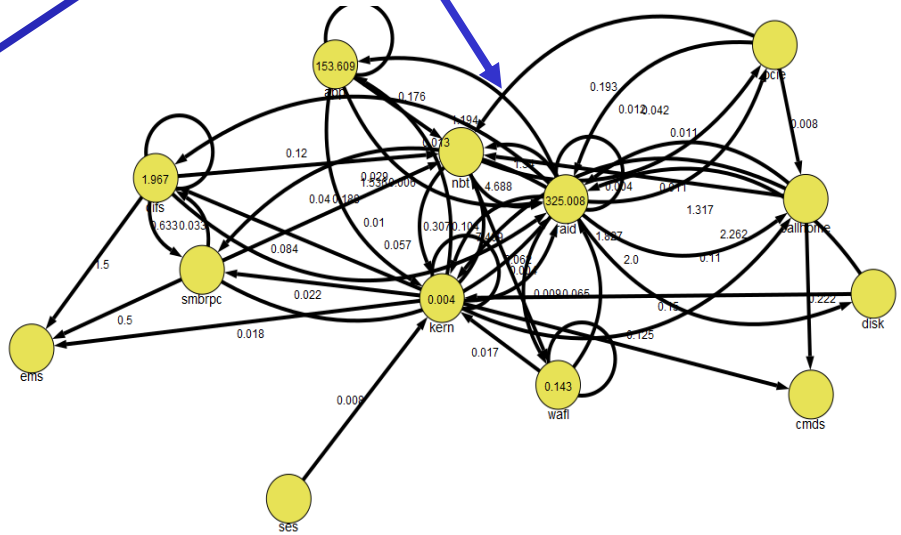
Edge weight is as follows, where κ is no of occurrences of edges and t_i is timestamp difference.

Sample Example

Corresponding to each case, we collect **18 weeks** of data - we construct a graph corresponding to each **week** -consequently, we get **18 graphs** from a single case. The **last two** graphs we assume is arising out of **anomalous state** of the system.







Graph Encoding

Vertex encoding (vbits):

- $\log_2 v$ bits to encode the number of vertices v in the graph
- $v * \log_2 u$ *bits to encode labels of all* v *vertices* where u is total unique no of labels of vertices.

$$\boldsymbol{vbits = \log_2 v + v * \log_2 u}$$

Edge encoding (ebits):

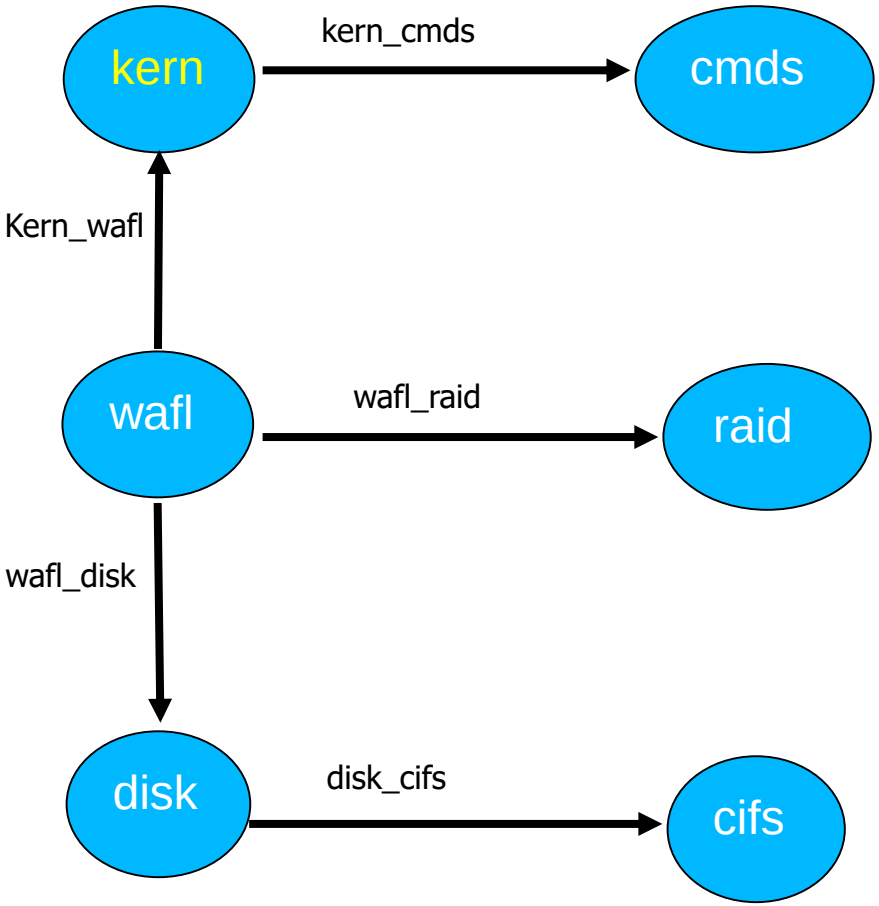
$$\textbf{ebits} = \boldsymbol{e * (1 + \log_2 u) + K * \log_2 m + \log_2 m}$$

e is total no. of edges, K is total no. of 1's in the adjacency matrix, m=max e(i,j)

Row encoding (rbits):

$$\boldsymbol{rbits = \quad v * \log_2 b \sum_{i=1}^{v} \log_2 \binom{v}{k_i}}$$

Encoding example



$$\begin{array}{l} \text{kern} \\ \text{cmds} \\ \text{wafl} \\ \text{raid} \\ \text{disk} \\ \text{cifs} \end{array} \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{vbits} = \log_2 6 + 6 * \log_2 11 = 23.33\ bits$$

$$\mathbf{ebits} = \boldsymbol{e} * (\mathbf{1} + \mathbf{log_2}\ \boldsymbol{u}) + \boldsymbol{K} * \mathbf{log_2}\ \boldsymbol{m}$$
$$= 5*(1+\log_2 11)+5*\log_2 1 = 22.25\ bits$$

$$\boldsymbol{rbits} = 21.49\ bits$$

No. of vertices: **6**
Unique labels: **11**
e=5; K=5; m=1

Total bits=67.07 bits

Step 1: Finding Abnormal Substructure (PCCS)

Subgraph:

A *substructure* is a connected subgraph of the overall graph.

Best Substructure:

we consider the best substructure to be one that minimizes the following value:

$$F1(S, G) = DL(G|S) + DL(S)$$

Where G is the entire graph, S is the substructure, DL(G|S) is the description length of G after compressing it using S, and DL(S) is the description length of the substructure

Intuition:

Anomalous substructure occurs very infrequently.

Abnormal Substructure finding steps

- First, we compute anomaly score by the transformation cost (using insertion and deletion of vertex and edges) to match the entity with the **best substructure**.

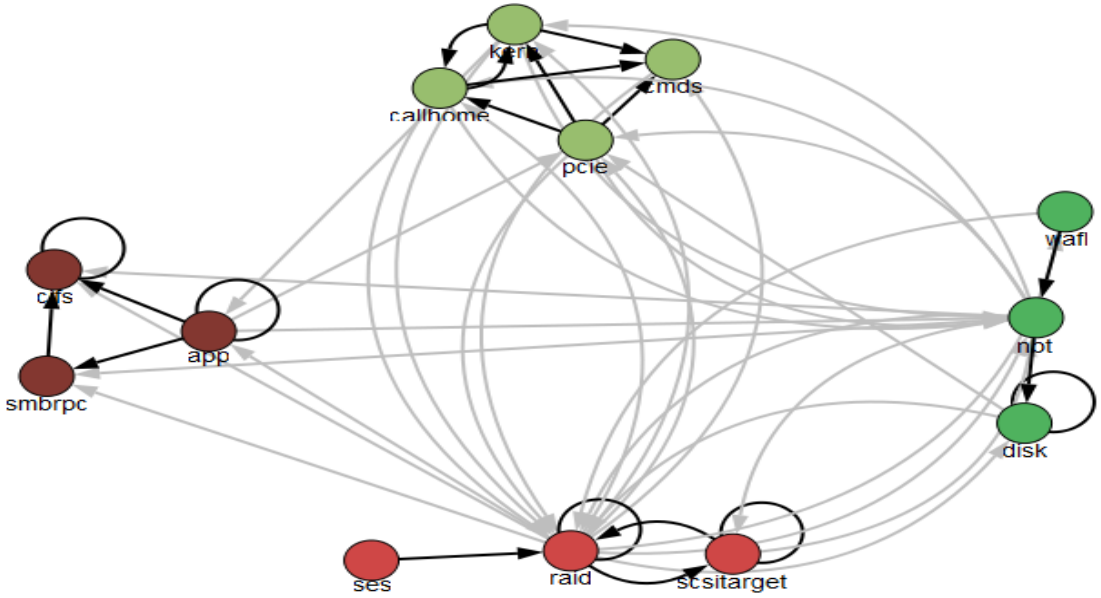
- We finally shortlist only those abnormal substructure where anomaly score exceeds a certain threshold (**0.95**).

- Hence the problem creating candidate set (PCCS) is the union of the modules present in the shortlisted anomalous structure

Step2: Community Detection

Intuition: If there is failure in one module of a community, other modules present in the group might be affected due to dependency between modules

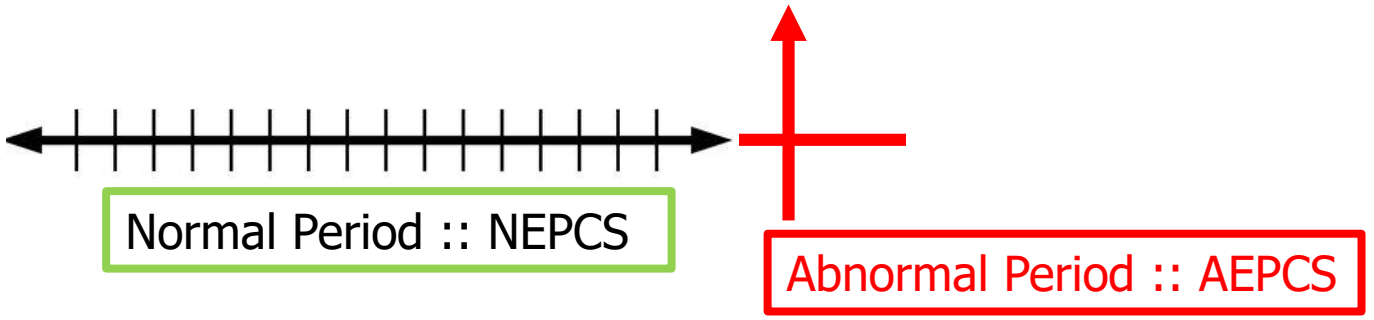
- We choose Louvain community detection algorithm



Step 3: Set Expansion

- We calculate normalized overlapping index between PCCS and each community

- If overlapping index exceeds some threshold (0.75) for a particular cluster, we expand PCCS by incorporating modules of that specific cluster

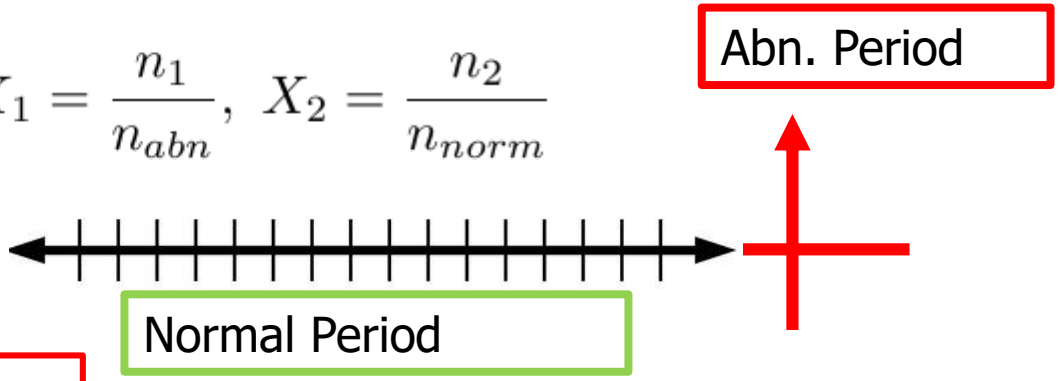


Final PCS Construction

- For a case, suppose we discover that module appears n_1 times in abnormal set AEPCS out of total n_{abn} samples and it also appears in NEPCS n_2 times out of total n_{norm} normal samples.

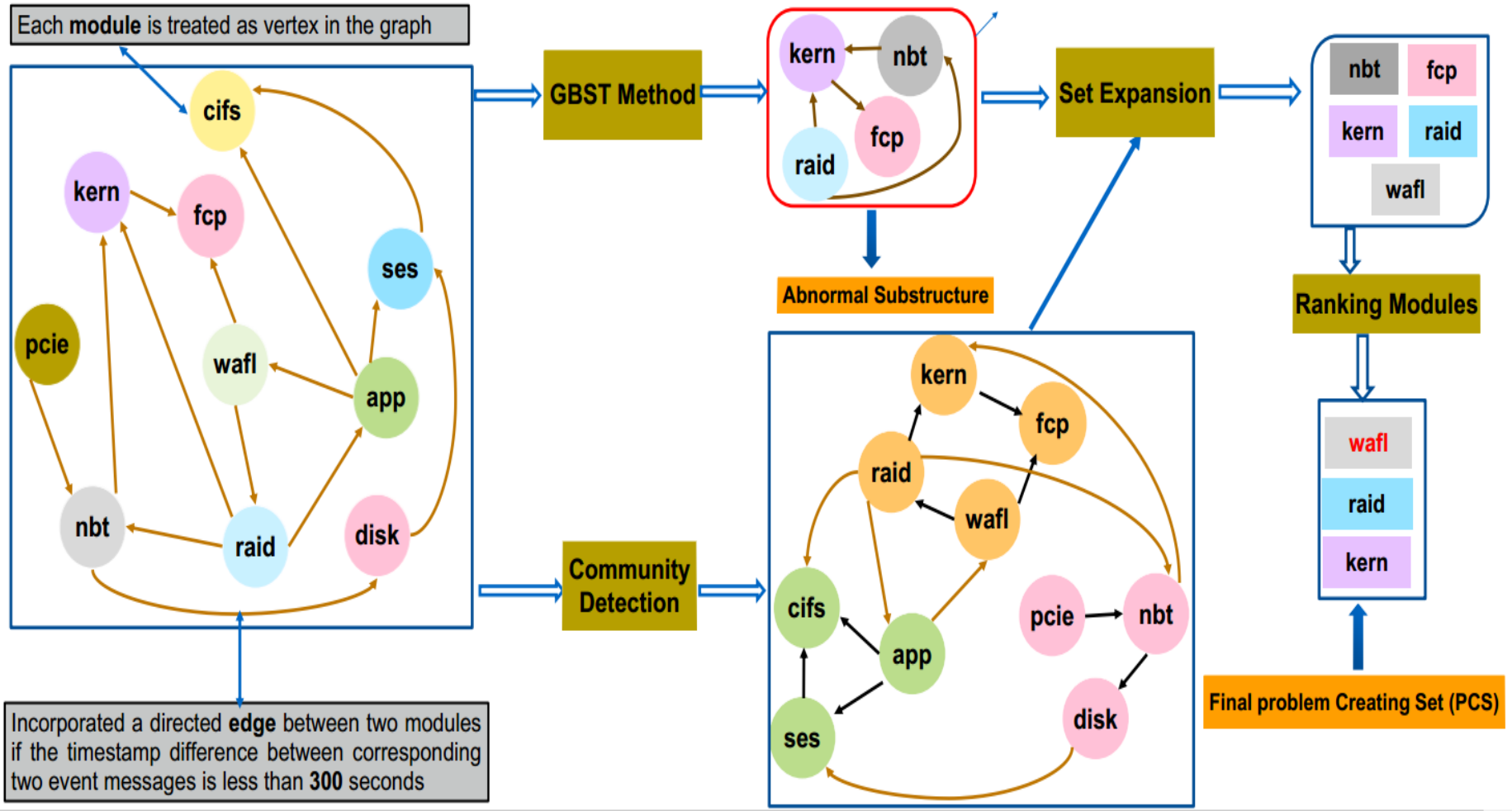
Then causality score (CS) of the module is as follows

$$CS = \frac{X_1}{(X_1 + X_2)};\ X_1 = \frac{n_1}{n_{abn}},\ X_2 = \frac{n_2}{n_{norm}}$$



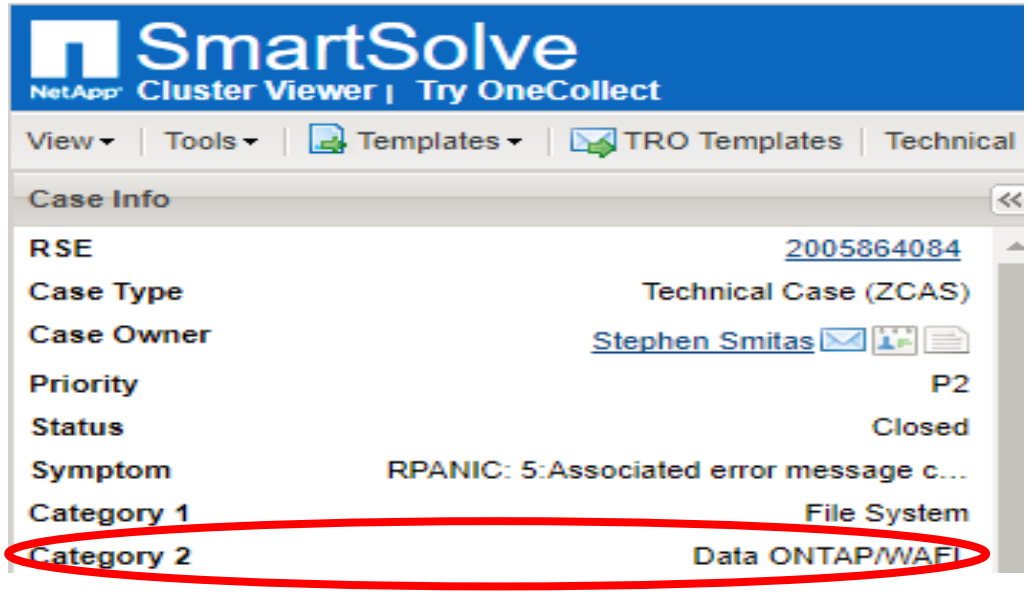
Top ranked modules considred as final problem creating set (PCS)

An Example


Each module is treated as vertex in the graph
cifs
kern
fcp
ses
pcie
wafl
app
nbt
raid
disk
Incorporated a directed edge between two modules if the timestamp difference between corresponding two event messages is less than 300 seconds
GBST Method
Community Detection
kern
nbt
fcp
raid
Abnormal Substructure
Set Expansion
nbt
fcp
kern
raid
wafl
Ranking Modules
wafl
raid
kern
Final problem Creating Set (PCS)

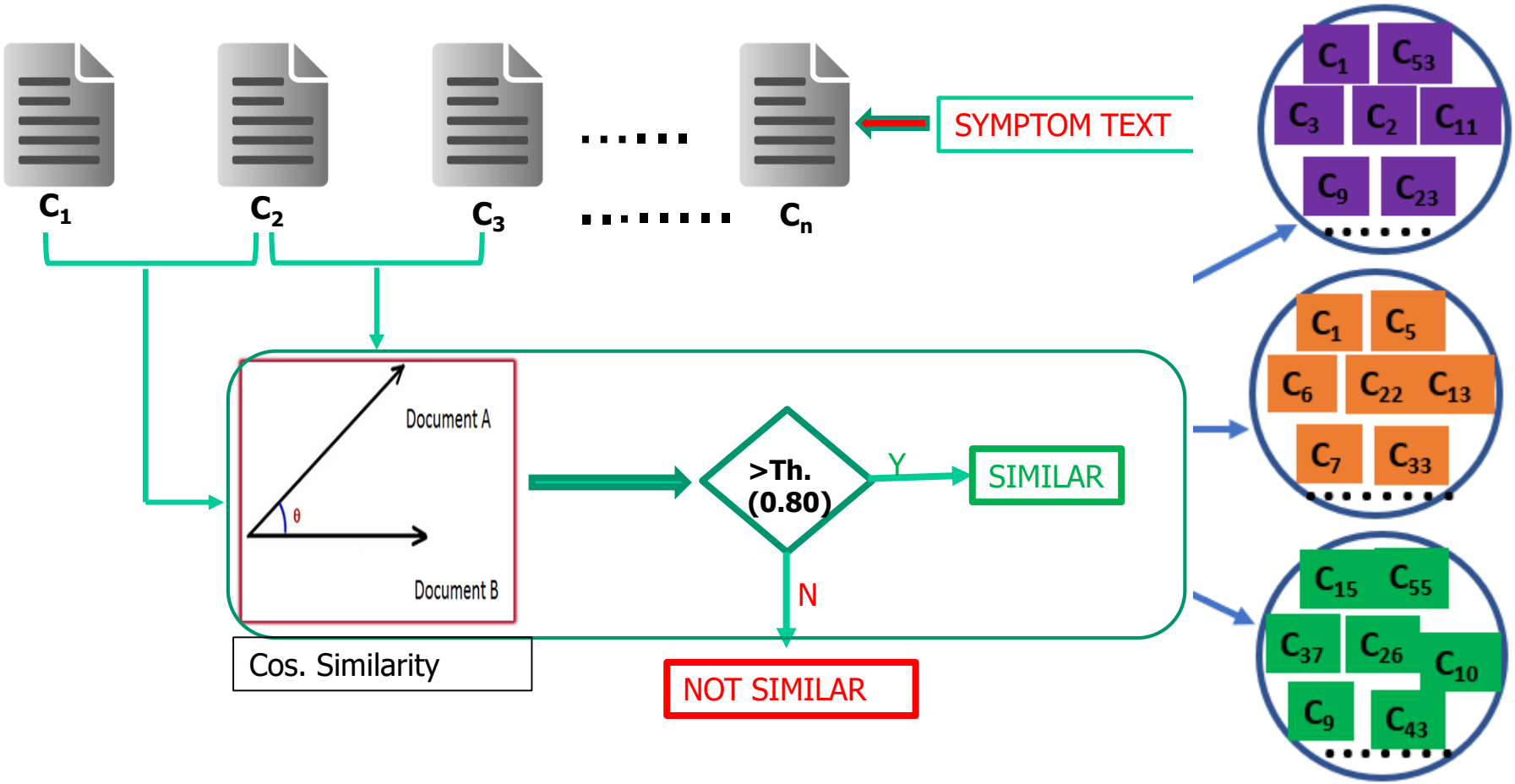
Validation

- ***Direct*** (**Ground Truth** available)
  - Support engineers extracted the trouble creating modules from domain knowledge and conversation with customer for only **20.50%** of cases, where evaluation becomes straightforward



- ***Indirect***
  - **Similar cases** will have approximately similar problem creating modules set.

Grouping Similar Cases (Sym-Text Based)



Similar Cases (EMS-Log Based)

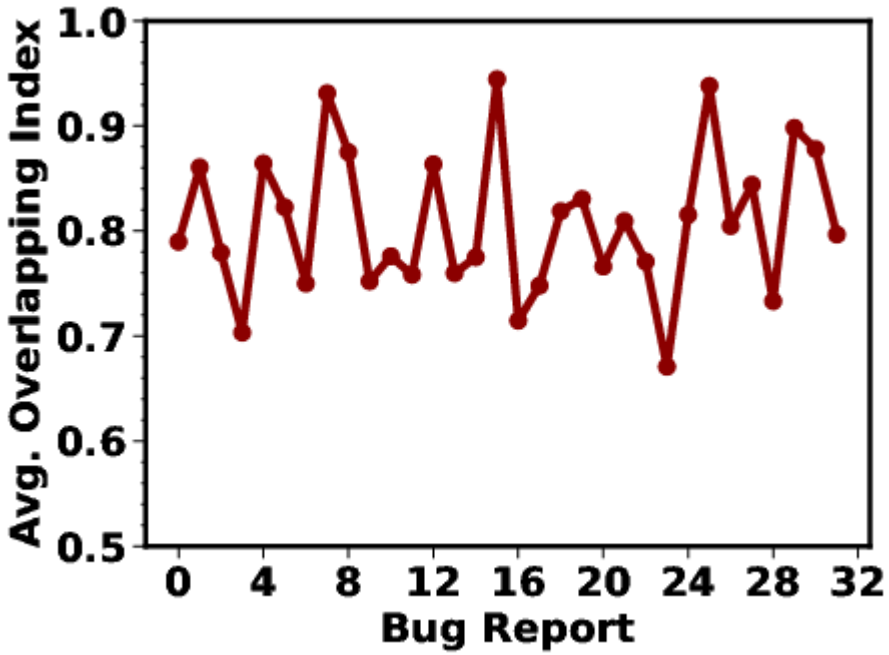
The similar cases belongs to both the group taken as final similar case set

- Prerequisite
- Dataset
- Objective
- Challenges
- Model Development
  - Anomaly detection framework
  - Building an automated troubleshooter
- Results

Overlapping Score (Indirect Validation)

Mathematically, for two arbitrary sets S1 and S2

Overlapping score (S_1, S_2)= $|S_1 \cap S_2| \,/\, |S_1 \cup S_2|$



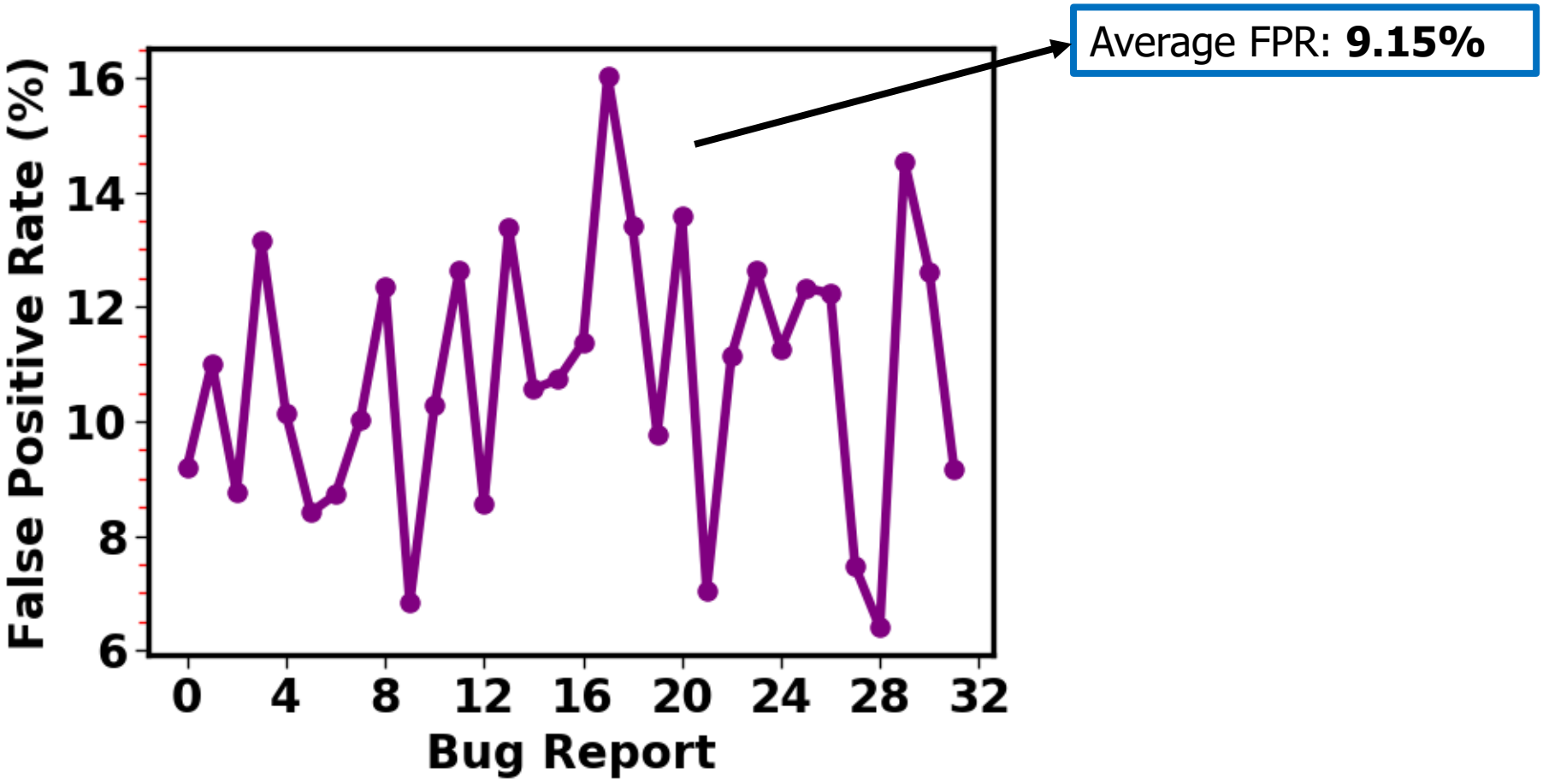
Average Overlap: **0.807**

The PCS of similar cases are ~ 80% similar

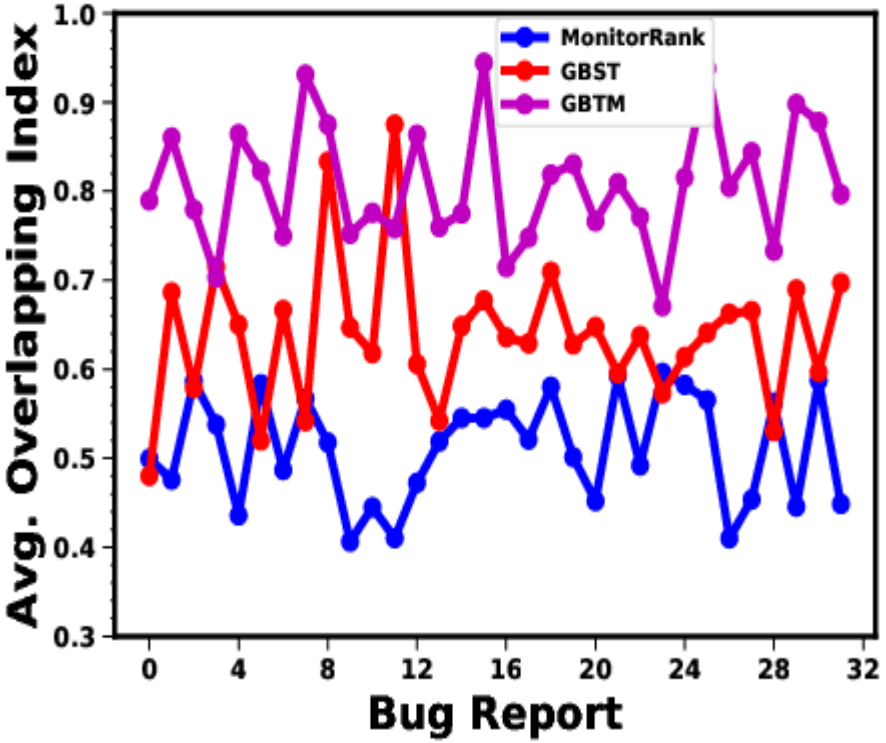
Indirect validation

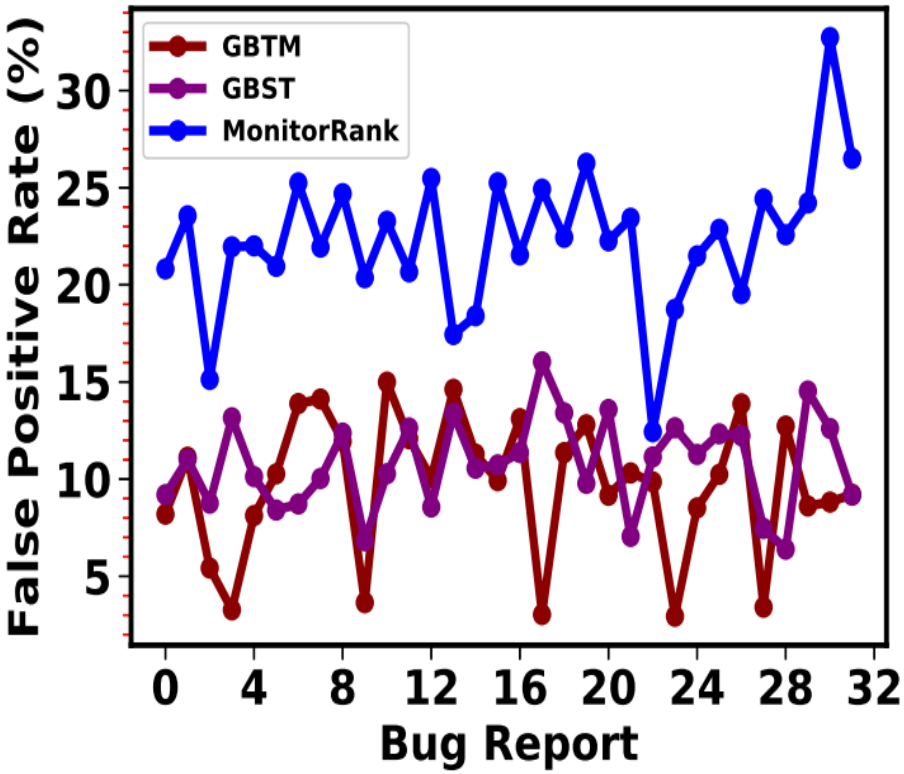
False Positive Rate

Intuitively, the problem causing modules should appear only in the abnormal state. If a module appears in both NEPCS and AEPCS set we treat that module as a false positive.



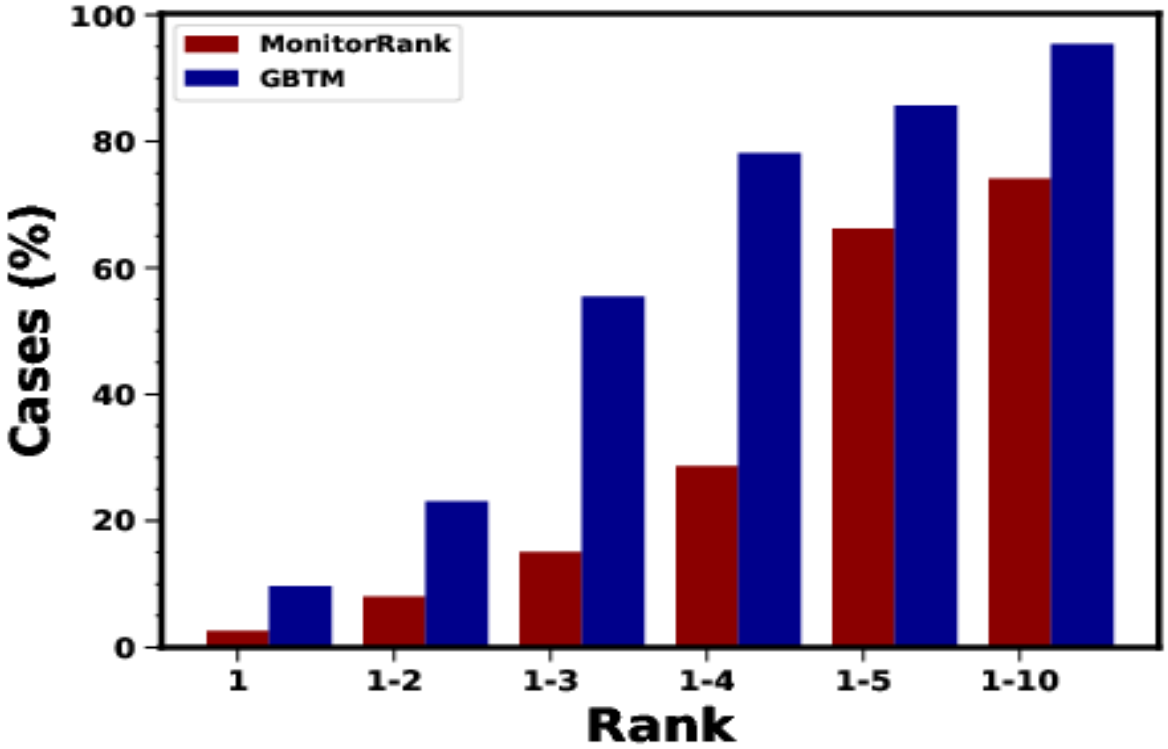
Comparison with Baseline





Model	FPR (%)	APCS Overlap (%)
MonitorRank	23	55.3
GBST	**9.15**	63.8
GBTM	9.71	**80.7**

Ranking Modules



We provide a ranked list of modules to the support engineers which can significantly narrow down the troubleshooting process for around **95% cases**

Conclusion

- Logs are challenging to analyze manually because they are noisy
- In large scale system, constituent system components exhibit functional dependencies.
- We proposed **ADELE**, a machine learning model to detect anomalies with high anomaly detection rate and low false alert.
- We proposed **GBTM**, troubleshooting tool which abstracts the raw log by a graph structure and infers a probable set of malfunctioning modules with the help of community structure.

Thank you!

Follow the work of Complex Network Research Group (**CNeRG**), IIT KGP at:
Web: http://www.cnergres.iitkgp.ac.in/
Facebook: https://web.facebook.com/iitkgpcnerg
Twitter: https://www.twitter.com/cnerg