Mining causality of network events in log data

National Institute of Informatics
Project Researcher
Satoru Kobayashi
Oct 17, 2018

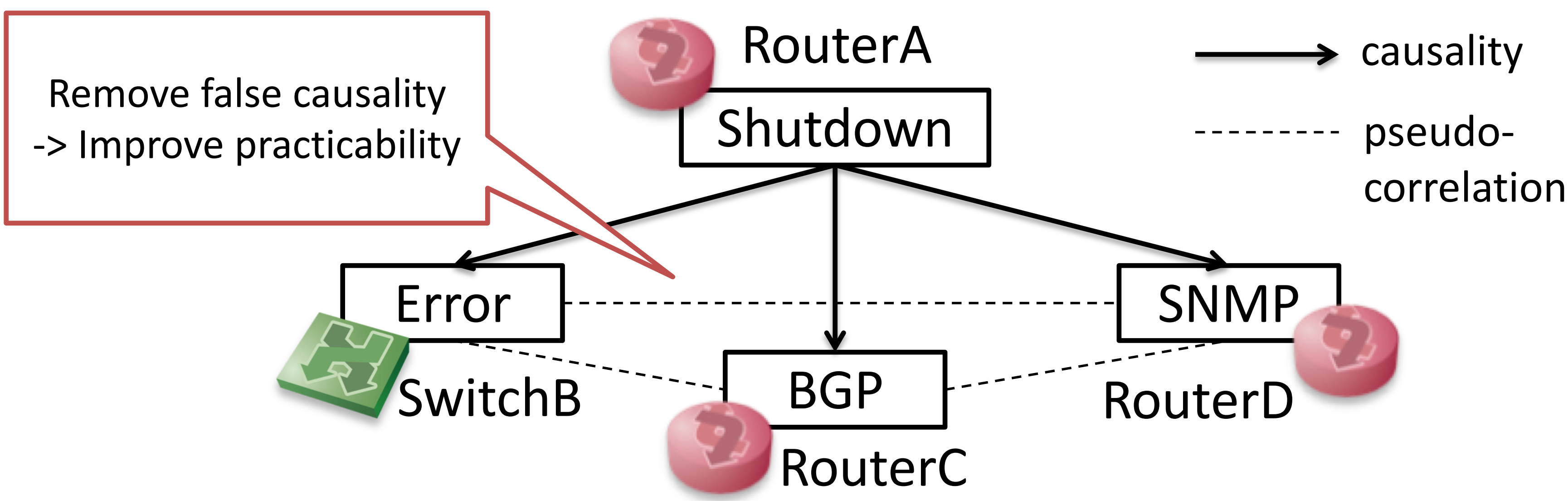
Inter-University Research Institute Corporation / Research Organization of Information and Systems
National Institute of Informatics

About this presentation

- Speaker: Satoru Kobayashi, Ph.D.
  - Postdoc researcher in National Institute of Informatics
  - Expertise: Network management, Data mining, ...
- Related paper
  - "Mining causality of network events in log data" S.Kobayashi et al, in IEEE TNSM, 2018
- Source code:
  - https://github.com/cpflat/LogCausalAnalysis

Challenge of this research

Oct 17 17:00:00 routerA System shutdown by root
Oct 17 17:00:05 switchB Error detected on eth0
Oct 17 17:00:15 routerC BGP state changed from Established to Idle
Oct 17 17:00:15 routerD SNMP trap sent to routerA
……



Outline

- Background
- Causal analysis
- System architecture
  - 4 underlying methods
- Evaluation
- Conclusion

Network management with data

- 3 tasks for operating large-scale network
  1. Continuous monitoring
  2. Fast recovery of failures
  3. Relapse prevention of failures
- Operational data
  – Used to find and analyze failures
  – Large variety
  – Large-scaled data -> Automated analysis

Active / Passive analysis of operational data

	Active analysis	Passive analysis
Example	• Traffic data • Routing table • Load average • Temperature • …	• System log • Access log • Trouble ticket
Feature	• For failures assumed in advance • Environment-dependent design	• Independent of the environment • Flexible data • Difficult to analyze automatically

System log

- System log
  - e.g., syslog
  - Contextual information (compared to measurement data)
- Difficulty in automated analysis
  - Free-format (Unclear message structure)
  - Mixture of frequent and sparse logs
  - Lengthy / Repeated data

```
Aug  1 12:43:55 host1 %%01VTY/5/ACL_DENY(l)[61179]: The TCP request was denied according to ACL rules. (IpAddress=192.0.2.1)
Aug  1 12:45:20 host1 %%01MCAST/6/SUPPRESS_REPORT(l)[61180]: Suppress report packet. (VlanID=64500, Group ip=192.0.2.100, Rece
iveInterface=Eth-Trunk0)
Aug  1 12:49:12 host1 %%01VTY/5/ACL_DENY(l)[61181]: The TCP request was denied according to ACL rules. (IpAddress=192.0.2.3)
Aug  1 12:57:50 host1 %%01INFO/4/SUPPRESS_LOG(l)[61182]: Last message repeated 1 times.(InfoID=1079644206, ModuleName=VTY, InfoA
lias=ACL_DENY)
```

Automated analysis of system log

3 challenges
for operating network

Continuous monitoring

- Anomaly detection
  - State modeling [1,2]
  - Bayesian estimation [3]

Fast recovery of failures

- Fault localization
  - State modeling [1,4]
  - Spatial analysis [5]

Relapse prevention of failures

- Root cause analysis
  - Heuristic-based [6]
  - Causal inference [7,8]

[1] K. Yamanishi et al. "Dynamic syslog mining for network failure monitoring". In ACM KDD’05, p. 499, 2005.
[2] F. Salfner et al. "Using hidden semi-Markov models for effective online failure prediction". In IEEE SRDS, pp. 161–174, 2007.
[3] P. Chen et al. "Causeinfer: Automatic and distributed performance diagnosis with hierarchical causality graph in large distributed systems". In IEEE INFOCOM, pp. 1887–1895, 2014.
[4] I. Beschastnikh, et al. "Inferring Models of Concurrent Systems from Logs of Their Behavior with CSight." In ICSE 2014, 468-479, 2014.
[5] T. Kimura et al. "Spatio-temporal factorization of log data for understanding network events". In IEEE INFOCOM, pp. 610–618, 2014.

Causal analysis of system log

- Problems of existing methods [7,8]
  - Large processing time
  - Analysis range is limited
  - Only considering logs of close time
- Graph-based causal analysis
  - Efficient analysis
  - Enable to point out root causes
  - Exploratory approach -> knowledge extraction

[6] B. Tak et al. "LOGAN: Problem Diagnosis in the Cloud Using Log-Based Reference Models," in IEEE IC2E, 2016, pp. 62-67.
[7] Z. Zheng et al. "3-Dimensional root cause diagnosis via co-analysis," in ACM ICAC, 2012, pp. 181.
[8] A. Mahimkar et al. "Towards automated performance diagnosis in a large iptv network," in ACM SIGCOMM, 2009, pp. 231–242.

Goal

- Infer causal relations among network events in log data
  - Time series analysis + Causal inference
  - Exploratory approach with wide-range data
  - Available in large-scale network
- Troubleshooting of system failures
  - Operators can understand system behavior easily

Dataset

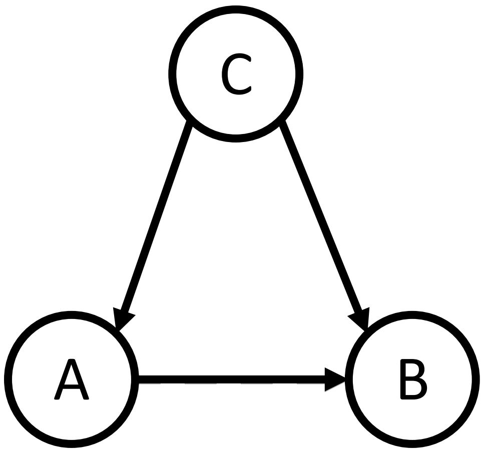
- SINET4
  - (https://www.sinet.ad.jp/en/top-en)
  - A nation-wide R&E network in Japan
  - 8 core routers and 100 over L2 switches
  - 15 months syslog data
    - 3.5 million lines to analyze
  - 12 months trouble ticket
    - For evaluations

Outline

- Background
- Causal analysis
- System architecture
  - 4 underlying methods
- Evaluation
- Conclusion

Causal Inference

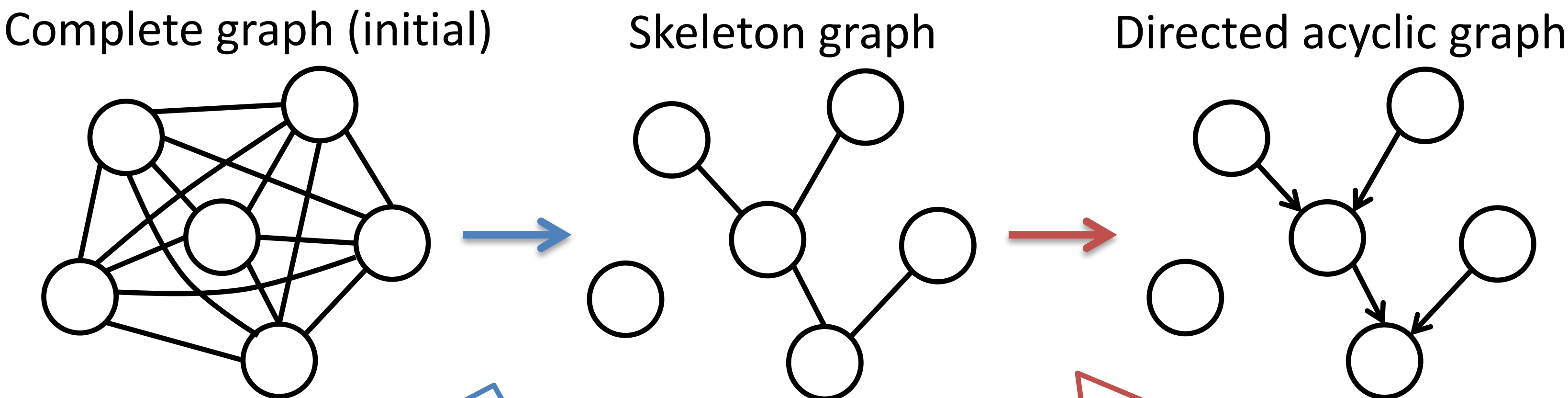
- Conditional Independence
  - A and B are independent if the effect of confounder C is excluded
  - A and B are conditionally independent given C
- PC algorithm [9]
  - Directed acyclic graph (DAG)
  - Explore conditional independence and remove false edges



$$P(A|C)P(B|C) = P(A,B|C)$$

[9] P. Spirtes et al. "An algorithm for fast recovery of sparse causal graphs", Social science computer review, vol. 9, pp. 62–72, 1991.

Flow of PC algorithm



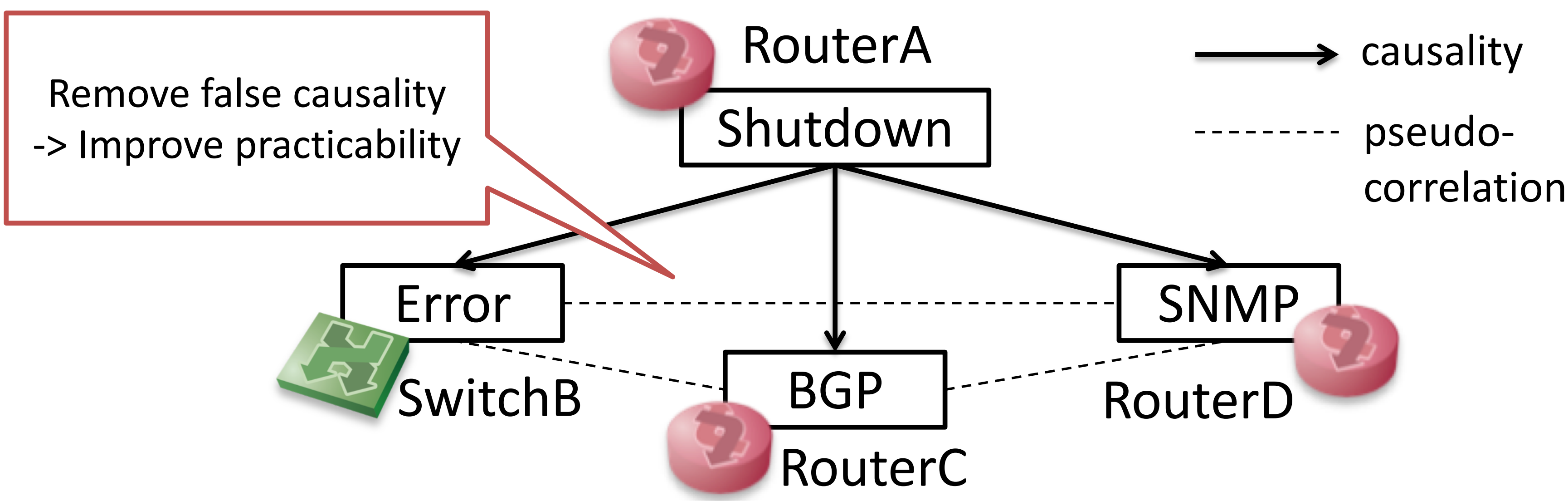
- Remove edges of conditional independence
- Statistical test for conditional independence
    - G2 test (for binary or multi-level data)
    - Fisher-Z test (for continuous data)

ps)

[10] R. E. Neapolitan. "Learning Bayesian Networks." Prentice Hall Upper Saddle River, 2004.
[11] T. Verma, et al. "An algorithm for deciding if a set of observed independencies has a causal explanation". In Proceedings of UAI'92, pp. 323–330, 1992.

Log analysis and causal inference

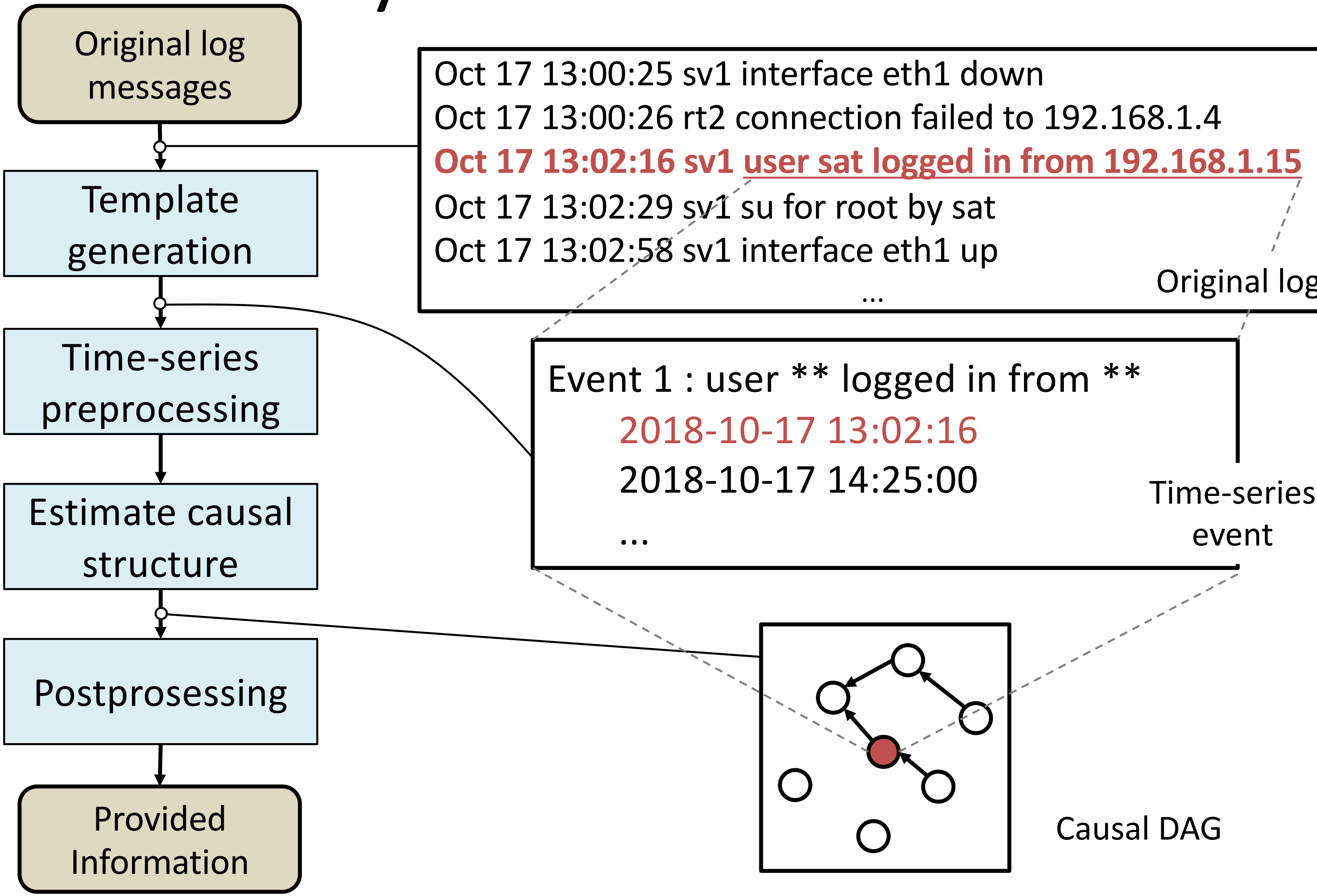
```
Oct 17 17:00:00 routerA System shutdown by root
Oct 17 17:00:05 switchB Error detected on eth0
Oct 17 17:00:15 routerC BGP state changed from Established to Idle
Oct 17 17:00:15 routerD SNMP trap sent to routerA
……
```



Outline

- Background
- Causal analysis
- System architecture
  - 4 underlying methods
- Evaluation
- Conclusion

System architecture



System ar…

Correspondence to 4 challenges

- Original log messages
- Template generation
- Time-series preprocessing
- Estimate causal structure
- Postprosessing
- Provided Information

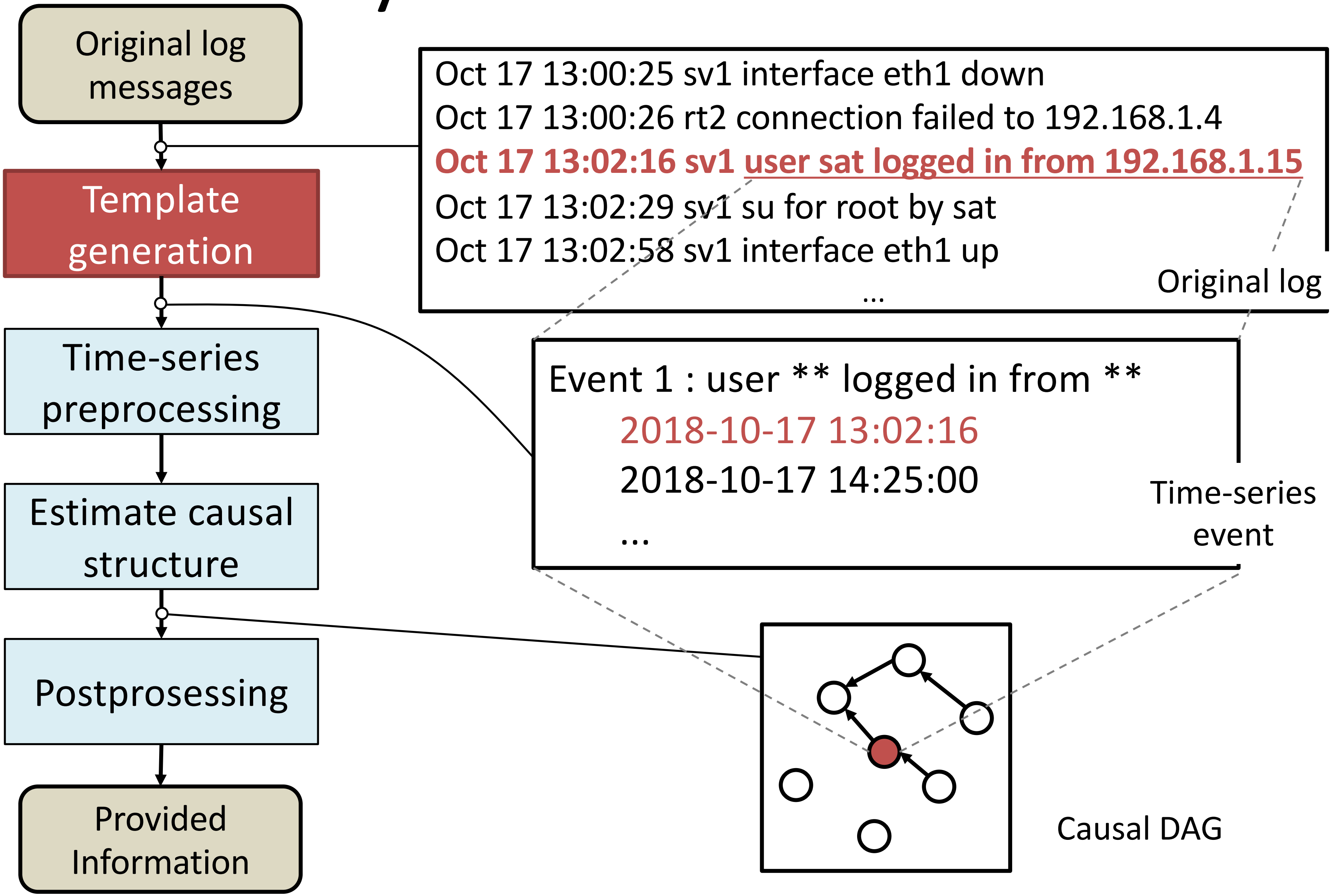
Difficulty of syslog analysis:
String with timestamp, not time-series

Practicability for operation:
Redundant (not important) messages

Difficulty of syslog analysis:
Sparse appearance of error messages

Practicability for operation:
Self-evident relations

System architecture



Log template

- Log template

Original log	Accepted password for sat from 192.168.1.1 port 12345
Log template	Accepted password for * from * port *

- Why log template is needed for analysis?
  - Generate time-series by message classification
  - Extract contextual properties
  - Most existing works use format-based classification

Log template generation methods

- 3 different approaches
  1. Generating templates from source codes
     - Most accurate approach
  2. Clustering messages
     - Major approach
  3. Estimating template structure
     - (Our contribution)
     - Effective in network logs

1. Template generation from source codes

- Extract logging functions in source codes [12, 13]
- Advantage
  - No estimation failure
  - Independent from message distribution
- Disadvantage
  - Available only in open source software
    - ➢Usually unavailable in commodity network devices

[12] W. Xu et al. “Detecting large-scale system problems by mining console logs”. In ACM SOSP , pp. 117-132, 2009.
[13] M. Zhang, et al. "GenLog: Accurate Log Template Discovery for Stripped X86 Binaries". In 2017 IEEE COMPSAC, pp. 337–346, 2017.

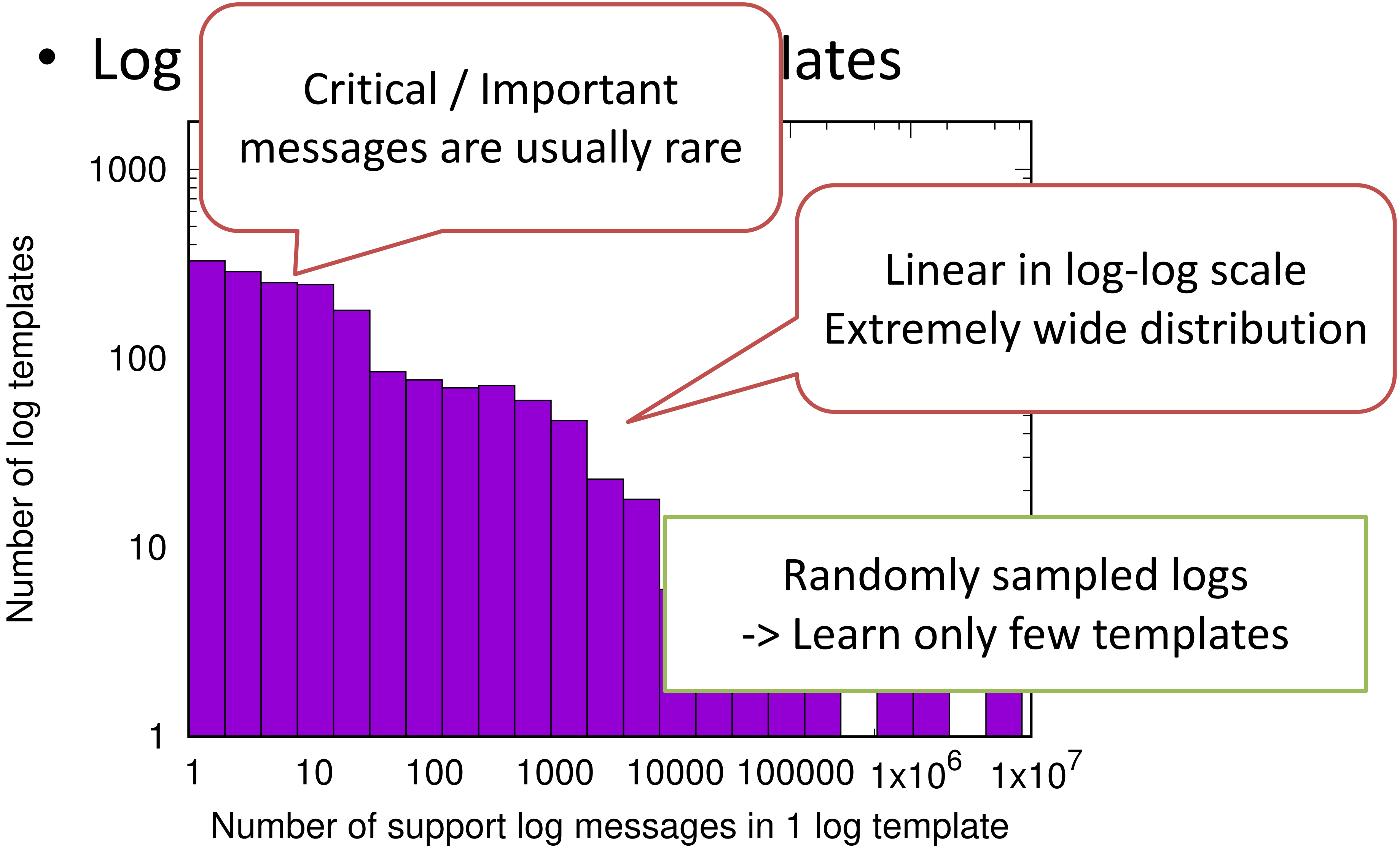
2. Clustering messages for generating templates

- Clustering messages and extract common part
  - Words, word locations or message length [14, 15]
- Advantage
  - Especially accurate in major log messages
- Disadvantage
  - Not accurate in minor log messages
  - Critical / Important messages are usually minor in network logs

[14] R. Vaarandi. “A data clustering algorithm for mining patterns from event logs”. In IEEE IPOM , pp.119-126, 2003.
[15] M. Mizutani. “Incremental mining of system log format”. In IEEE SCC’13, pp. 595–602, 2013.

Challenge of template generation of network logs

- Log … lates



Problem statement of 3. Template structure estimation

- Classify words into **Descriptions** and **Variables**
  - **Descriptions** are common in message instances
  - **Variables** can be different in message instances
  - Log template: word structure of message, presented in **Description** and **Variable** (**Variable** -> wild card)

Description **Variable**

Accepted password for sat from 192.168.1.1 port 12345

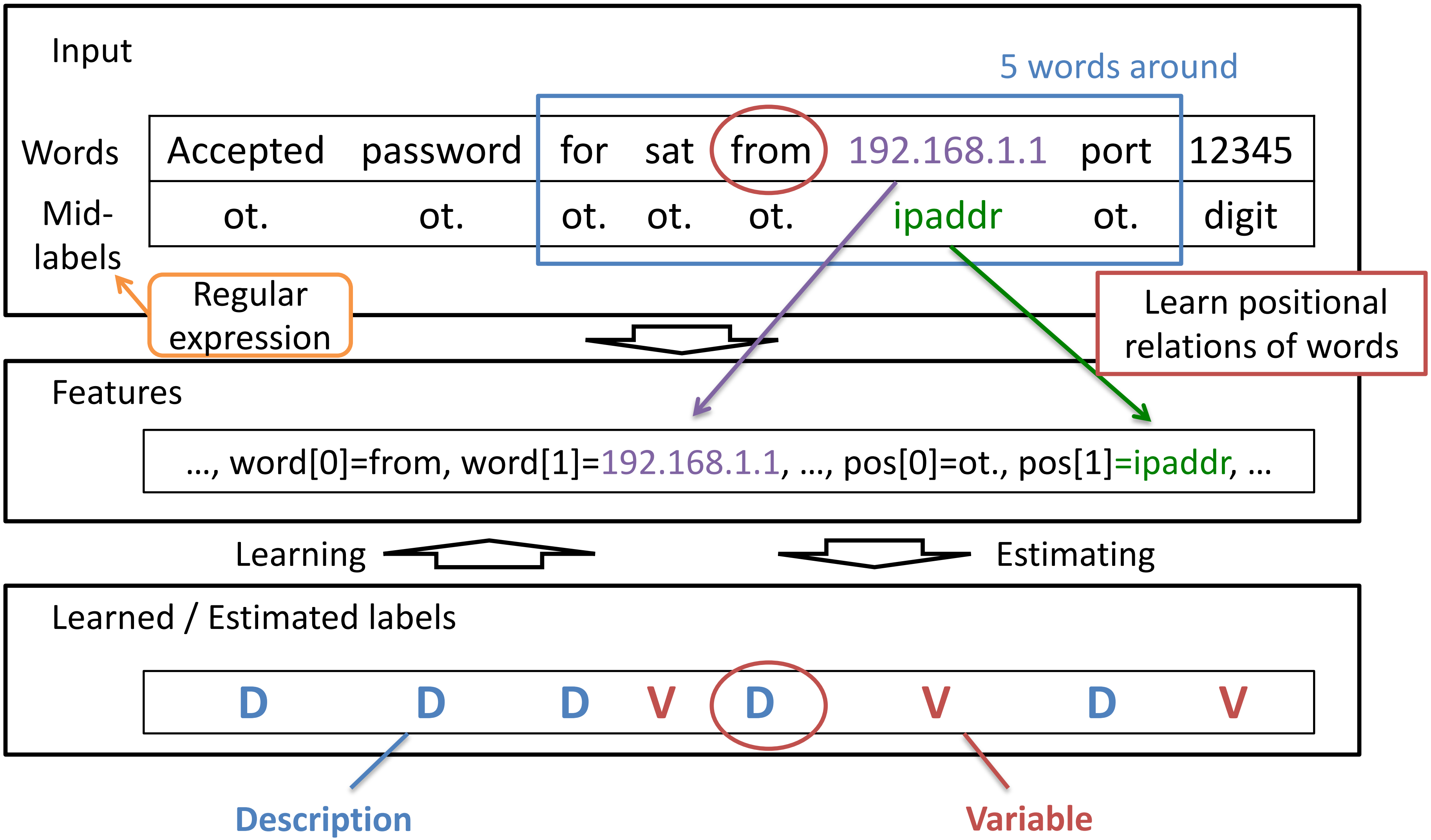
Accepted password for * from * port *

Structure learning for template generation

- Machine learning based template generation [16]
  - Labeling words as Description or Variable
- Conditional Random Fields (CRF) [17]
  - Supervised Learning
  - Labeling sequential data
  - Mainly used in Natural Language Processing
    - Part-of-speech tagging, Chunking, etc...

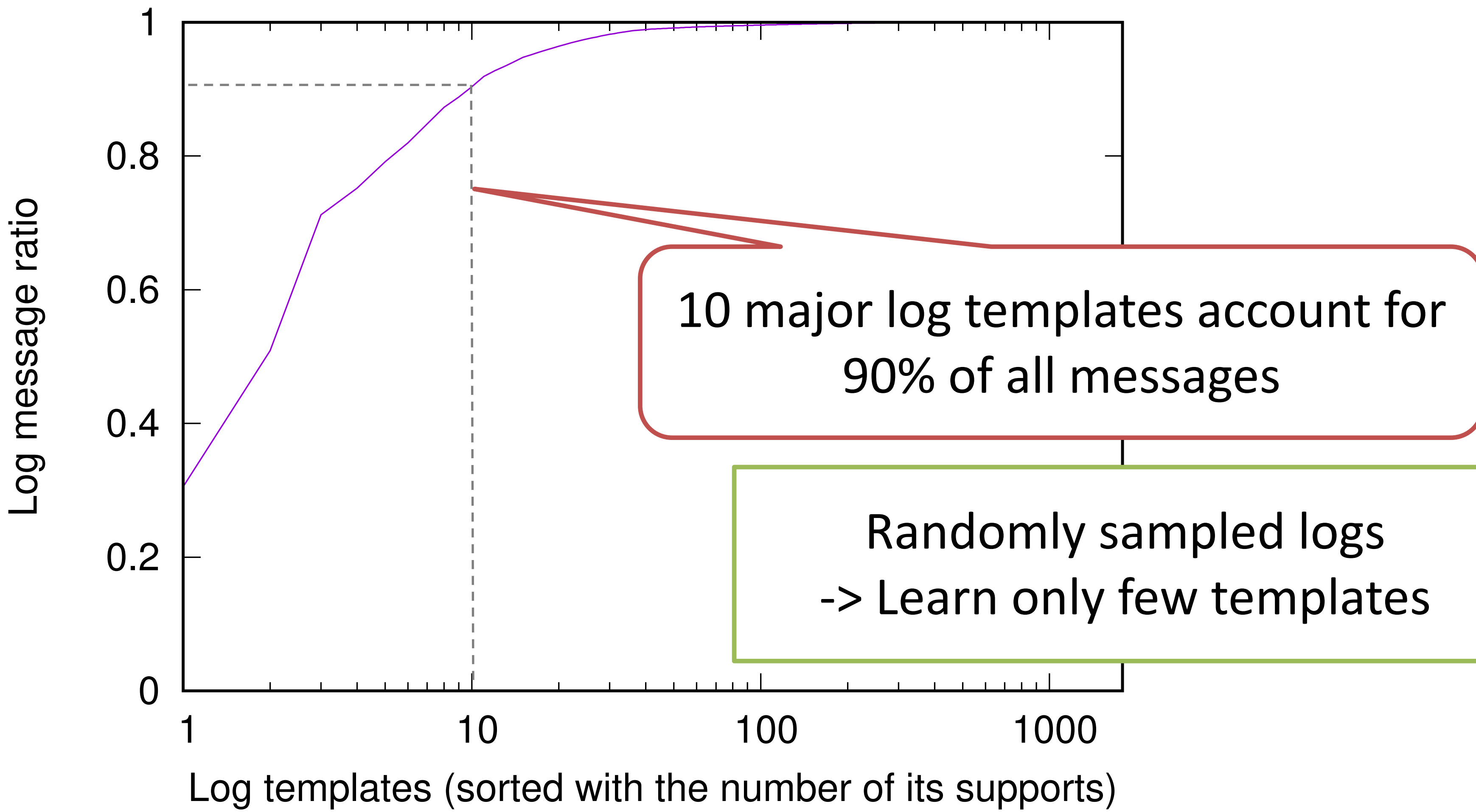
[16] S. Kobayashi, et al. “Towards an NLP-based Log Template Generation Algorithm for System Log Analysis”. In CFI, 2014
[17] J. Lafferty, et al. “Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data”. In ICML 2001, pp. 282–289, 2001.

CRF-based log template estimation



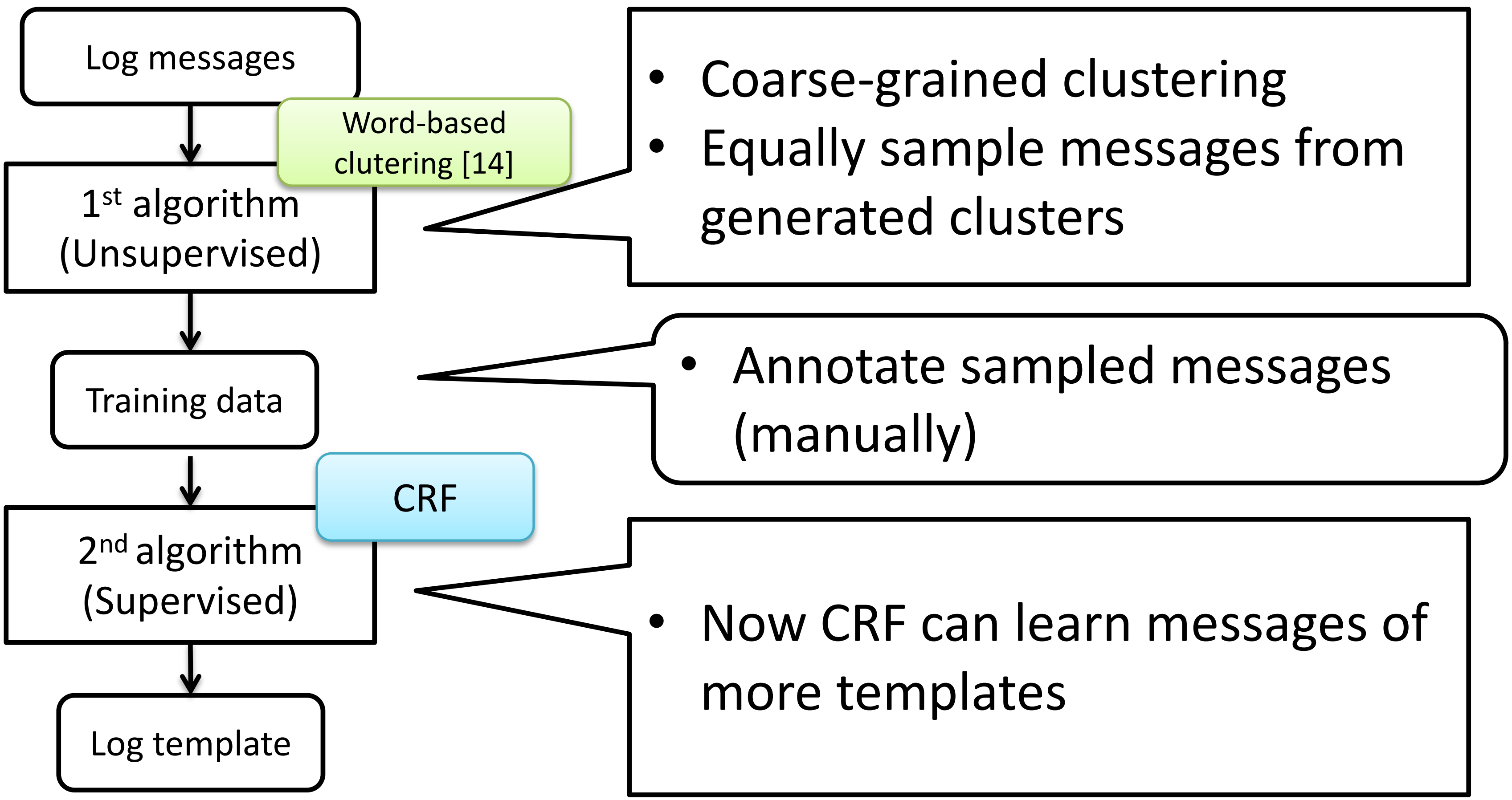
Difficulty of supervised learning in network logs

- Cumulative ratio of messages in log templates



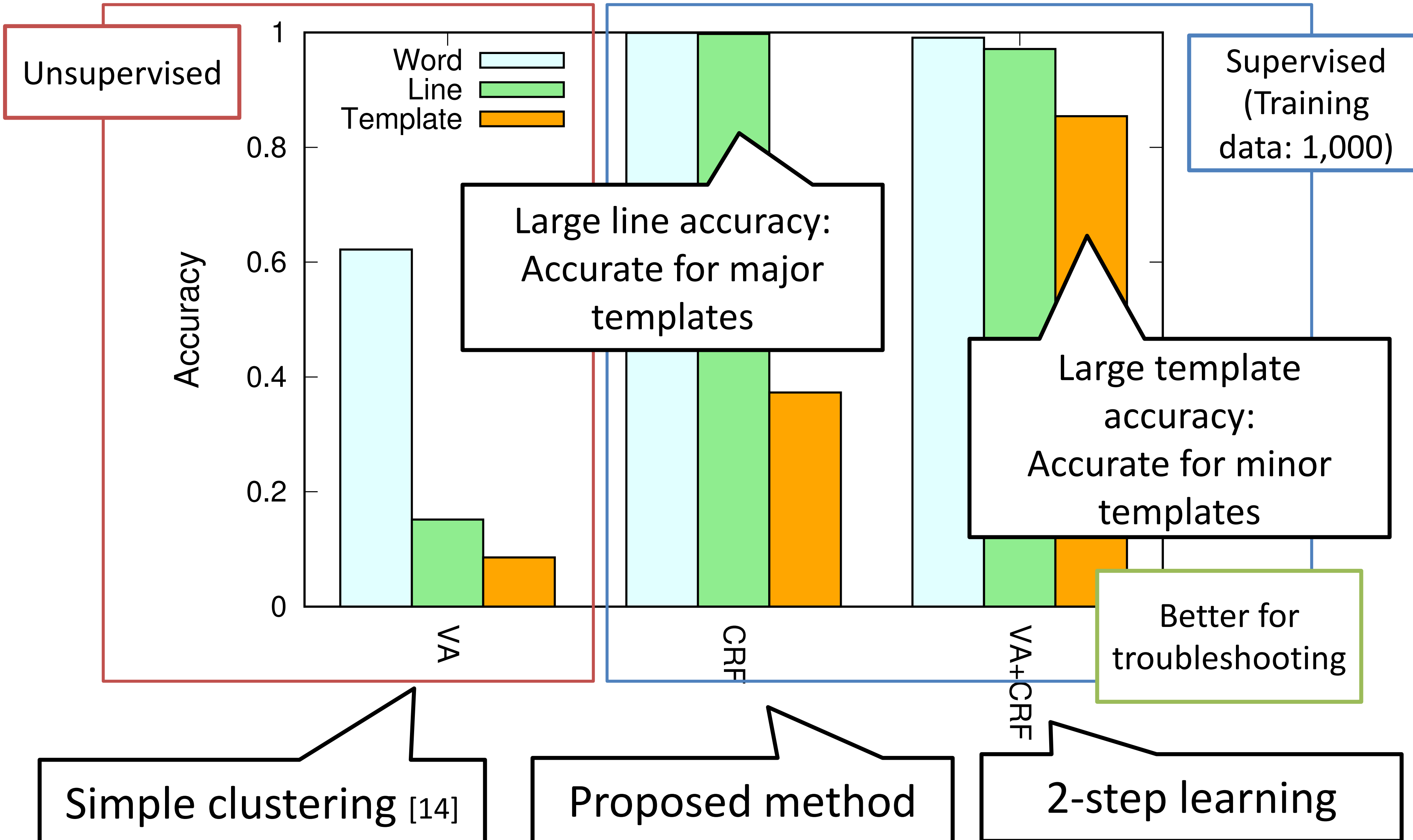
Randomly sampled logs
-> Learn only few templates

2-step log template estimation



[14] R. Vaarandi. A data clustering algorithm for mining patterns from event logs. In IEEE IPOM , pp.119-126, 2003.

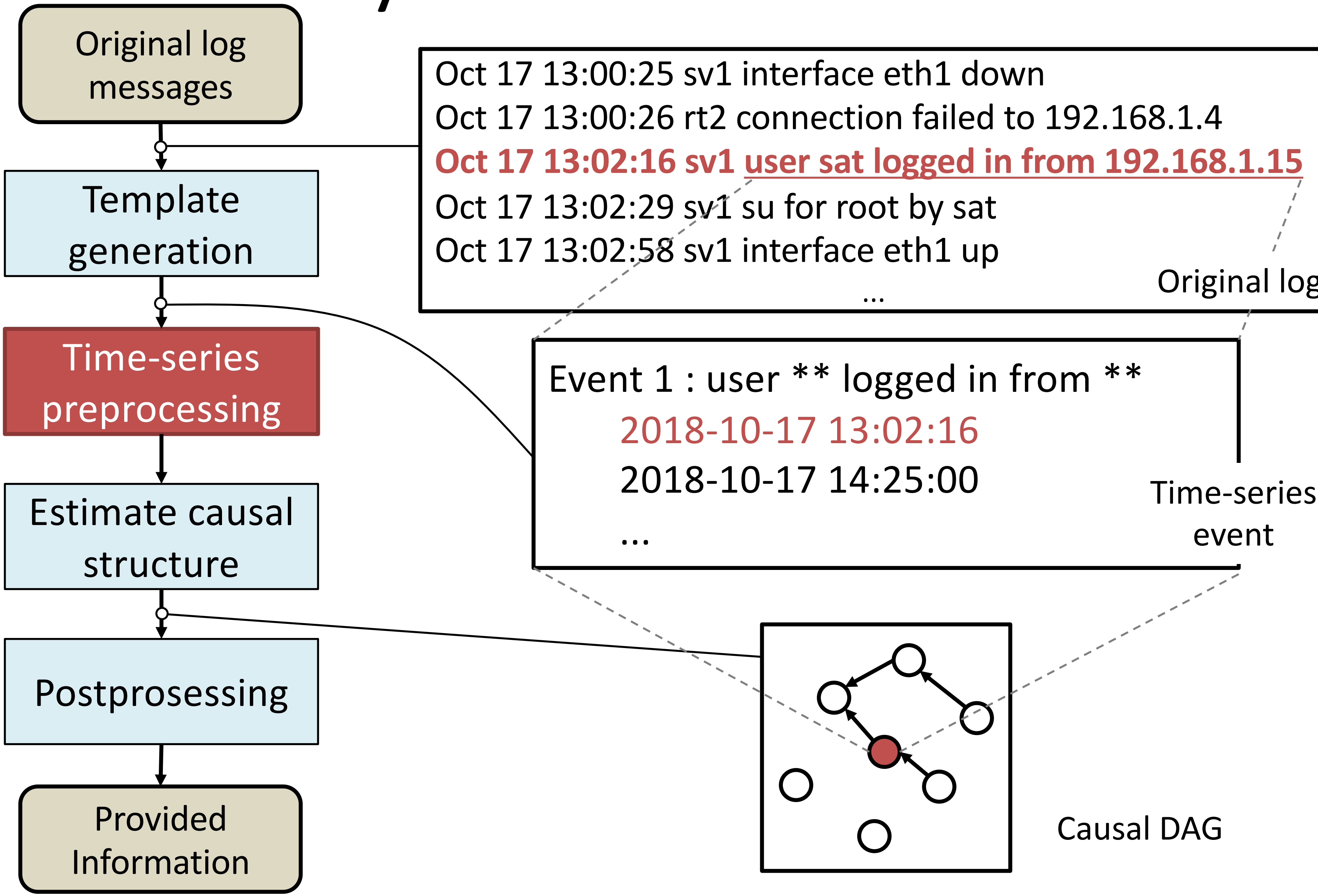
Comparison of estimation accuracy



Summary of template generation

- Classify messages based on log templates
  - Message -> Time-series event
- Supervised learning of log templates with CRF
  - Labeling words into Description and Variable
- Efficient learning with 2-step CRF
  - Learn more various log template
  - Generate accurate templates for minor log messages (important in troubleshooting)

System architecture

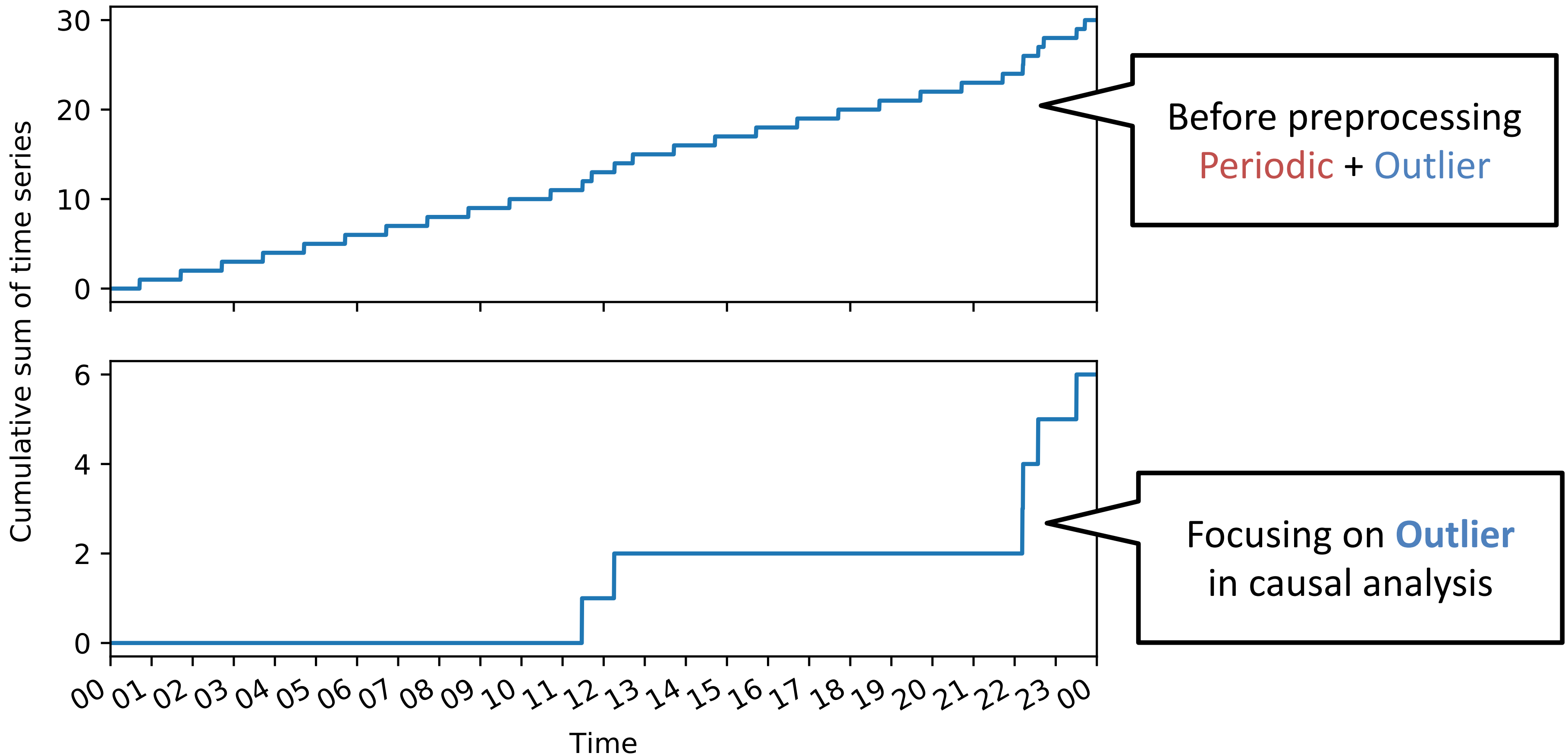


Removing redundant logs

- Redundant log events
  - Self-evident information for operators
  - Regular messages (constant or periodic)
  - Bursting messages
- Regular messages
  - Different distribution from other sparse logs
  - Cause more false positive in causal analysis

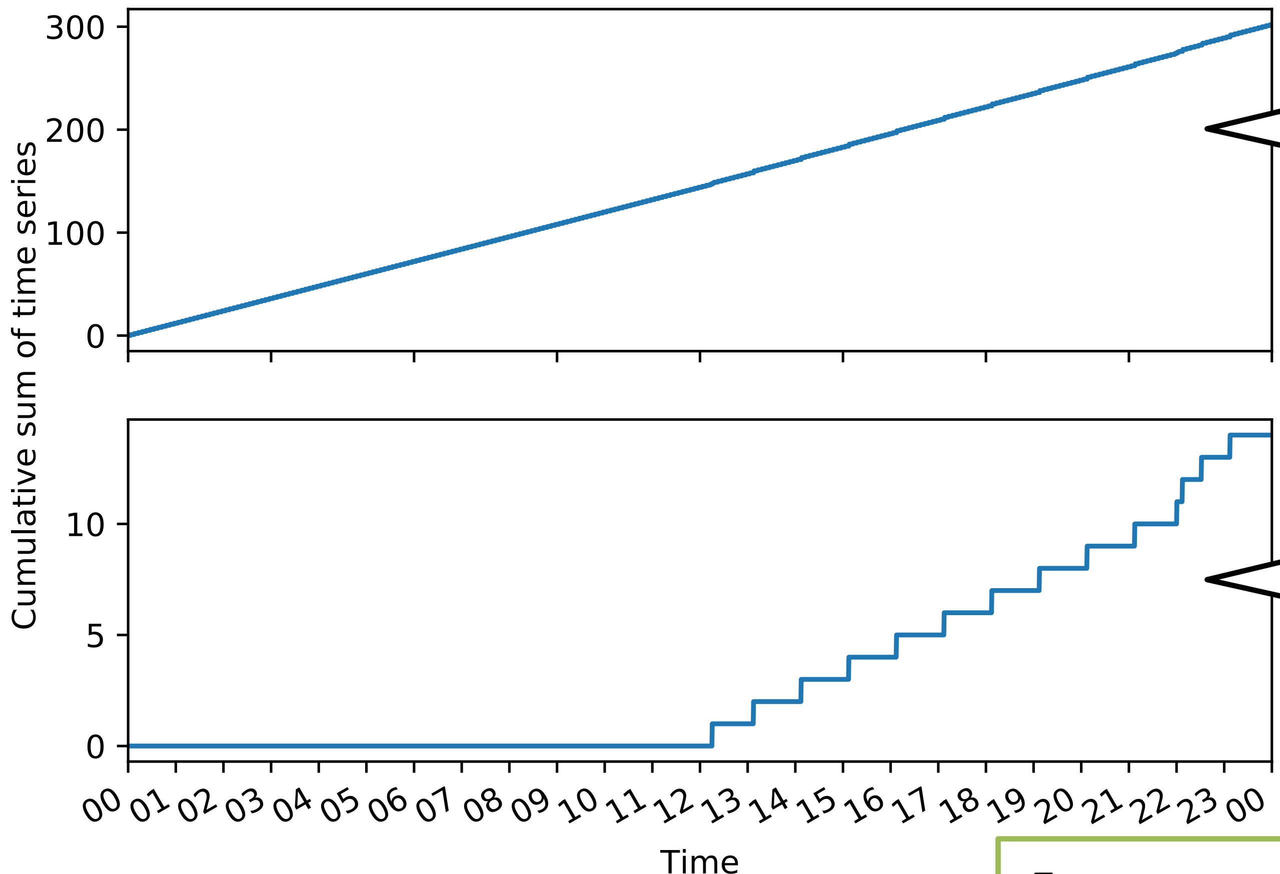
Example of redundant logs

mgd[**]: UI_AUTH_EVENT: Authenticated user '**' at permission level '**'



Example of redundant logs (2)

sshd[**]: Accepted password for ** from ** port ** **

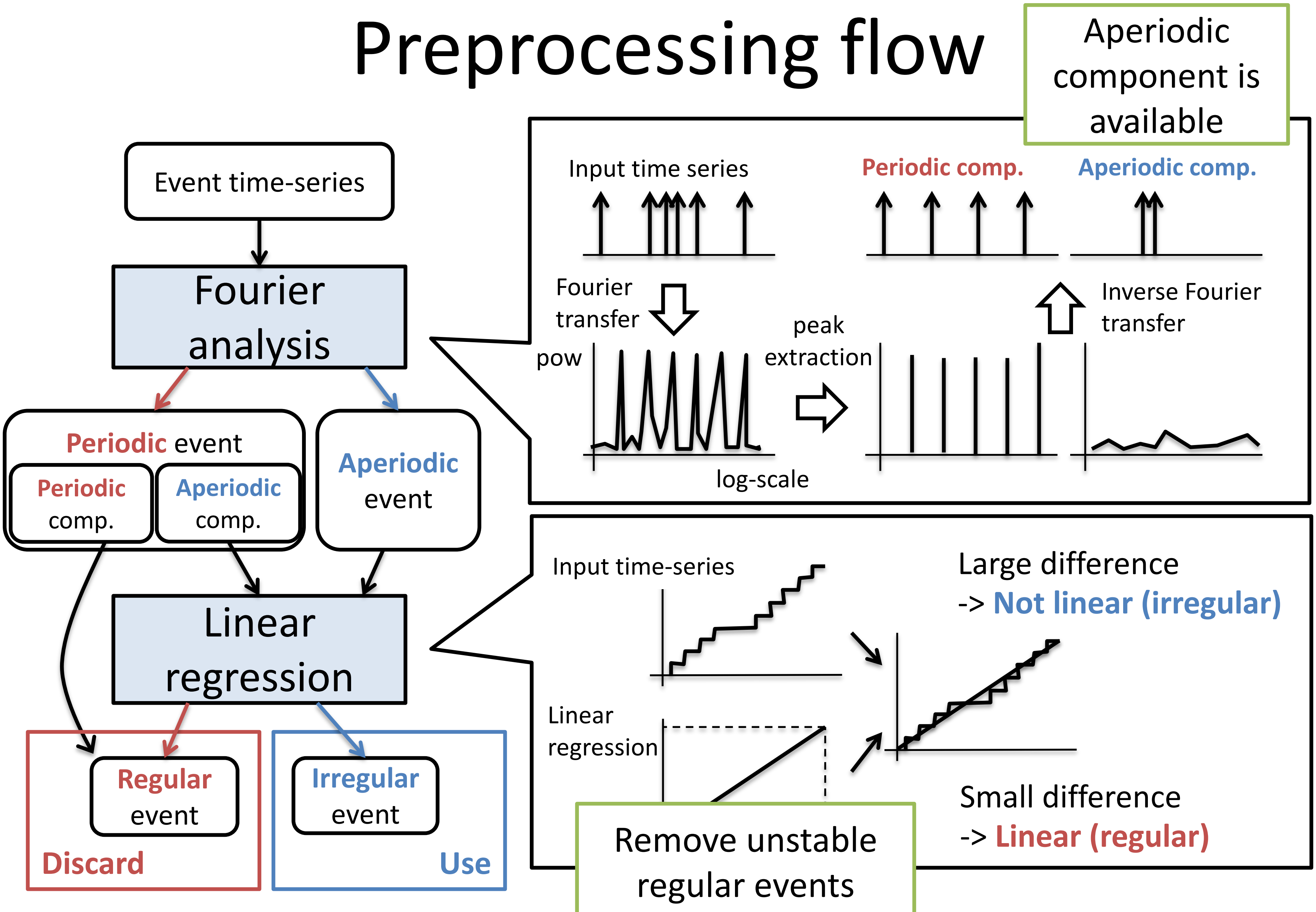


Indistinguishable in manual investigation

Appearance trend changed

Extract notable features in time-series

Preprocessing flow



Comparison of preprocessing methods

- Fourier+Linear: Proposed 2-step method
- Corr: Past approach based on self-correlation [18]

Method	Nodes	Edges	Time (sec/day)
None	293,252	74,919	**1,744**
Corr	134,020	16,210	417
Fourier+Linear	244,051	57,302	**671**

Keep major part of causal relations

Decrease 60% processing time

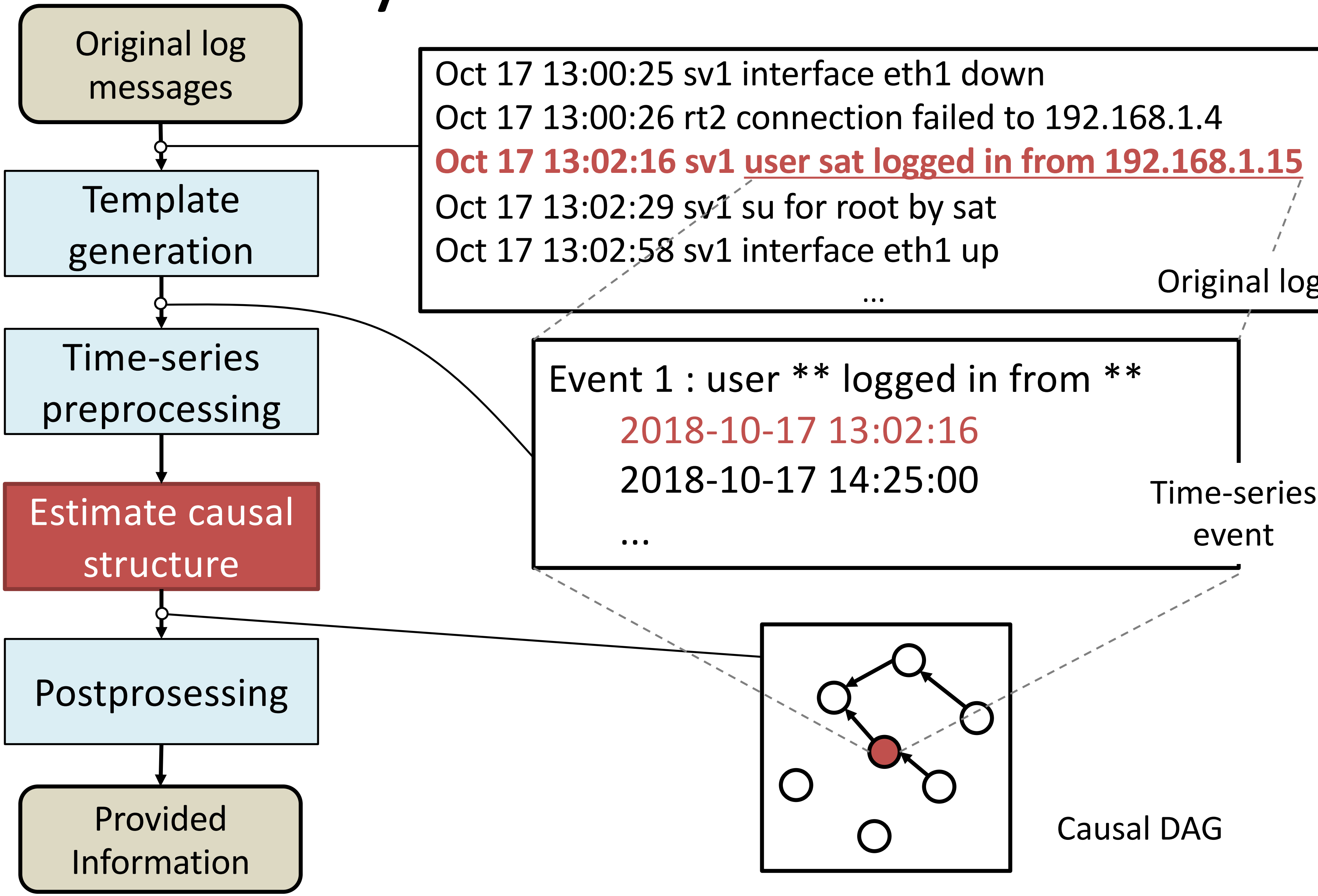
[18] S. Kobayashi et al. Mining causes of network events in log data with causal inference. In IEEE IM, pp.45-53, 2017.

Summary of preprocessing

- Preprocessing to decrease false positives
  - Remove periodic events
    - Fourier analysis
    - Leave aperiodic components in periodic events
  - Remove regular events
    - Linear regression
    - Available in regular events with unstable interval
- Leave useful information that is removed in past approach [18]
- Decrease 60% processing time

[18] S. Kobayashi et al. Mining causes of network events in log data with causal inference. In IEEE IM, pp.45-53, 2017.

System architecture



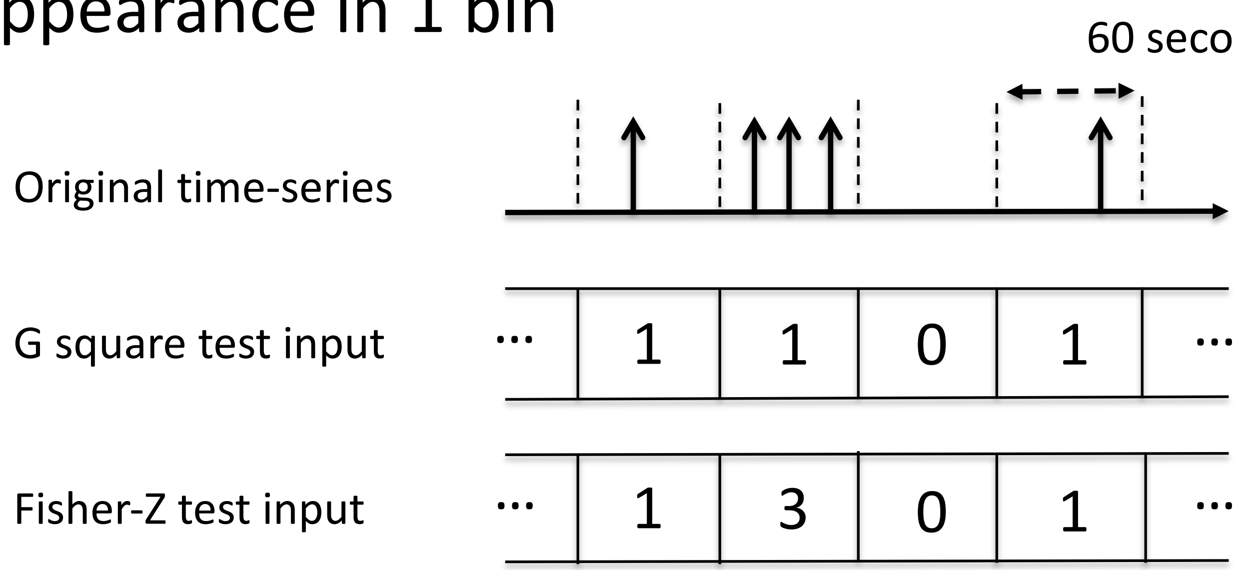
Conditional independence test

- Conditional independence test (CI test)
  - Used repeatedly in PC algorithm
- CI test from time-series data
  - G square test [10]
    - Based on information theory
    - Conditional cross entropy
  - Fisher-Z test [10]
    - Based on statistics
    - Pearson-based, partial, population correlation

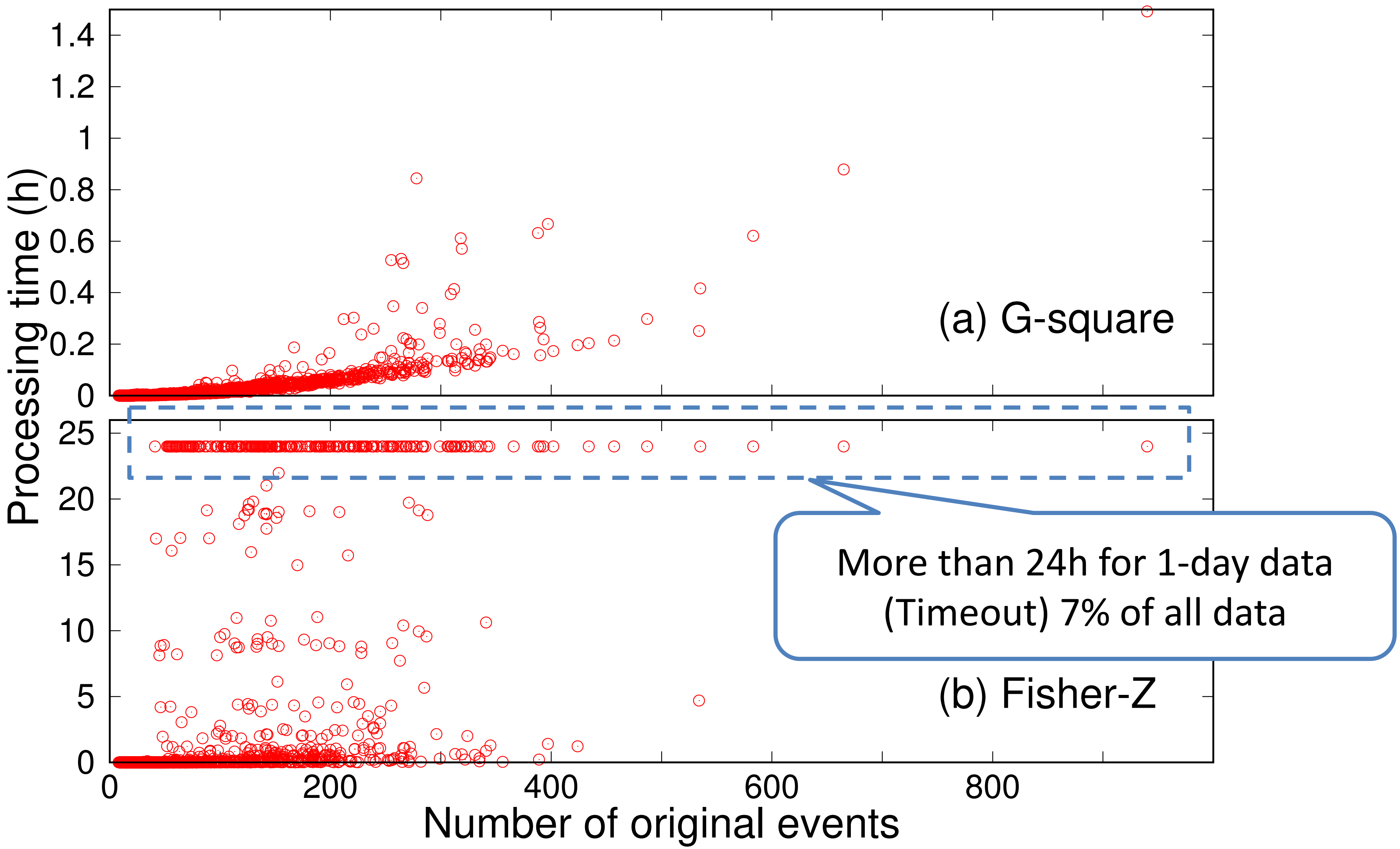
[10] RE. Neapolitan. "Learning Bayesian Networks." Prentice Hall Upper Saddle River, 2004.

CI test in network log time-series

- G square test
  - Input: binary or multi-level values
  - Ignoring multiple appearance in 1 bin
- Fisher-Z test
  - Input: integer
  - Assume Gaussian distribution data

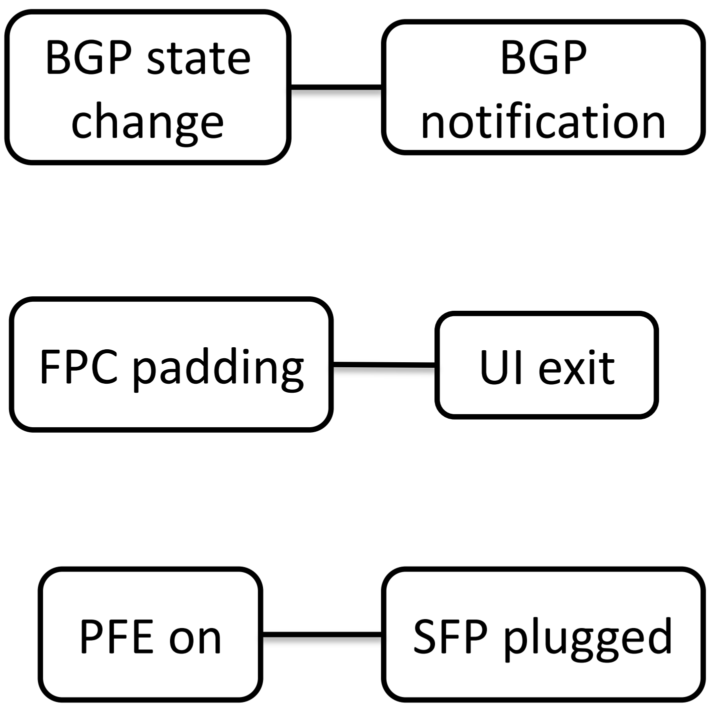


Processing time comparison

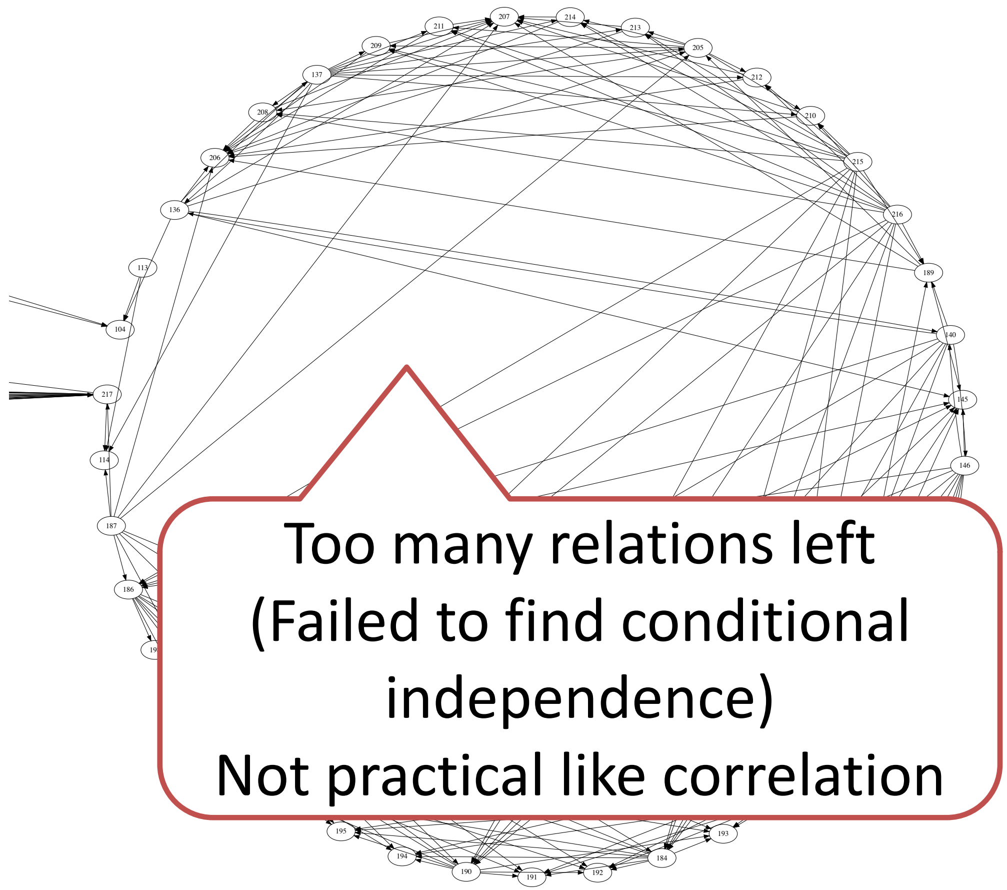


Example of estimated DAG

- G square test



- Fisher-Z test (partial)

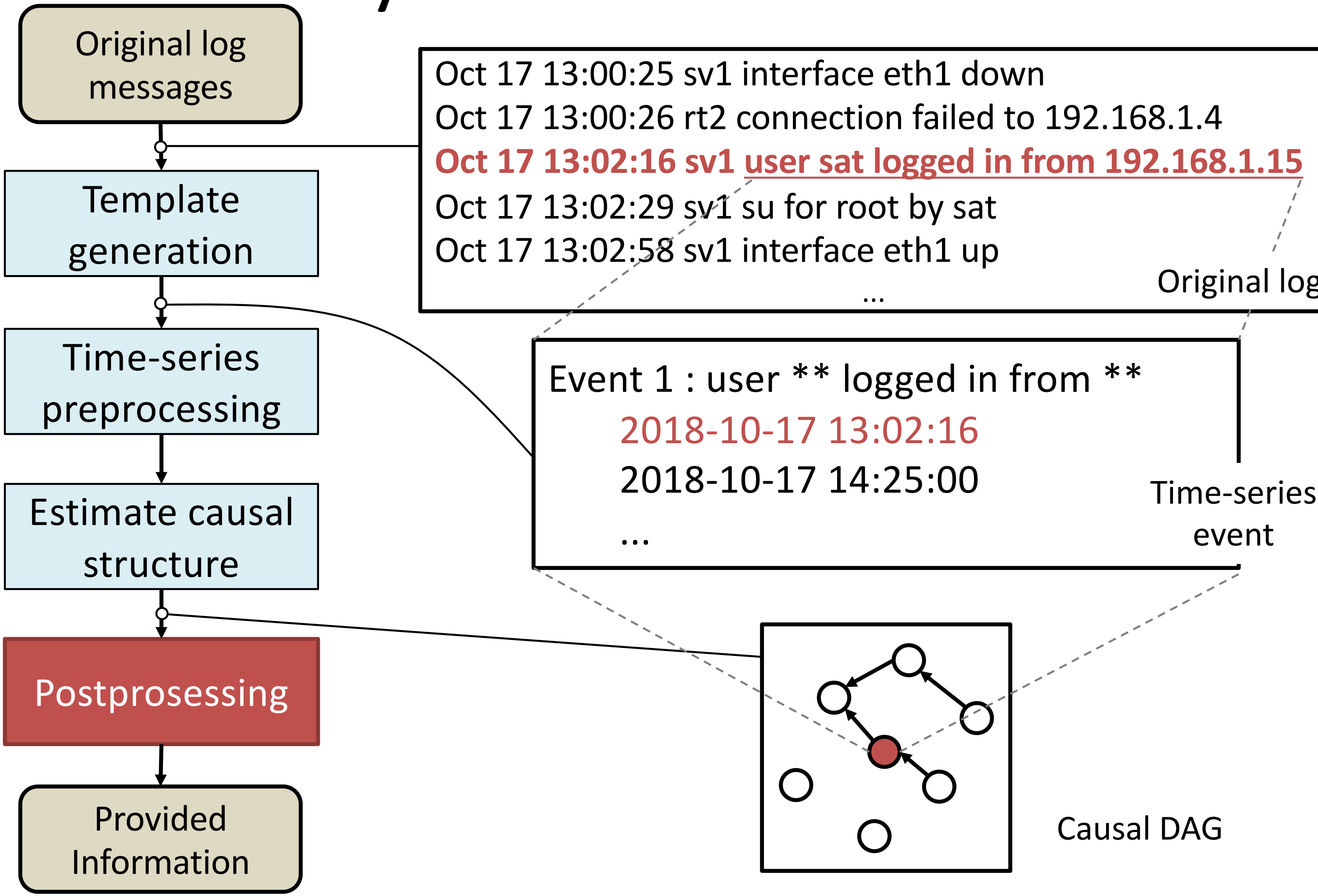


Too many relations left
(Failed to find conditional independence)
Not practical like correlation

Summary for CI tests

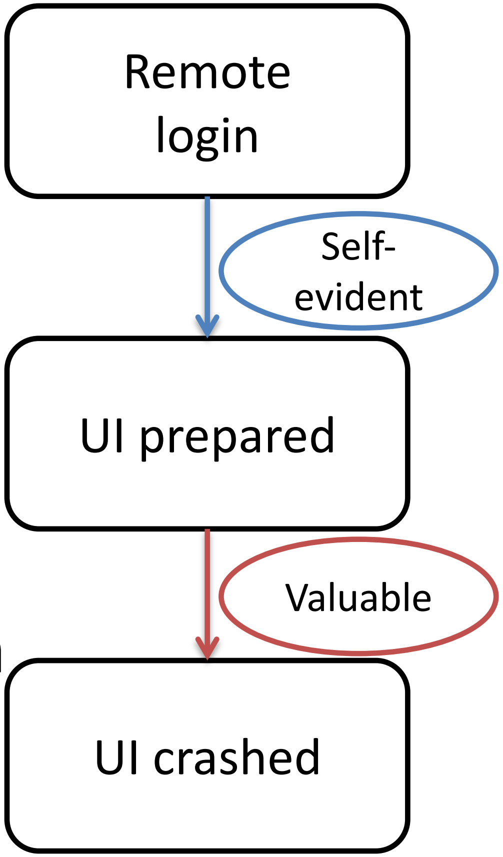
- Conditional independence tests
  - G square test
  - Fisher-Z test
- G square test is more useful in network logs
  - Sparse appearance of error-related logs
  - Processing time, quality of edges (i.e., information concreteness)

System architecture



Post-processing

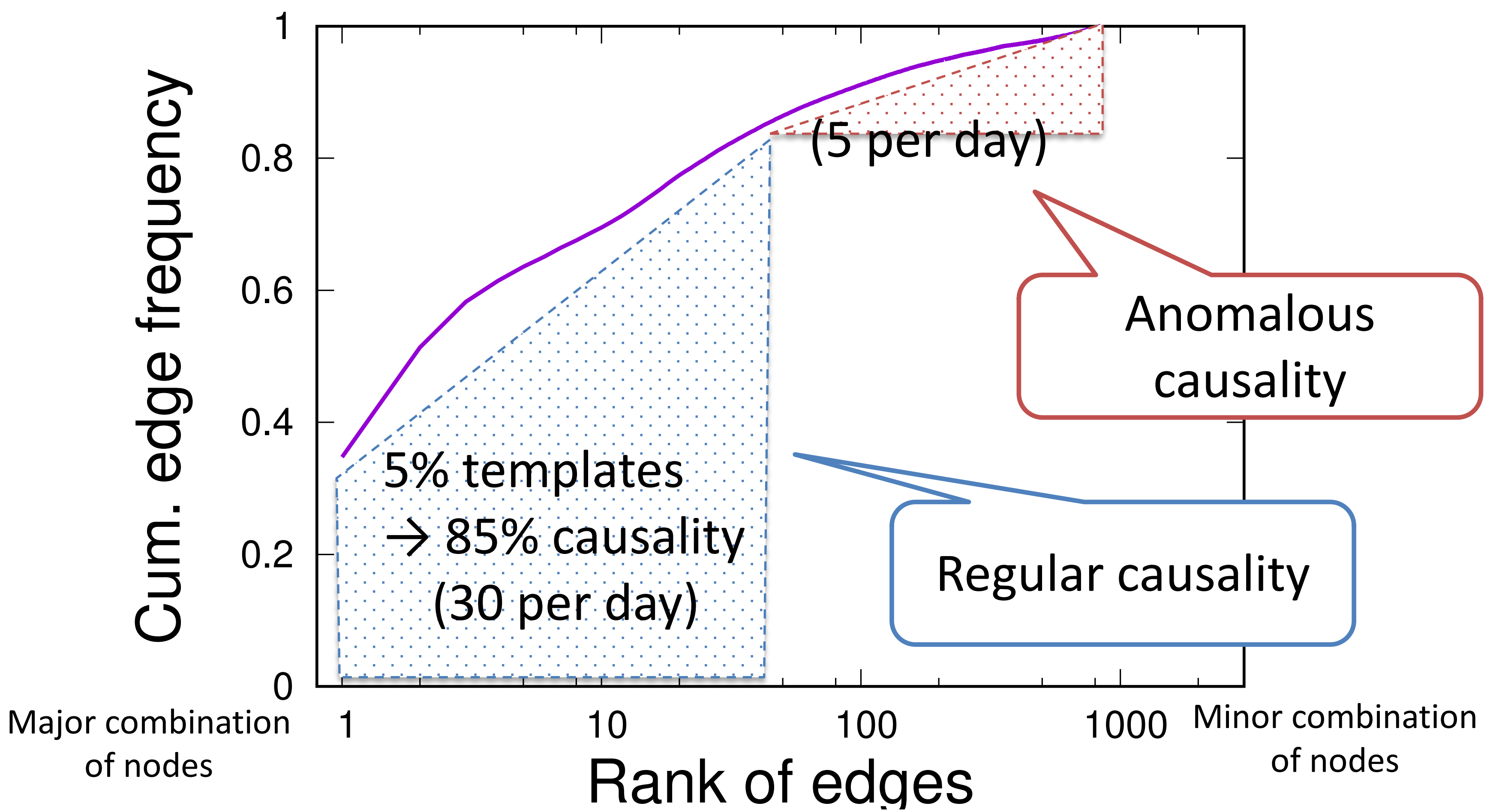
- Found causal relations
  - Self-evident causality
    - Regular behavior
  - Valuable causality
    - Irregular or anomalous behavior
- Classify edges with regularity
  - ➢Give priority to irregular information



Contribution of exploratory analysis
(Regularity not available in existing works)

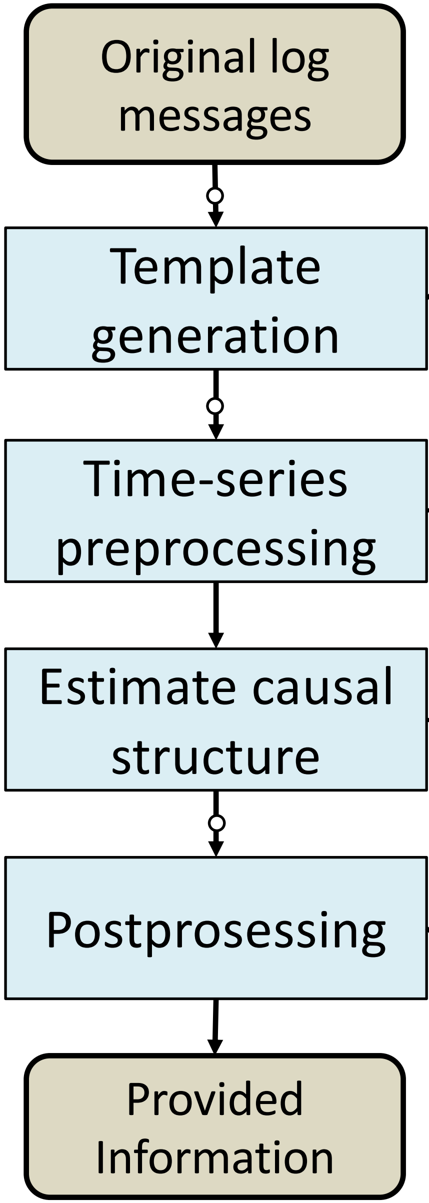
Effect of post-processing

- Cumulative ratio of causal edges



System ar

Correspondence to challenges



String with timestamp, not time-series
-> **CRF-based log template estimation**

Redundant (not important) messages
-> **Preprocessing with Fourier analysis and linear regression**

Sparse appearance of error messages
-> **Conditional independence with G square test**

Self-evident relations
-> **Causality classification based on edge regularity**

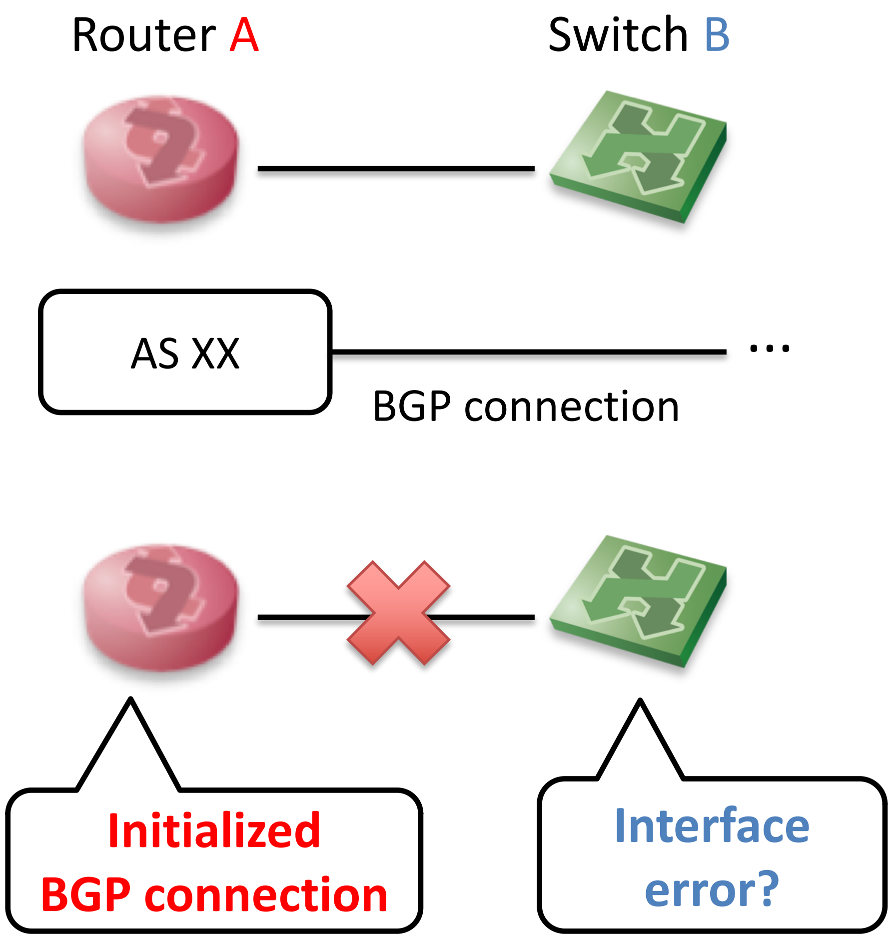
Outline

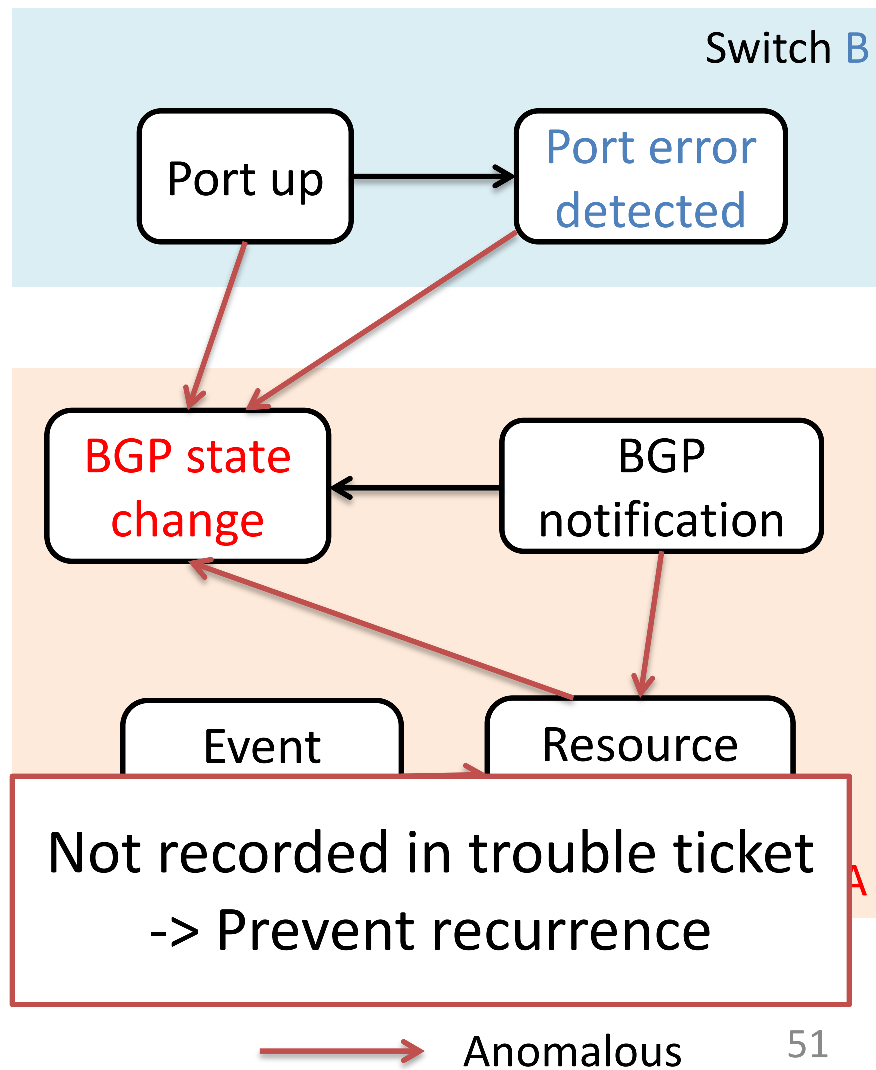
- Background
- Causal analysis
- System architecture
  - 4 underlying methods
- Evaluation
- Conclusion

Evaluation

- 15 months log data for causal analysis
  - 3.5 million lines (1789 Log templates, 131 hosts)
  - ➢ Causal edges: 57,302 (sum for every 1-day data)
- Investigation of found causal edges
  - Case study
  - Comparison with trouble ticket

Example of detected DAGs





Comparison with trouble tickets

- Detectability of causality related to tickets

Tickets with related causal edges

Tickets with related log messages

One-off events
-> difficult to detect

Event type	Associated tickets	All tickets	Detection rate
Routing-EGP	91	106	86%
System	11	36	31%
VPN	19	19	100%
Interface	10	15	67%
Monitor	7	10	70%
Network	1	1	100%
Management	0	1	0%
Total	139	188	74%

Manually labeled event type

Provide valuable information in major parts of troubles

Discussion about detected causality

- Detect valuable causality
  - Causality corresponding to trouble tickets
    - Practicability
  - Causality that is not recorded in trouble tickets
    - Useful to prevent recurrence
- Decrease false positive causality
  - Redundant edges <- preprocessing
  - Decrease false positive <- preprocessing / G2 test
- Self-evident causality can be filtered

Conclusion

- Causation mining in network logs
  - Estimate causal DAG with PC algorithm
  - 4 devices for challenges of log causal analysis
- Evaluation with large-scale network logs
  - Detect useful information for troubleshooting
  - Prevent excessive extraction of information for practicability by decreasing false positives and unimportant relations