Mining Causality Graph For Automatic Web-based Service Diagnosis

Xiaohui Nie† , Youjian Zhao† , Kaixin Sui† , Dan Pei†, Yu Chen‡ , Xianping Qu‡

清華大學 Tsinghua University †

Baidu 百度 ‡

IPCCC 2016

Web-based service

- **Web-based service is indispensable in our daily life.**
  - Search
  - E-commerce
  - Social
  - Video

Web-based service

- **The failures of web-based service cause great loss.**
  - **Web Search**

    Google latency increases 100ms ~400ms, query number decrease 0.2%~0.6%[1]

    bing latency increases 50ms, revenue decrease 1.2% [2]
  - **E-commerce:**

    amazon went down for 45 minutes, causing $5M loss [3]

    PayPal went down for 1 hour, causing $7.2M loss [4]

Quick and precise diagnosis for web-based service is crucial.

[1] J.Brutlag. (June, 2009). Speed matters for Google web search.

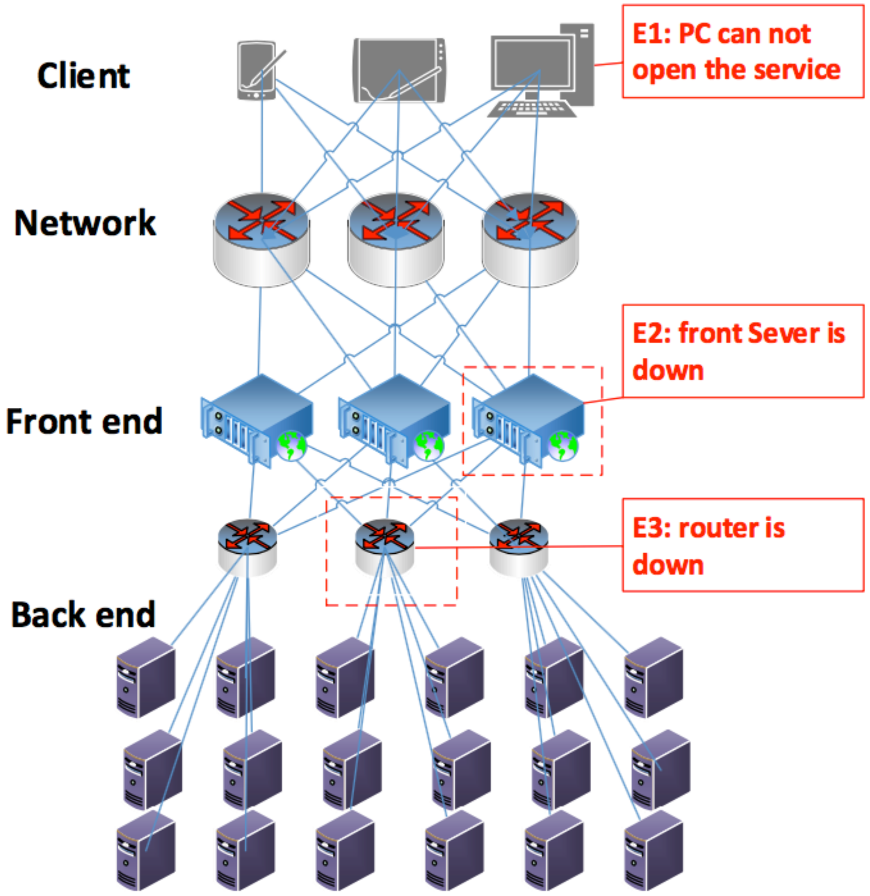
[2] E.Schurman,J.Brutlag.(June,2009).The User and Business Impact of Server Delays, Additional Bytes and Http Chunking in Web Search.

[3] S.K.Abudheen. (August, 2013). Amazon.com goes down for 45 minutes,loses $5M in business.

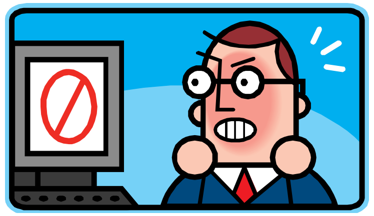
[4] S. Shankland. (August 3, 2009). PayPal suffers from e-commerce outage. Available: http://news.cnet.com/8301-1023_3-10302072-93.html

Diagnosing web-based service

- **Simple example of diagnosing web-based service.**



Users

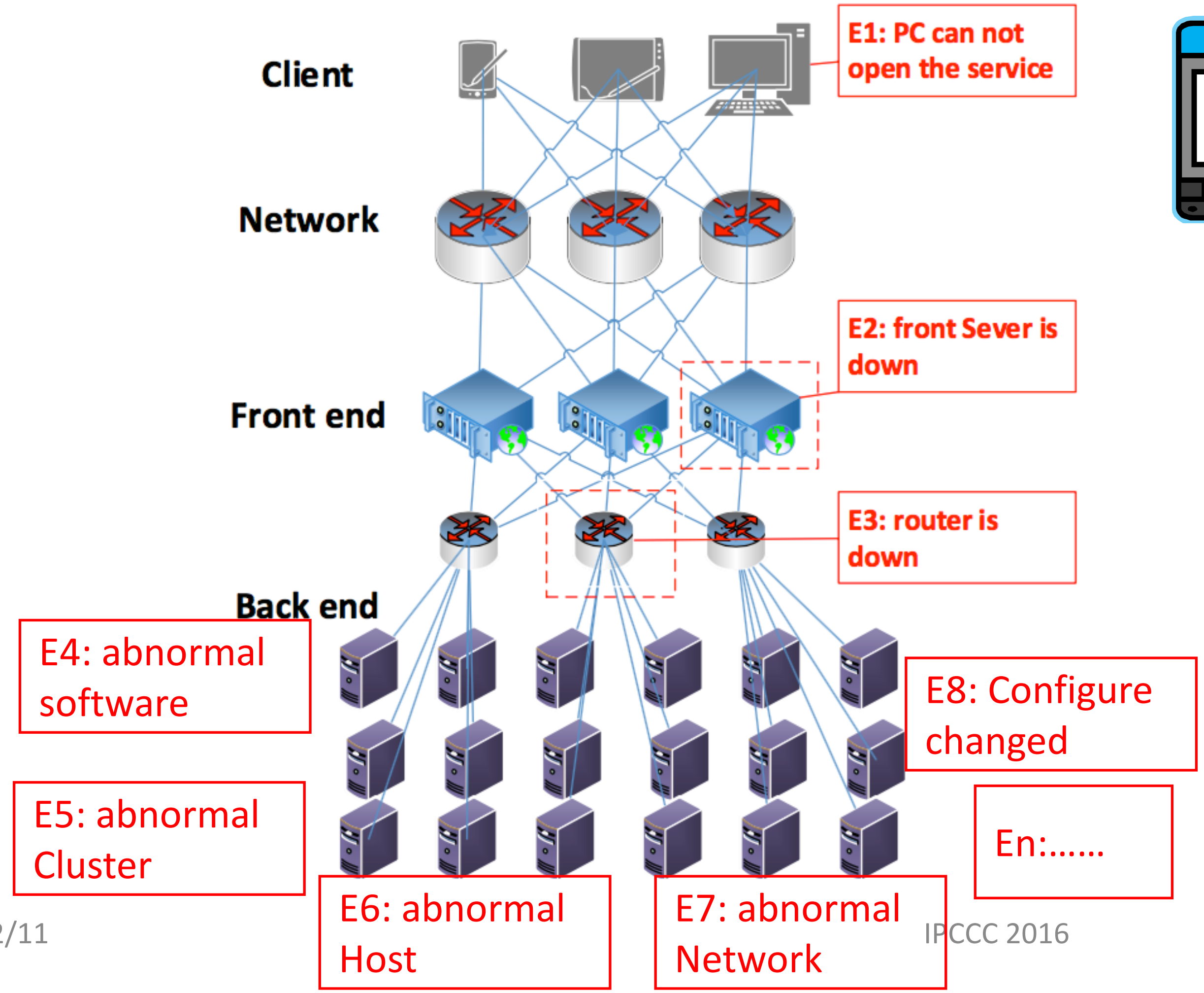


Failure: Anomalous KPI (Key performance Indicator)

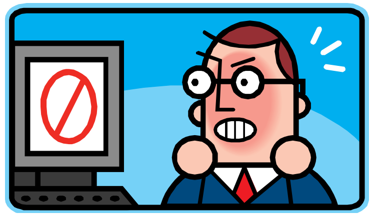
Operators

Diagnosing web-based service

- **Simple example of diagnosing web-based service.**

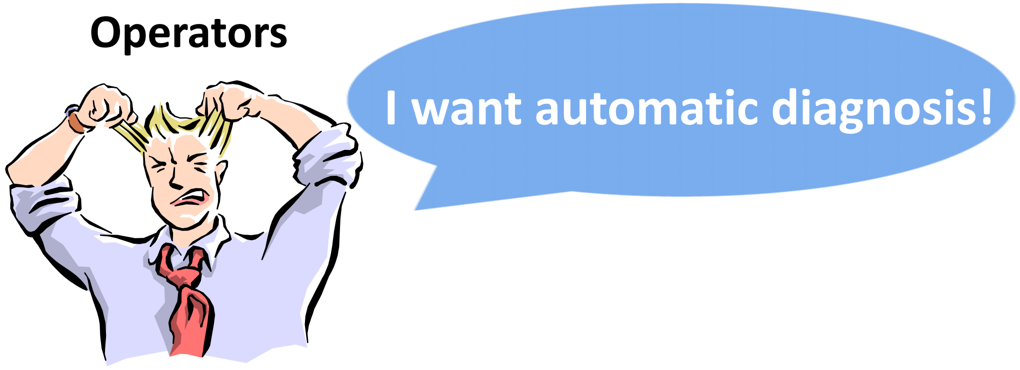


Users



Failure: Anomalous KPI (Key performance Indicator)

Operators

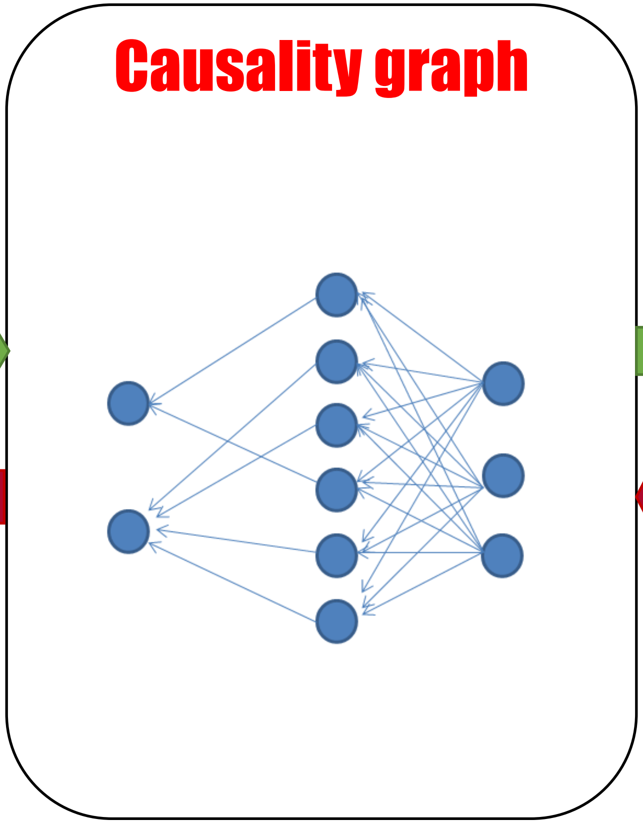


Challenges

- **Diagnosing web-based service is a thorny problem.**
  - **Challenges:**
    - **1. Large-scale infrastructure, complex software interaction.**
      - Hundreds or thousands of machines.
      - Many software components.
    - **2. Large-scale symptom events.**
      - 10~20 thousand symptom events are generated per week in a major service of Baidu.
      - Hard to find user-perceived root cause.
    - **3. Complex relationship between symptom events.**
      - No one can understand all the relationship .

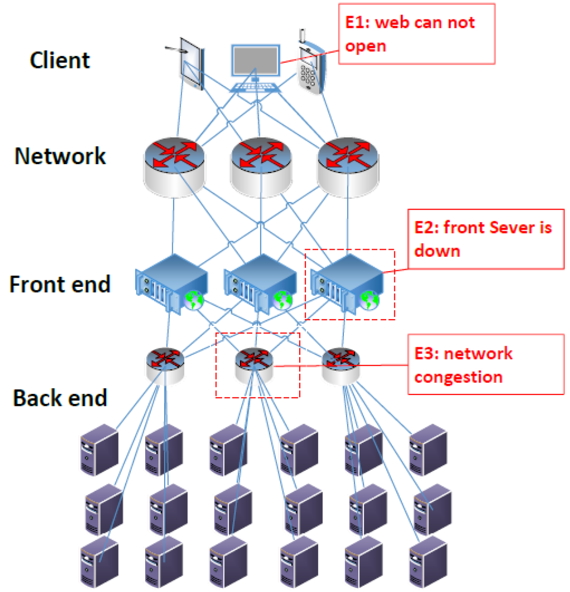
Key idea

Failure
browser saw error codes
An application was observing intermittently high response times to its server.
Database server refused to start.
The network latency between hosts was high.



Root cause
A software update had changed the Web server's configuration.
An unrelated process on the server's machine was intermittently consuming a lot of memory.
The server was misconfigured.
A buggy process was broadcasting UDP packets at a high rate.

Key idea



Web-based service

List Suspects

Feedback

Automatic diagnosis system

Service's Operator

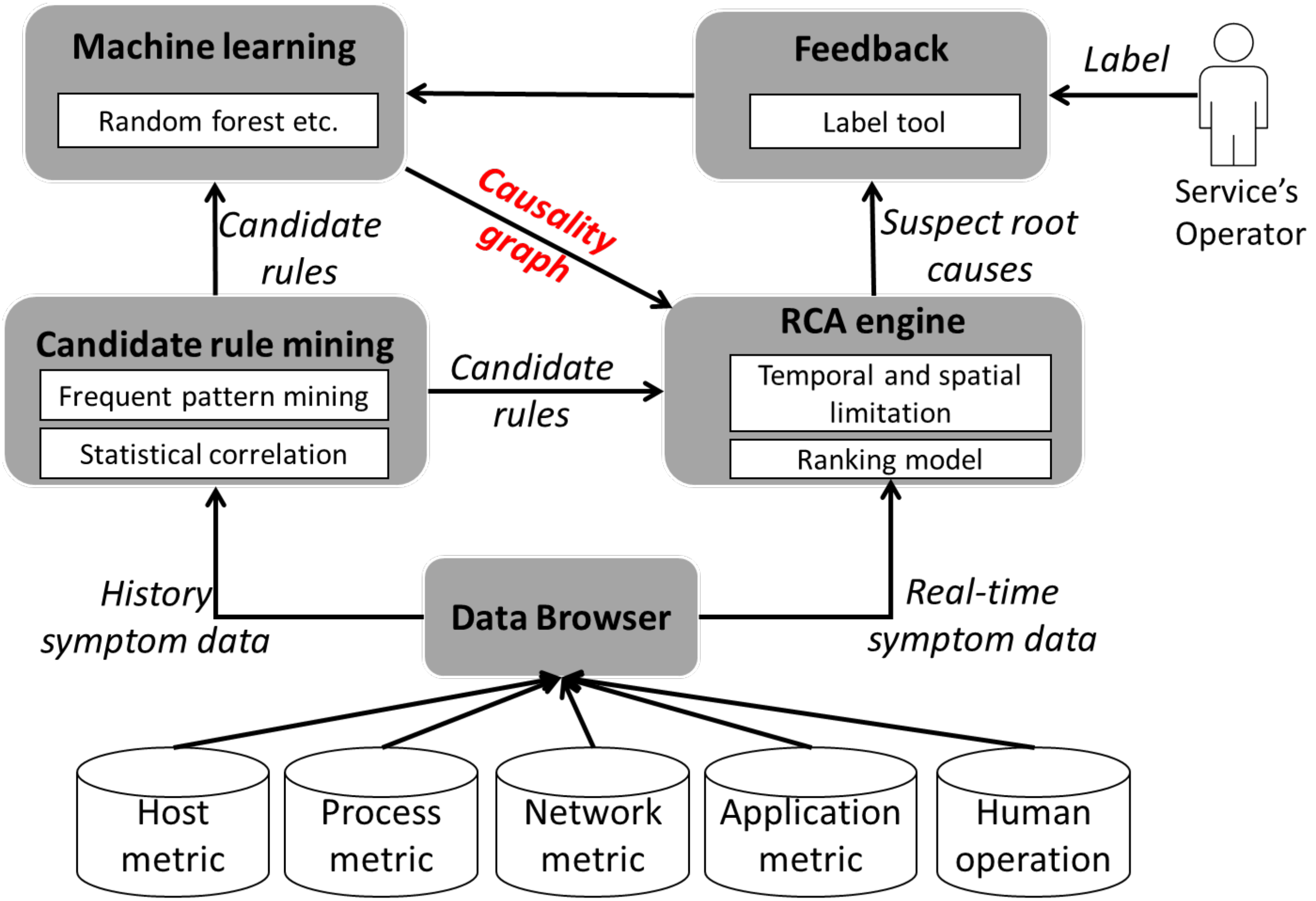
System Overview

1. **Diagnosis is an inference problem with causality graph.**

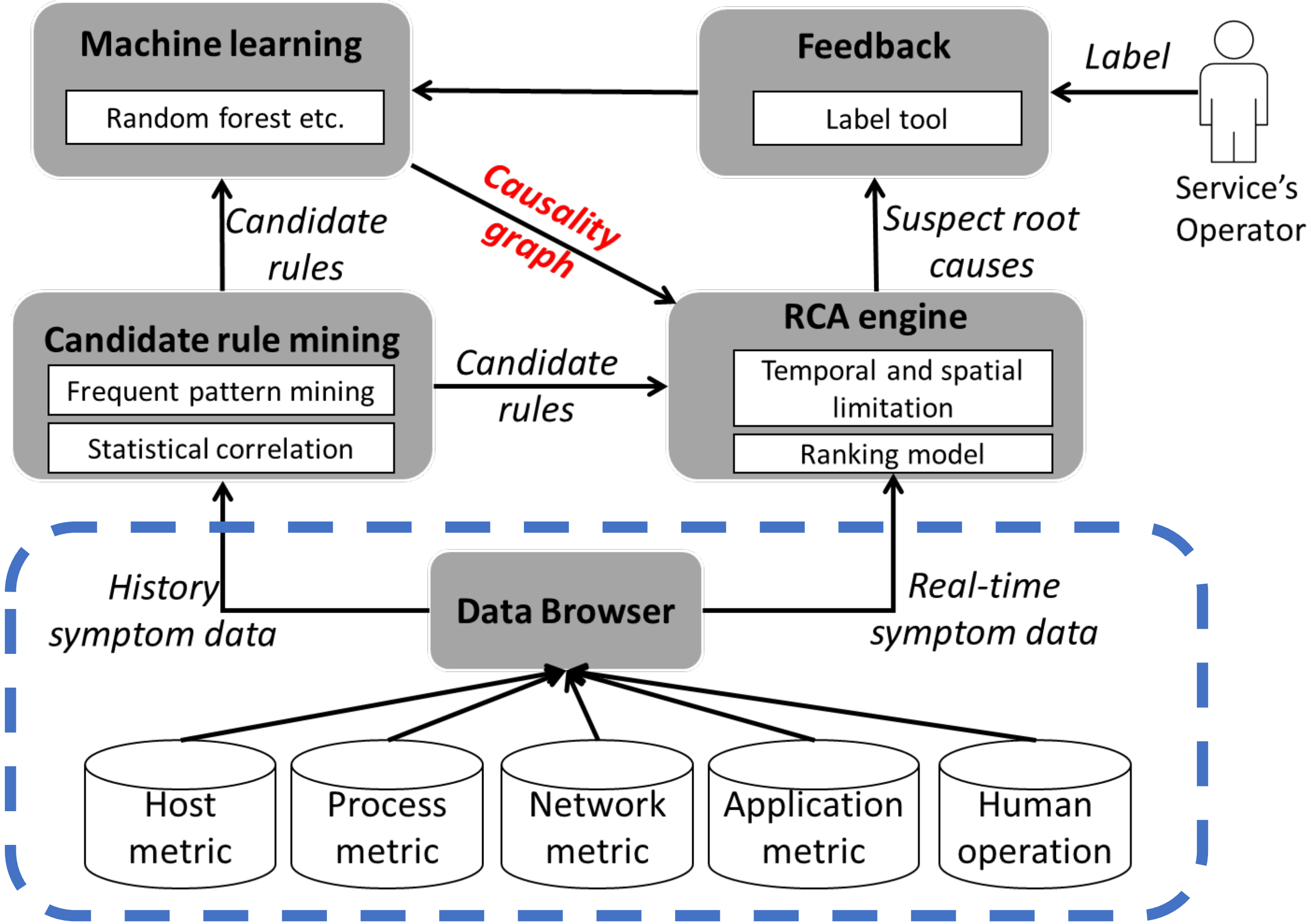
2. **Causality graph is in the operator's mind.**

3. **Our key idea is converting domain knowledge to causality graph with low overhead.**

4. **It is a supervised learning problem.**



Data Browser



Data Browser

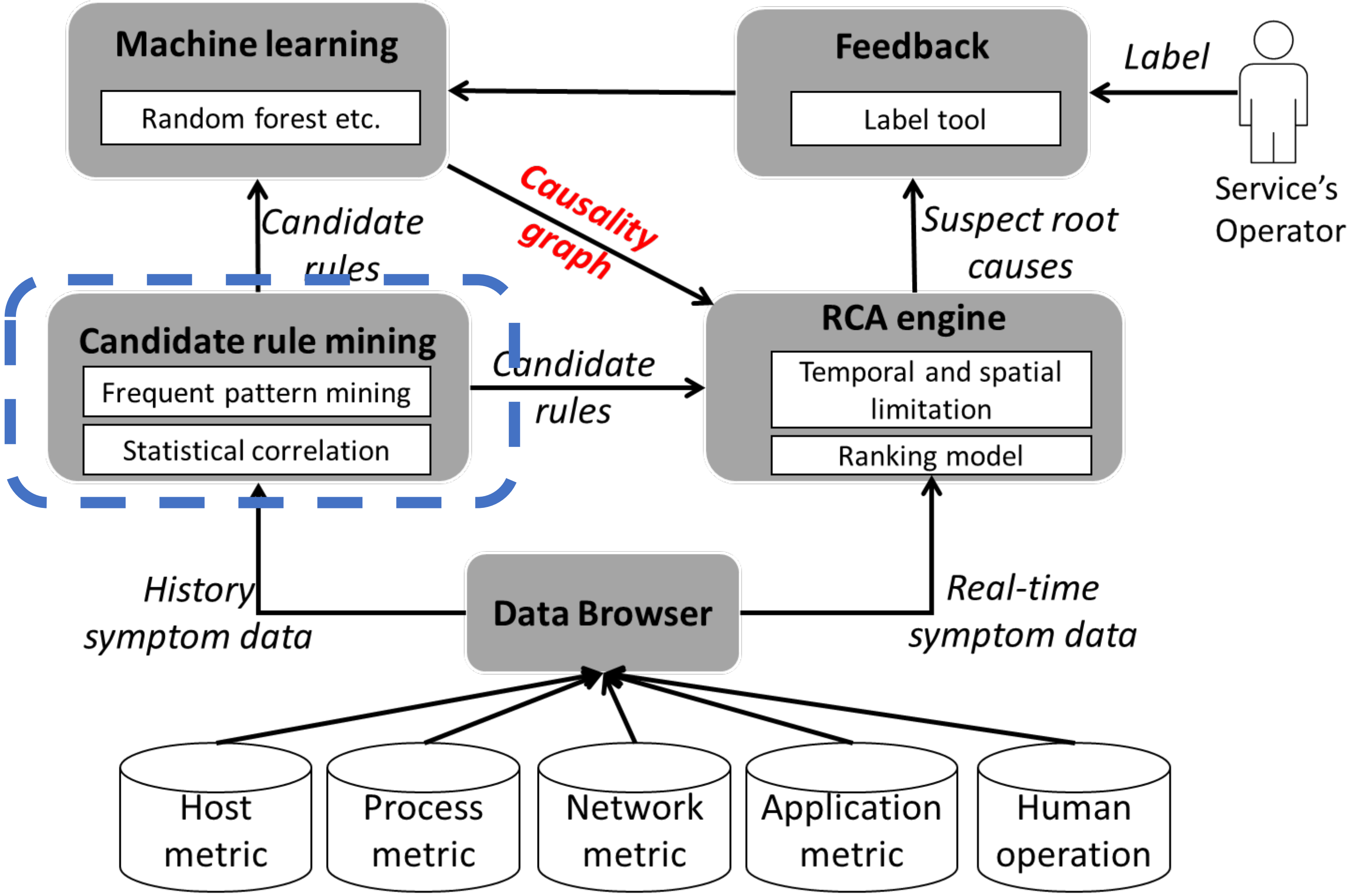
TABLE I

DESCRIPTION OF THE DATA METRIC. THE DATA IS DIVIDED INTO TWO TYPES: TIME SERIES AND EVENT SEQUENCE, EVENT SEQUENCE IS EQUAL TO 0 OR 1, 1 MEANS THE SYMPTOM EVENT HAS HAPPENED AND VICE VERSA.

Data metric	Event Description	Location	Type
Machine	CPU usage, memory usage, NIC, disk usage, context switch, *etc.*	Host	Time series
Process	CPU usage, memory usage, port status, file handle number, *etc.*	Process	Time series
Application	function return value, page view number, port status, error log number, *etc.*	Application	Time series, Event sequence
Network	network segment down, bandwidth decrease, *etc.*	Network	Time series, Event sequence
Manual operation	configuration upgrade, software upgrade	Operators' action	Event sequence

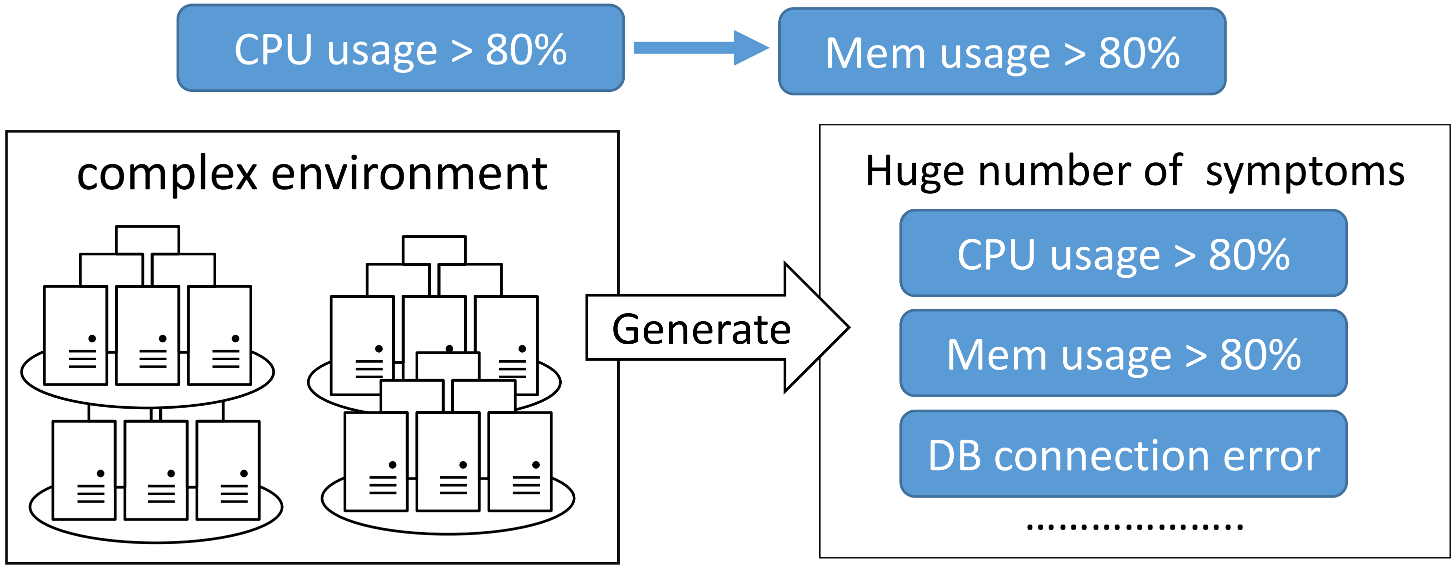
Symptom Event		
Timestamp	Name	Detail info (machine, process, application, network…)

Candidate rule mining



Candidate rule mining

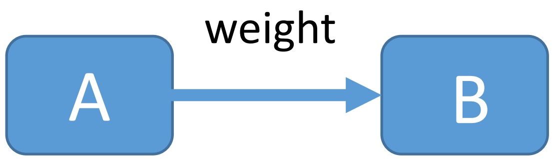
- **Rule Definition:**
  - E is symptom events set, $A, B \in E$. $A \rightarrow B$ means A will lead B happened. → presents the causality.



Candidate rule mining

- **How to mine rules?**
  - N symptom events, potential rule number = A(n, 2)
  - Frequent pattern mining

- **How to compute rules' weight(feature)?**
  - Support
  - Confidence



- **How to decide rule direction?**
  - Lag correlation

Candidate rule mining

- **How to mine rules?**
  - Mining historical data of the symptom events.
  - A rule is likely right if it is a frequent pattern.

Input

time	symptom event
2014-10-29 06:09:10	http port unreachable
2014-10-29 06:09:10	cpu usage
2014-10-29 06:10:10	page view number< 500
2014-10-29 06:11:10	mem usage
…	…

FP-growth [5]

Output

Association rules list
cpu usage – mem usage
cpu usage – page view number< 500
cpu usage – http port unreachable
http port unreachable – mem usage
…

[5] J Han, J Pei, Y Yin -Mining frequent patterns without candidate generation ACM SIGMOD Record, 2000

Candidate rule mining

- **How to compute rules' weight(feature)?**

TABLE II
RULES' FEATURES TO EVALUATE THE CORRELATION

Feature($A \rightarrow B$)	**Description**
$Support$ [13]	The frequency of A, B's concurrence
C_1 [13]	Conditional probability:$P(B\|A)$
C_2 [13]	Conditional probability:$P(A\|B)$
$Pearson$ [14]	Novel statistical pearson correlation
$Lift$ [13]	$P(AB)/((P(A) * P(B)))$
$KULC$ [13]	$(P(A\|B) + P(B\|A))/2$
IR [13]	$P(A)/(B)$
Location relation	A, B happened in the same host, cluster, software component or not

Pearson

Time window　Time window　Time window

A 00001000000000000010000000000001000000

B 0010000000000000100000000000010000000

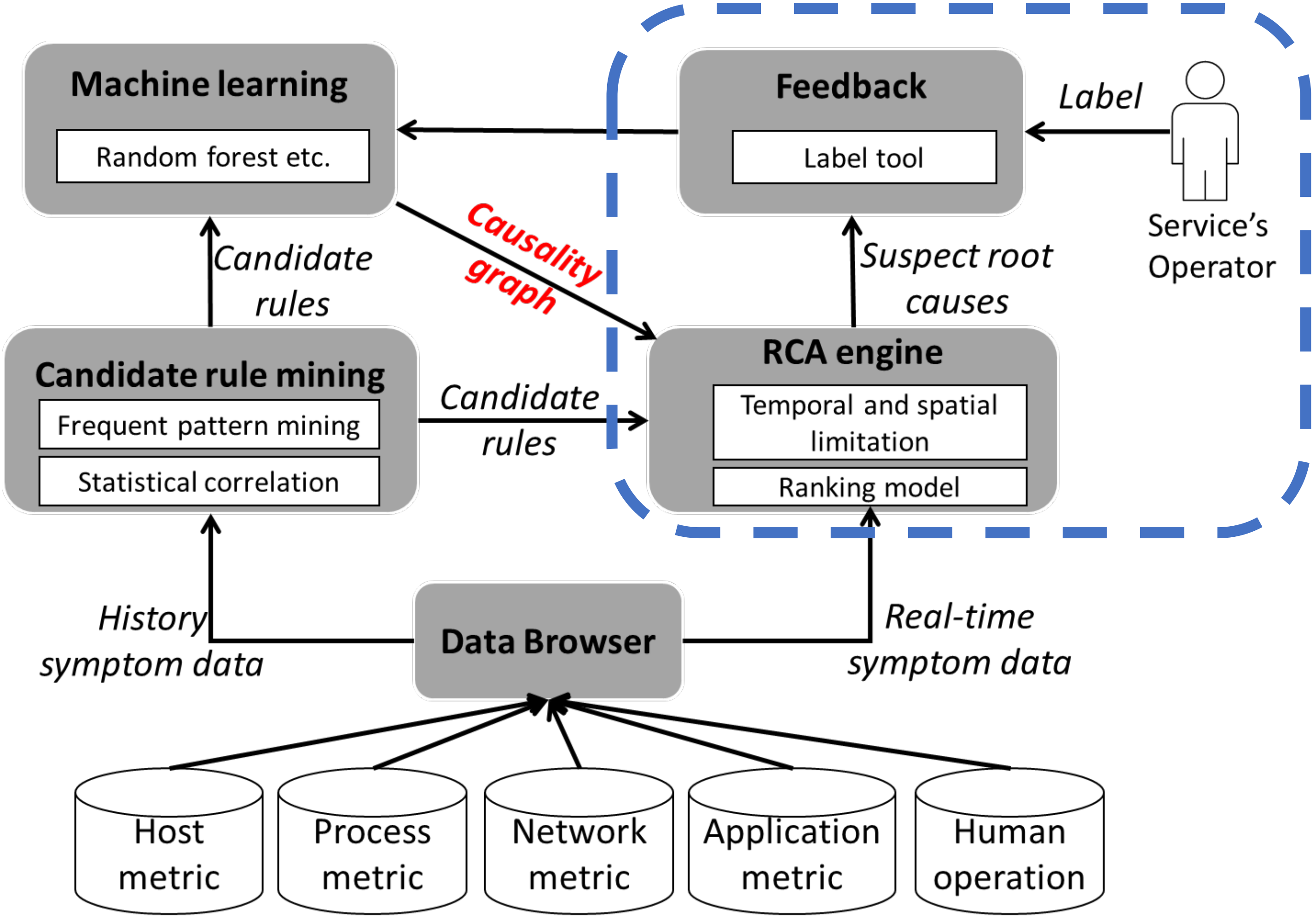
00001000000000000010000000000001000000

00001000000000000010000000000001000000

$$Pearson_{old} = -0.088$$

$$Pearson_{new} = 1$$

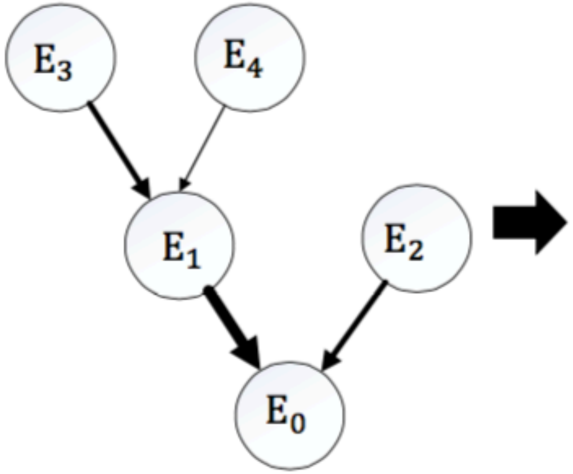
RCA engine and Feedback



RCA engine

- **Root cause analysis:**
  - **Temporal and spatial limitation**
  - **Ranking model**
    - Greedy method(depth-first)

$$W_{r(e_1 \to e_2)} = \begin{cases} 0.5, default \\ F(f_1, f_2, f_3, ...), F \in [0,1] \end{cases} \quad (2)$$



Inference

Root Causes	Interference
E_3	$E_3 \to E_1 \to E_0$
E_4	$E_4 \to E_1 \to E_0$
E_2	$E_2 \to E_0$

⟶ Thicker arrow means lager $W_{r(E_i \to E_j)}$

Fig. 4. A simple example of root cause ranking

Feedback

Root cause analysis

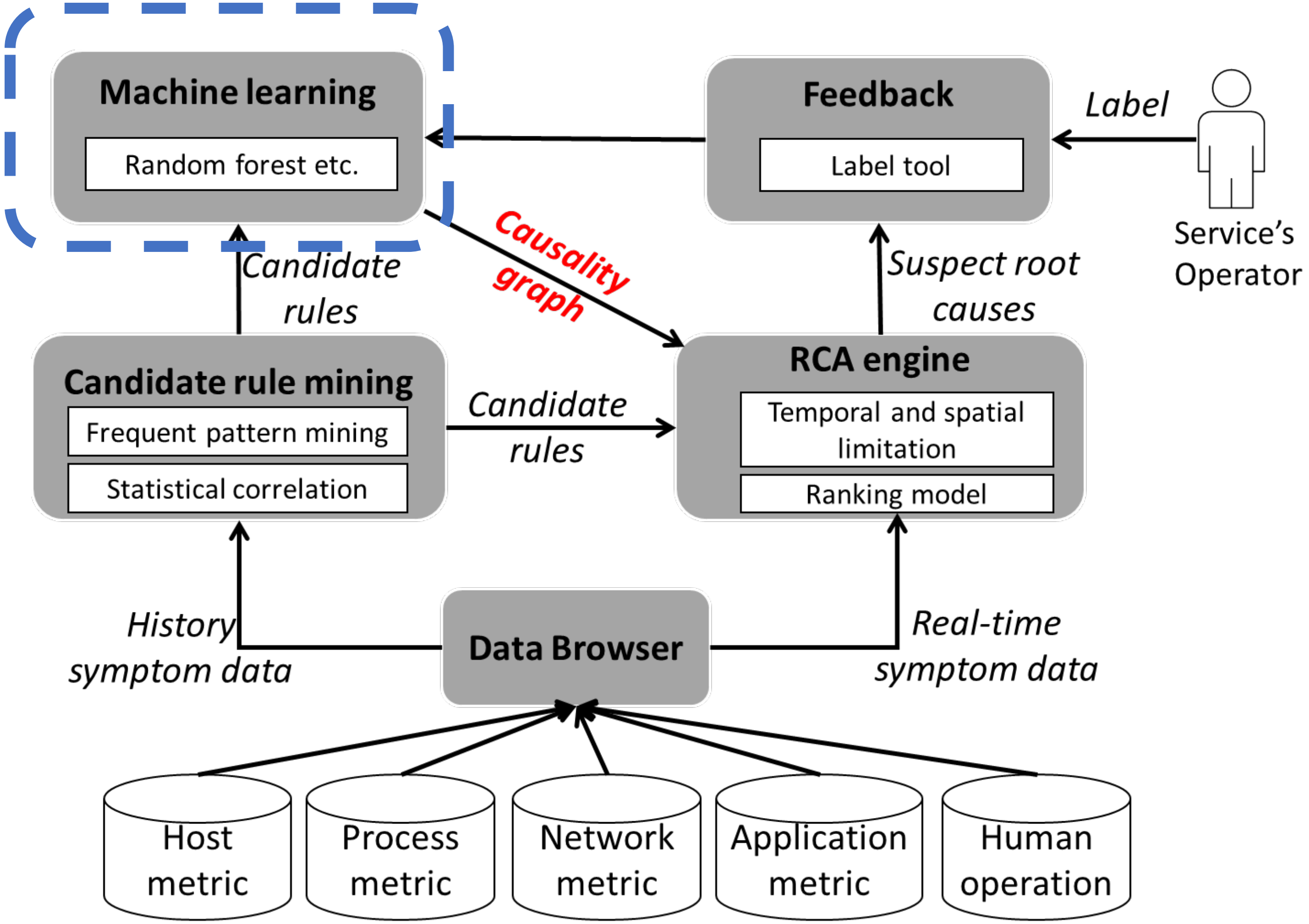
es_quan_url_monitor

Root Causes	Inference	Feedback
Sms_center_service_unavailable	Sms_center_service_unavailable → es_quan_url_monitor	
sh01_client_alive_and_response	sh01_client_alive_and_response → Sms_center_service_unavailable → es_quan_url_monitor	

Refresh

uncertain right wrong rules are all right

Machine learning



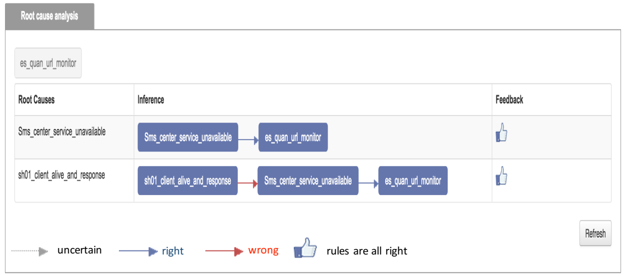
Machine learning

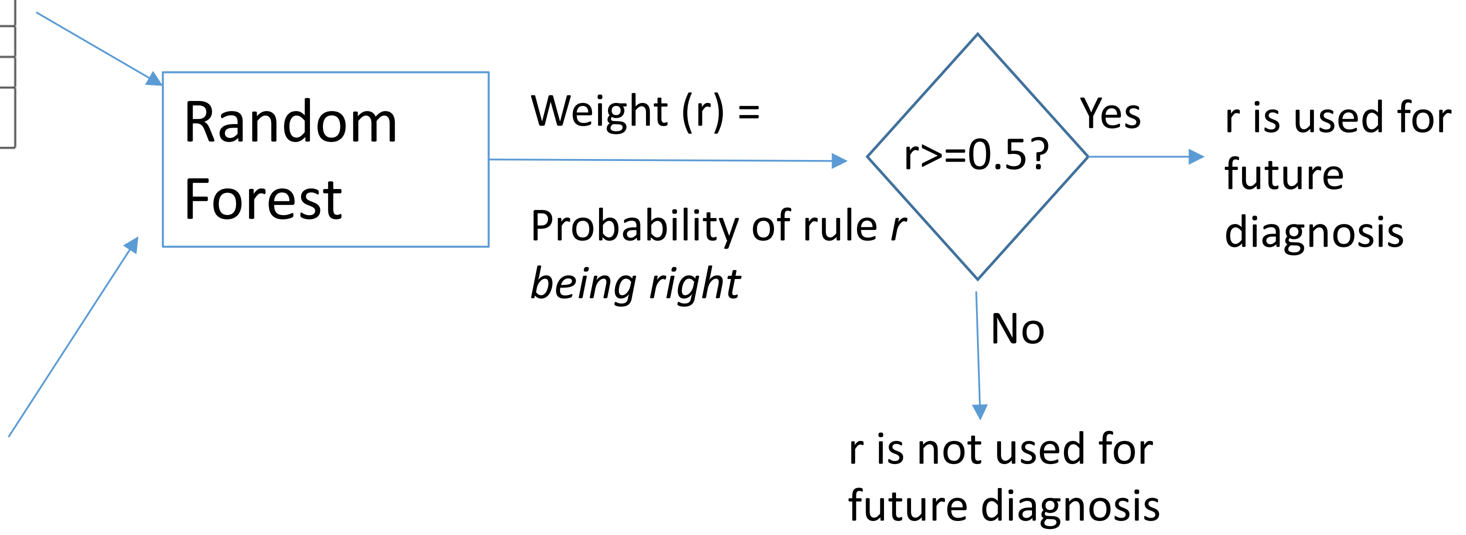
Features of rules

TABLE II
RULES' FEATURES TO EVALUATE THE CORRELATION

Feature($A \rightarrow B$)	Description
$Support$ [13]	The frequency of A, B's concurrence
C_1 [13]	Conditional probability:$P(B\|A)$
C_2 [13]	Conditional probability:$P(A\|B)$
$Pearson$ [14]	Novel statistical pearson correlation
$Lift$ [13]	$P(AB)/((P(A) * P(B)))$
$KULC$ [13]	$(P(A\|B) + P(B\|A))/2$
IR [13]	$P(A)/(B)$
Location relation	A, B happened in the same host, cluster, software component or not

Labels of rules





Controlled experiment

- Because the ground truth of the web-based service can not be obtained easily, we evaluate our system though a controlled experiment with explicit ground truth.

Assumption:

1. Root causes are the leaf nodes in the ground truth.
2. Edges and its direction means the causality.
3. Feedback is based on ground truth.

Data simulation:

1. Randomly let one root cause event happen in every 15 minutes.
2. Add noisy events (e11~e29) to co-occur with the root causes.
3. One month data.

Diagnosis:

1. Do root cause analysis (RCA) when e0 is happened
2. Every 4 times of RCA triggers machine learning.

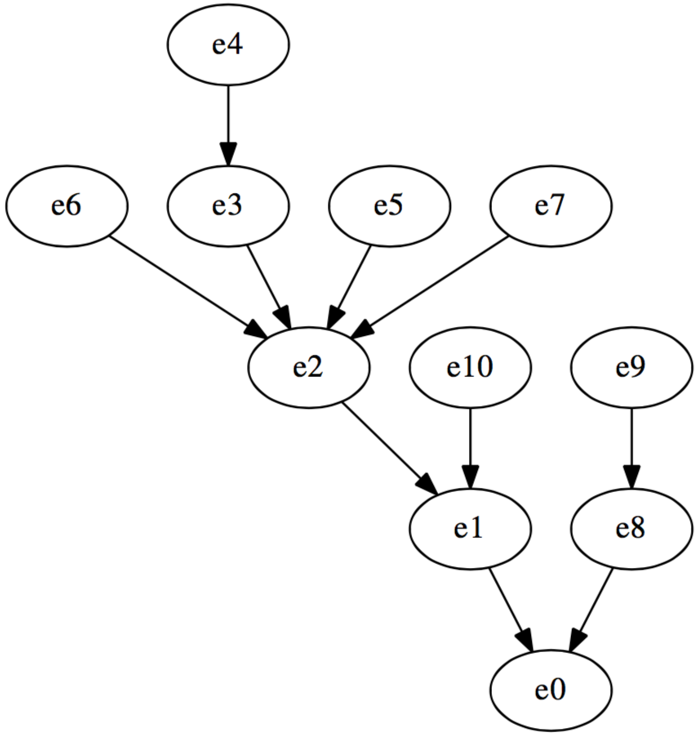


Fig. 6. Ground truth of a simple causality graph.

Evaluation of Machine learning method

- **Random Forest is the most suitable algorithm**
  1. Accuracy
  2. Speed

TABLE III
THE CONFIGURATION OF FIVE MACHINE LEARNING ALGORITHMS IN OUR EXPERIMENT.

Algorithm	Sampled Parameters
J48 (Decision tree)	confidenceFactor = 0.25, minNumObj = 2, numFolds = 3, seed = 1
NaiveBayes	useKernelEsimator = false, useSuper-viseDiscretization = false
Random Forest	maxDepth = newFeatures = 0, numTrees=100, seed =1
RBFNetwork	clusteringSeed = 1, numClusters = 2, min-StdDev = 0.1, ridge = $1.0E-8$, maxIts = -1
Logistic	ridge = $1.0E-8$, maxIts = -1

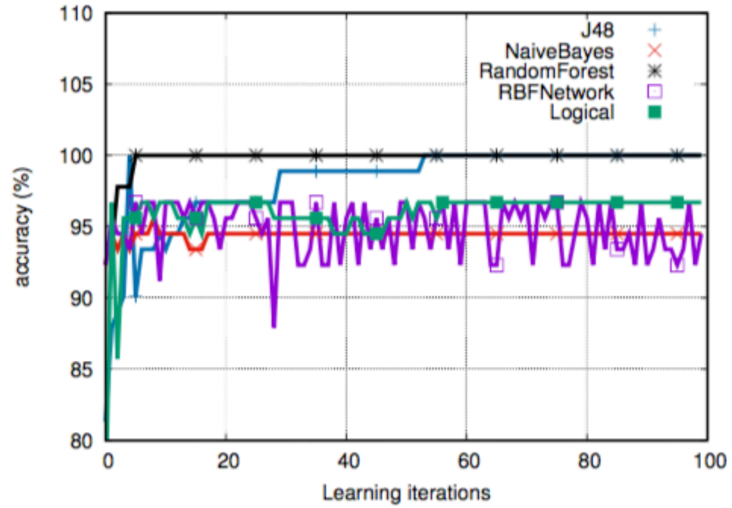


Fig. 7. Different algorithms' accuracy at different learning iteration, the x-axis means the iteration times of learning.

Evaluation of causality graph

- 29 times of RCA and feedback, our system can learn the causality graph.
- This result show our system can learn the causality graph.

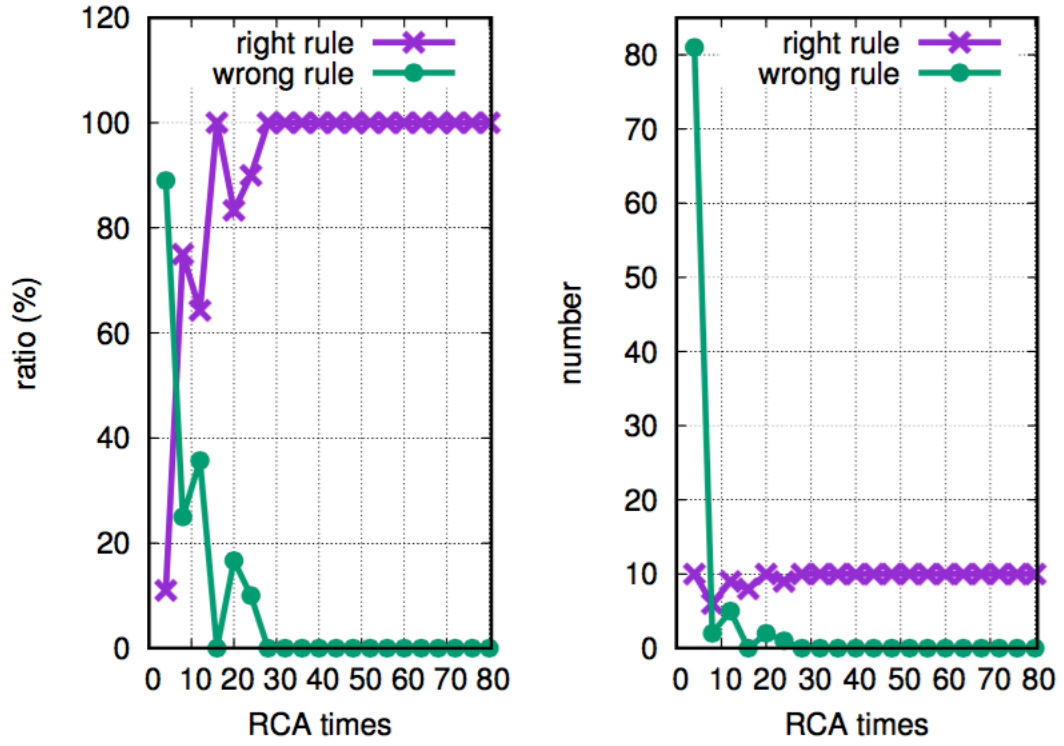


Fig. 8. The rules in causality graph. "RCA times" means the number of each diagnosis for failure event e_0.

Whether root causes are listed in top-3?

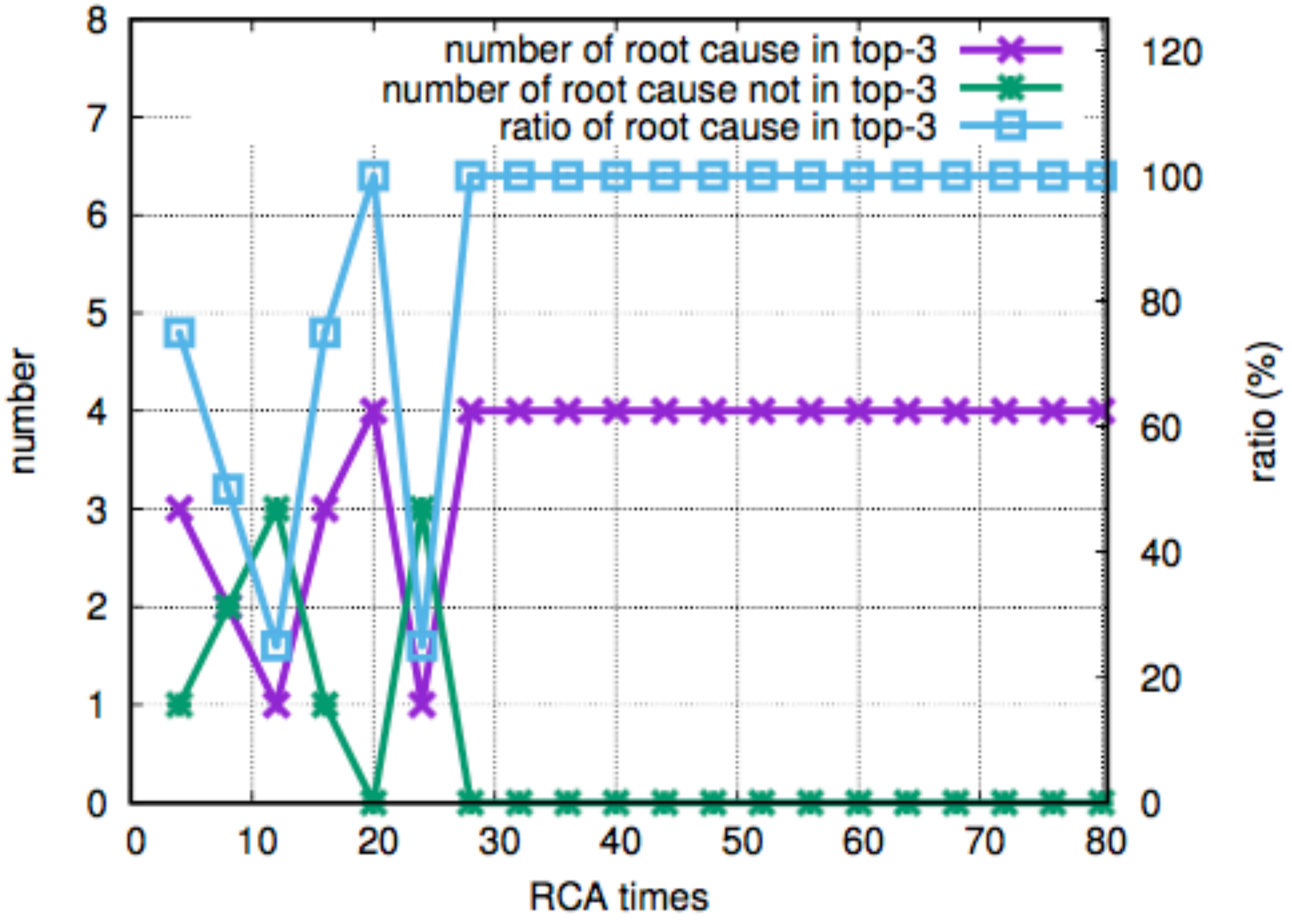
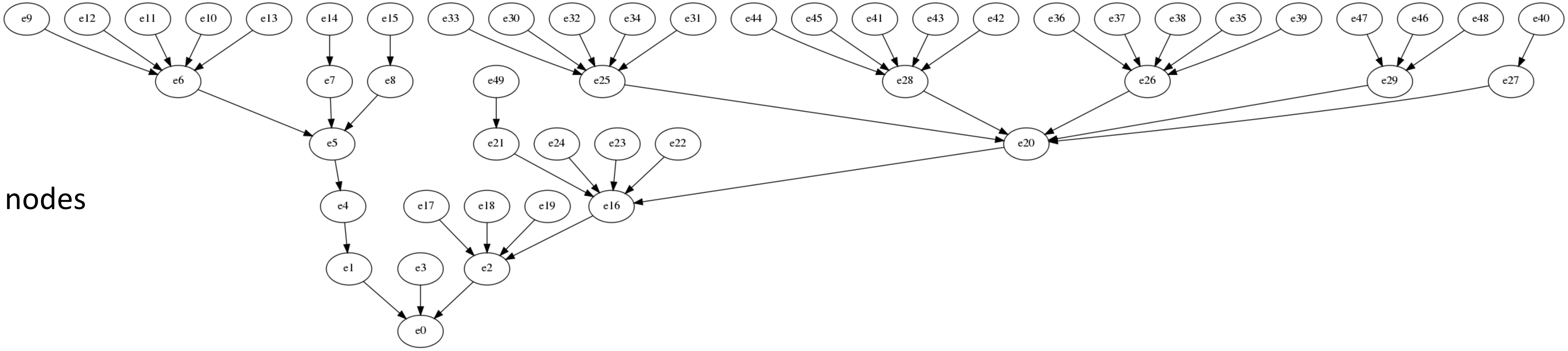
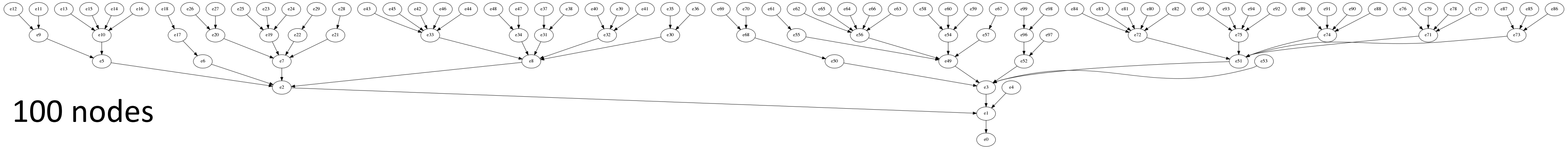


Fig. 9. The ratio of root cause in top-3.

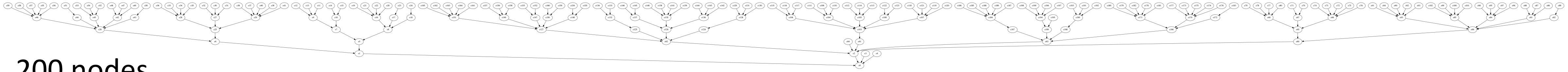
Evaluation of complex ground truth



50 nodes



100 nodes



200 nodes

Evaluation of complex ground truth

The ratio of right rule = $\frac{\#right\ rule}{\#\ the\ rule\ in\ causality\ graph}$

The ratio of right rule = $\frac{\#right\ rule}{\#\ the\ rule\ in\ ground\ truth}$

The ratio of root cause in top3 = $\frac{\#root\ causes\ in\ top3}{\#\ all\ the\ root\ causes}$

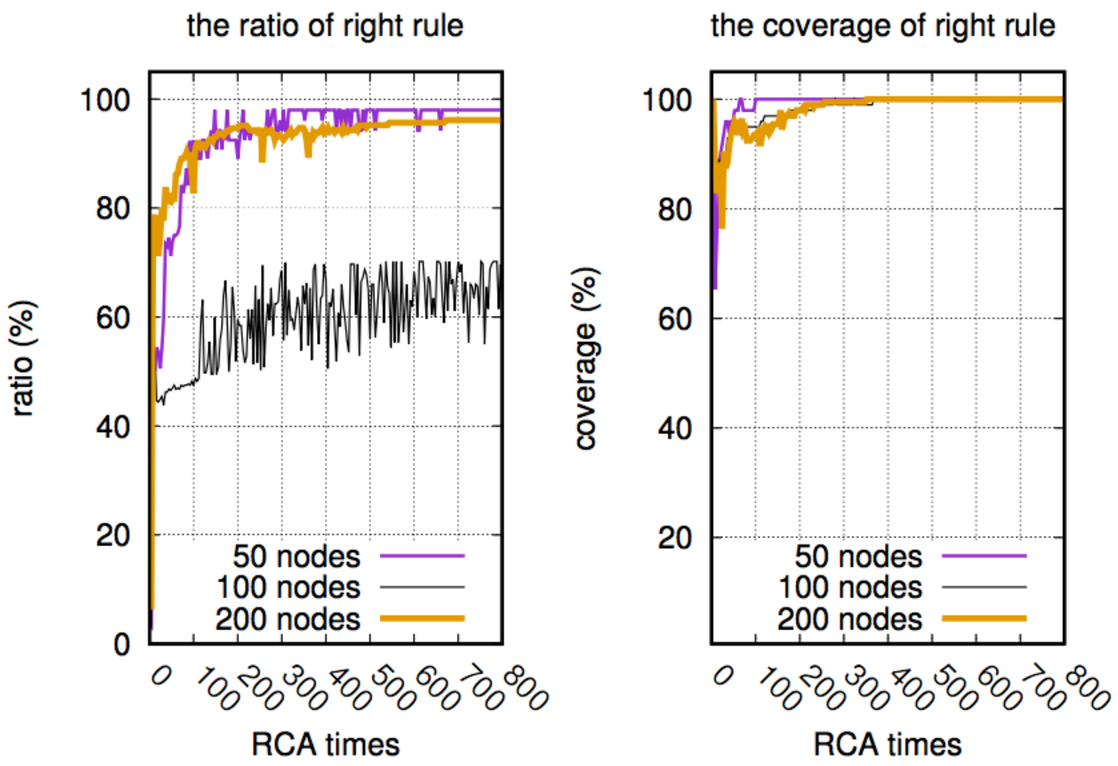


Fig. 10. The learning result of complex ground truth

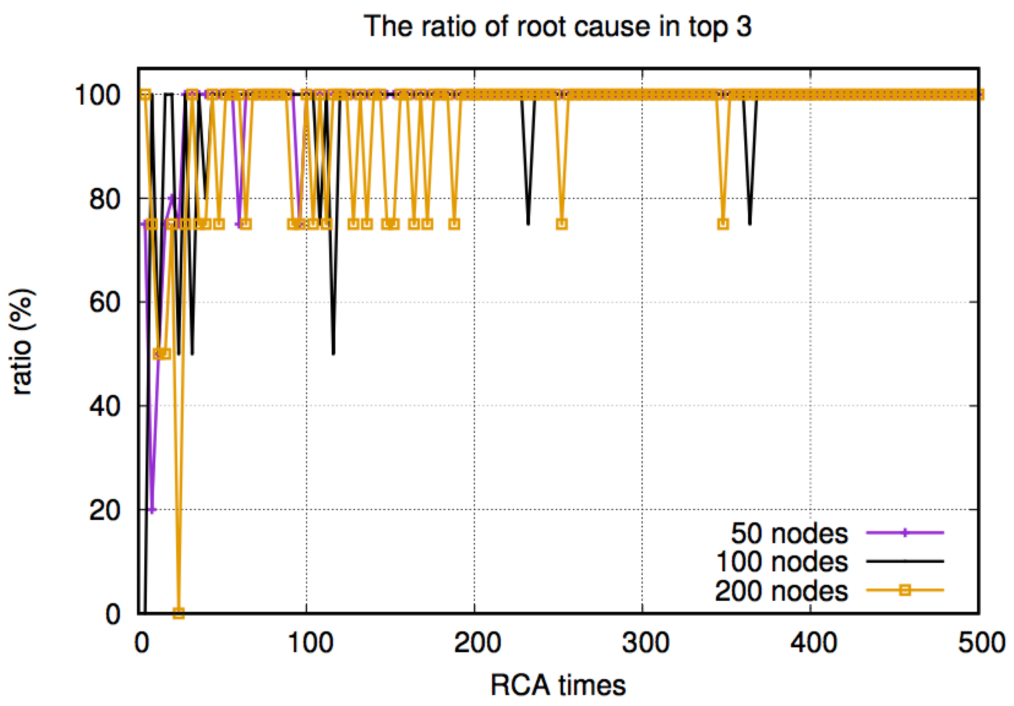


Fig. 11. The ratio of root causes in top 3.

Conclusion

1. we propose a generic diagnosis system for web-based services.
    - Based on causality graph.
    - Learn from operators' experiment.
    - Utilize data mining and machine learning
    - Low overhead.

2. Root causes can be ranked in top 3 with 100% accuracy after countable learning iterations.

Thanks

Xiaohui Nie

nxh15@mails.tsinghua.edu.cn