Comparing rule mining approaches for computer security

Martin Kopp

Cognitive Intelligence, Cisco Systems

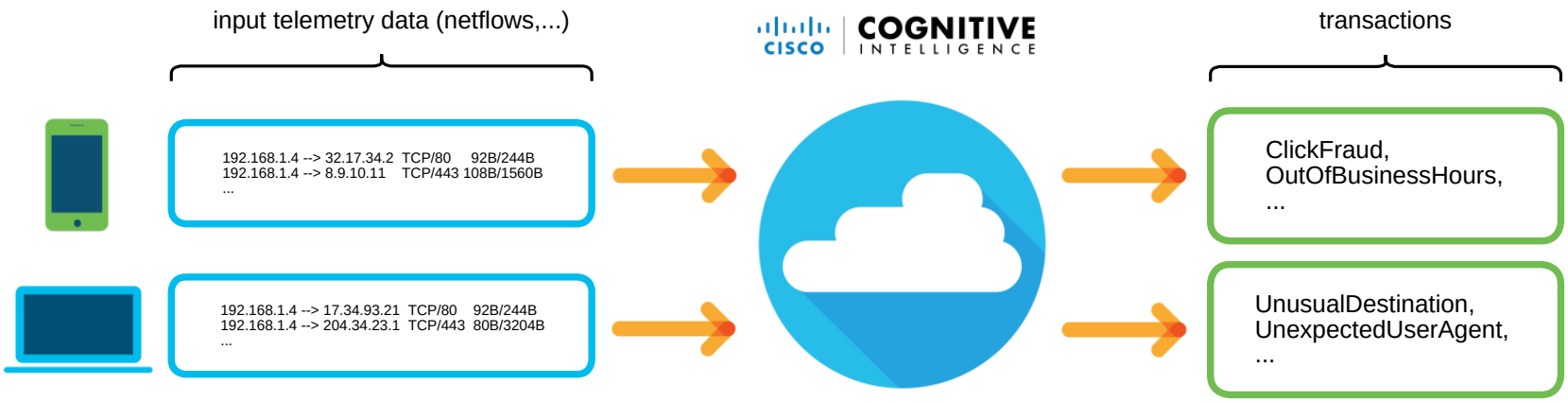
October 28, 2020

Outline

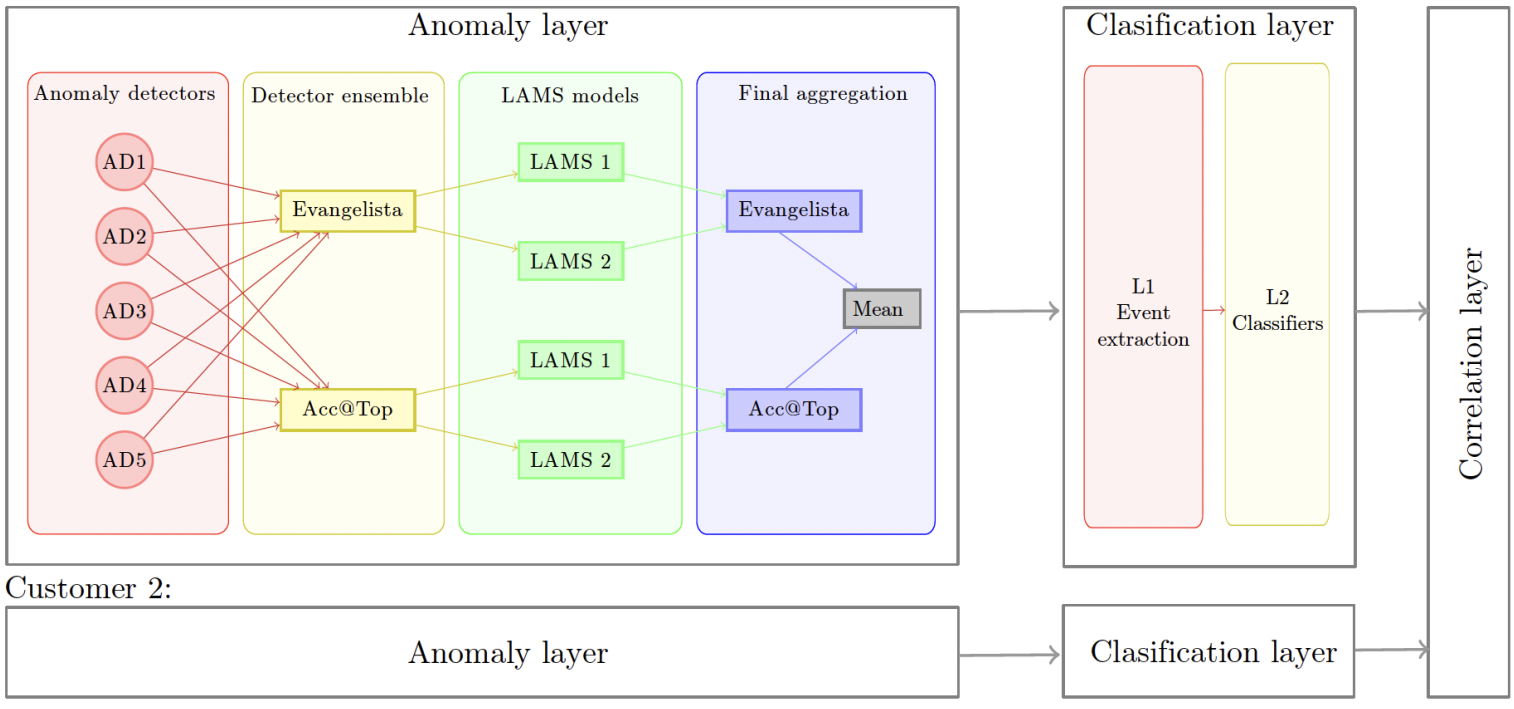
1. Motivation
   - Domain introduction
   - Problem overview
2. Algorithms
   - Random Forests
   - Explainer
   - Frequent Item Set Mining
   - Logical Item Set Mining
3. Comparison
   - Performance
   - Explanations
   - Take aways

Domain introduction

Domain introduction



Domain introduction



Grill, Martin "Combining network anomaly detectors" (2016)

Problem statement

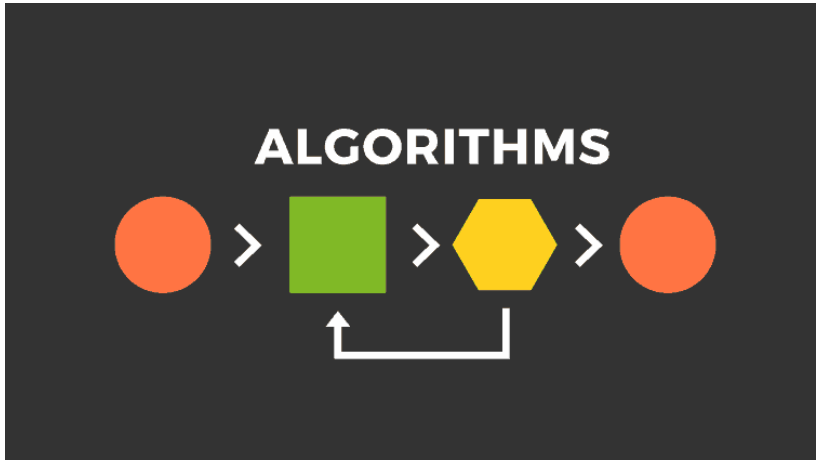
- 50 billion flows daily
- imbalance problem 1:100 – 1:1000
- multi-class (200+)
- analysis is expensive
- speed is critical

Problem statement

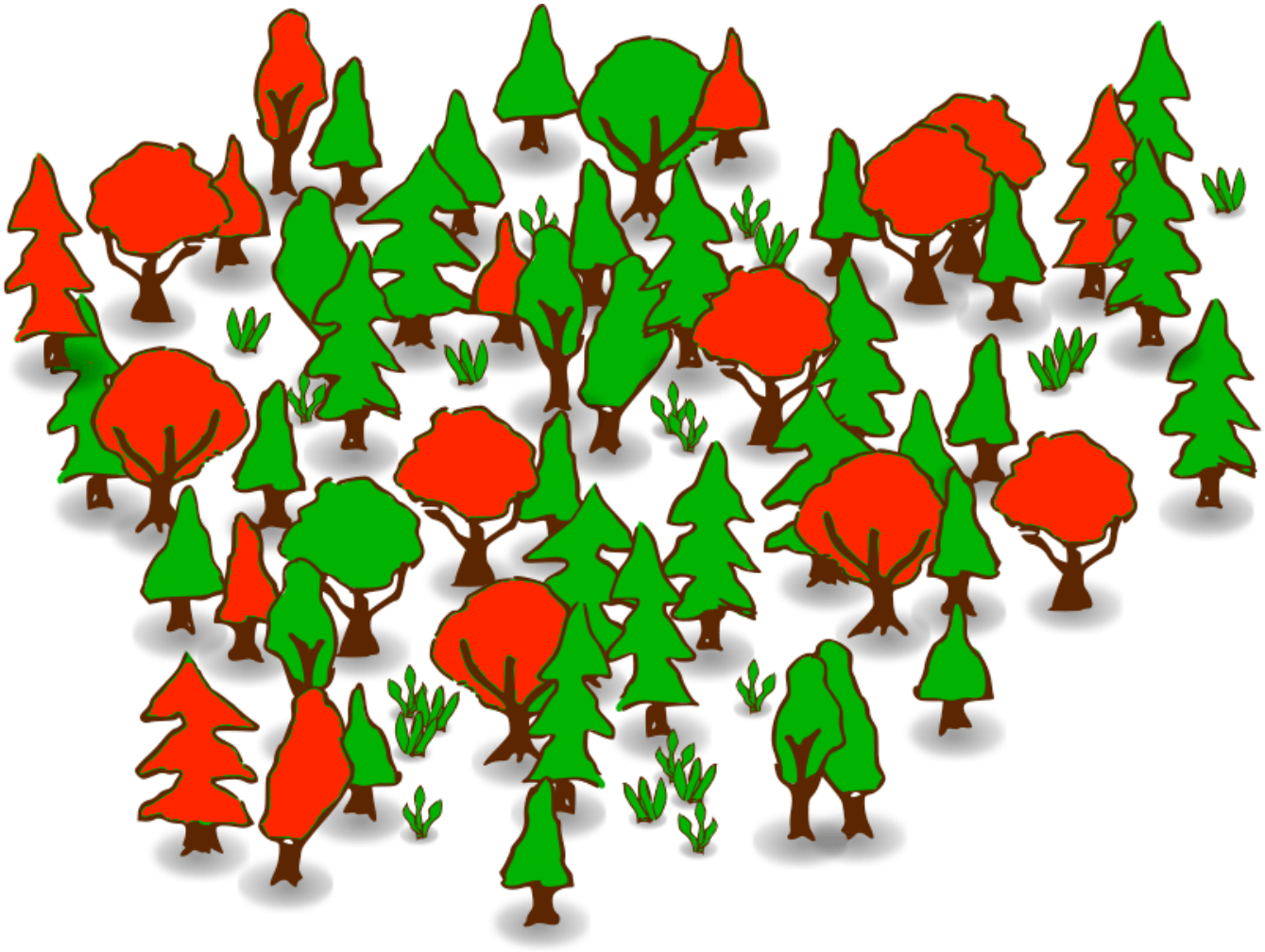
- flow $\rightarrow$ event $\rightarrow$ transactions
- 3 months of data
- 50M transactions
- 12 high level malicious classes
- rule mining

Problem statement

- flow $\rightarrow$ event $\rightarrow$ transactions
- 3 months of data
- 50M transactions
- 12 high level malicious classes
- rule mining

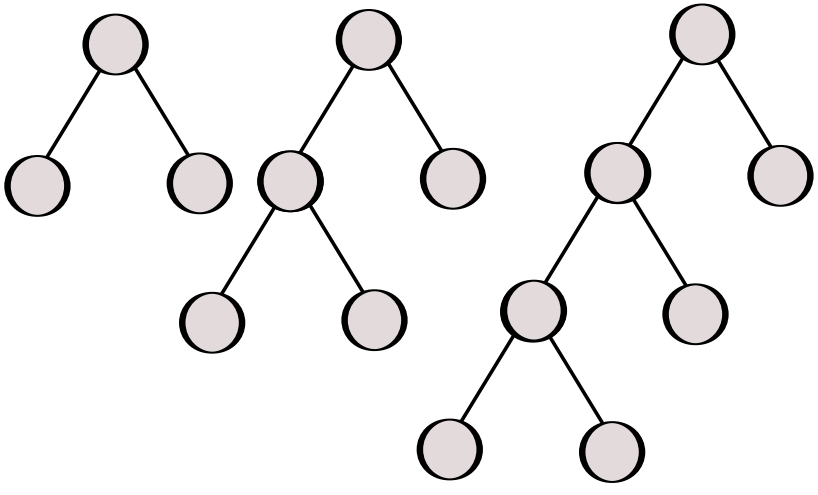

ALGORITHMS

Random Forests



Breiman, Leo "Random forests." (2001)
Ho, Tin Kam "Random decision forests" (1995)

Explainer



Kopp, Martin, et al. "Anomaly explanation with random forests." (2020)

Explainer

Summary of the Explainer algorithm for a minimal explanation

Input:

$data$ - input dataset; $size$ training set size; x^a anomalous sample; n_T number of trees to be trained.

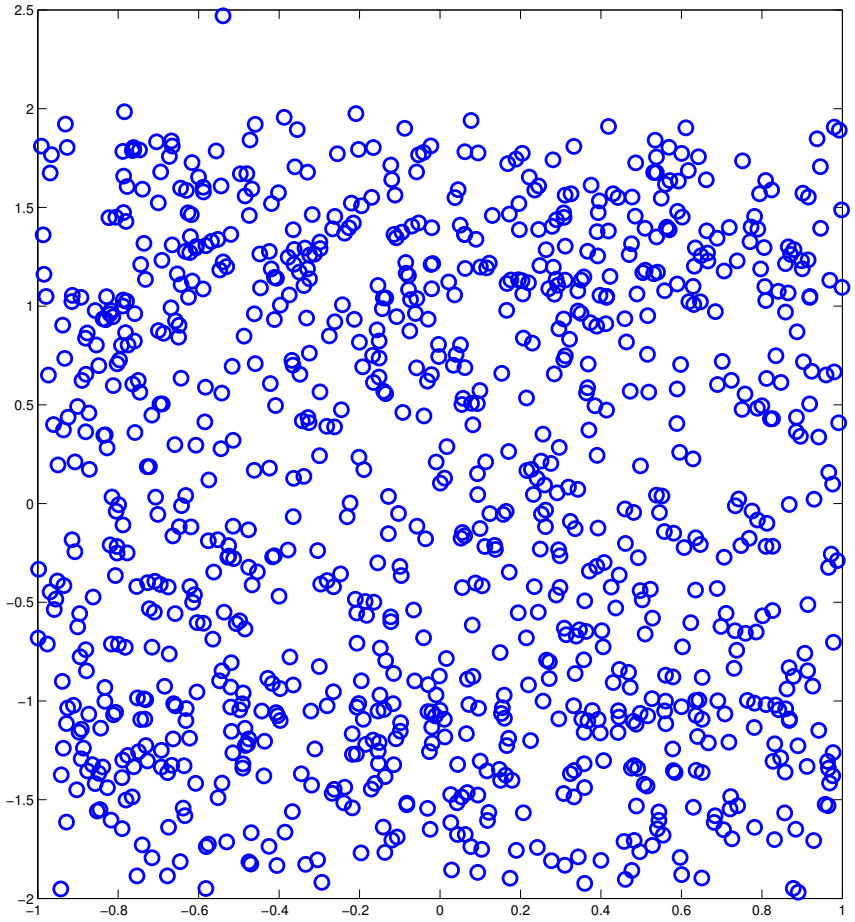
Output:

$rules$ - rules explaining x^a

$Forest \leftarrow \emptyset$
for $i \leftarrow 1 \ldots n_T$ **do**
    $\mathcal{T} \leftarrow \text{createTrainingSet}(data, size, x^a)$
    $t \leftarrow \text{trainTree}(\mathcal{T})$
    $Forest \leftarrow Forest + t$
end for
$rules \leftarrow \text{extractRules}(Forest)$

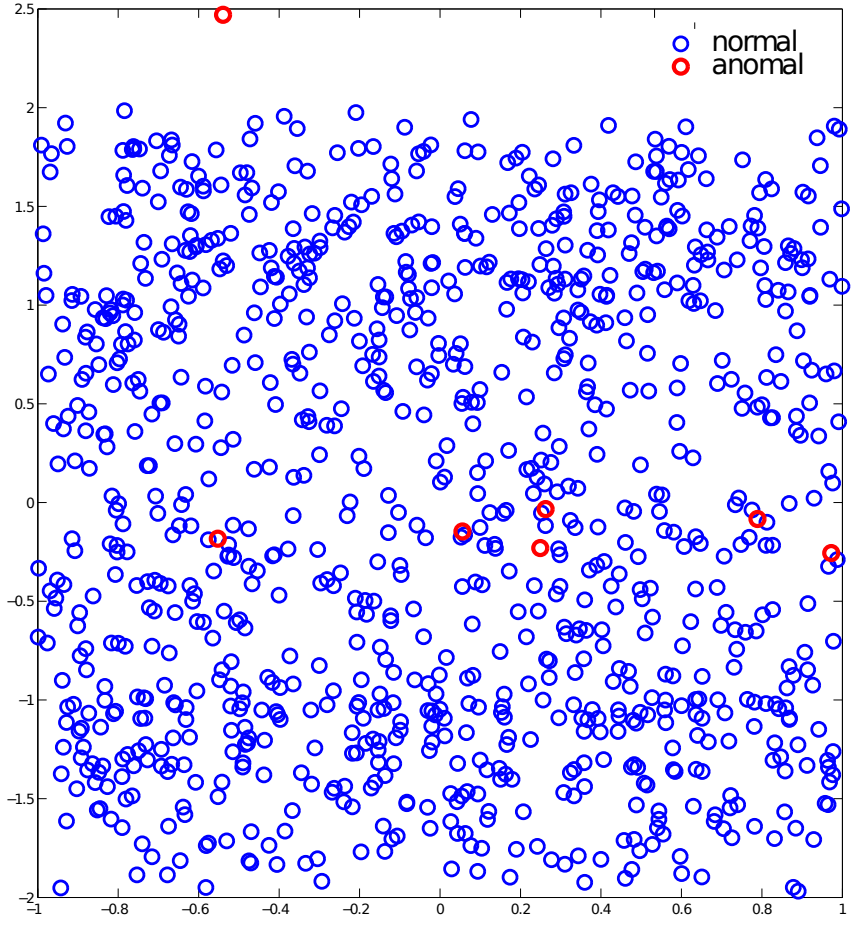
Explainer

Input: *data*



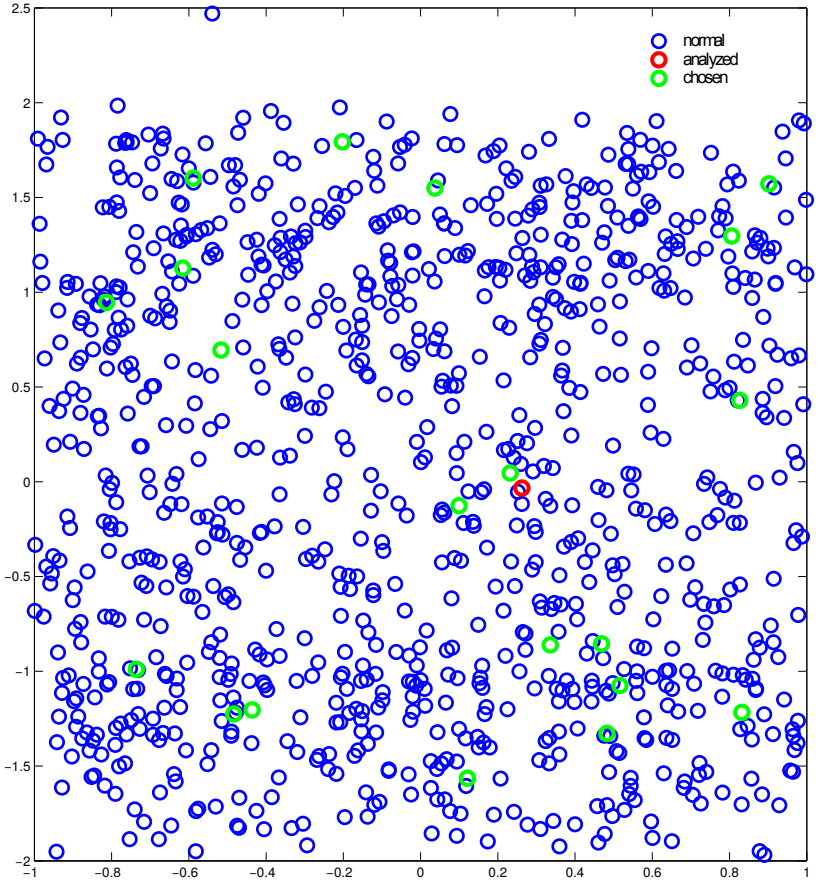
Explainer

$$labels \leftarrow anomalyDetector(data)$$

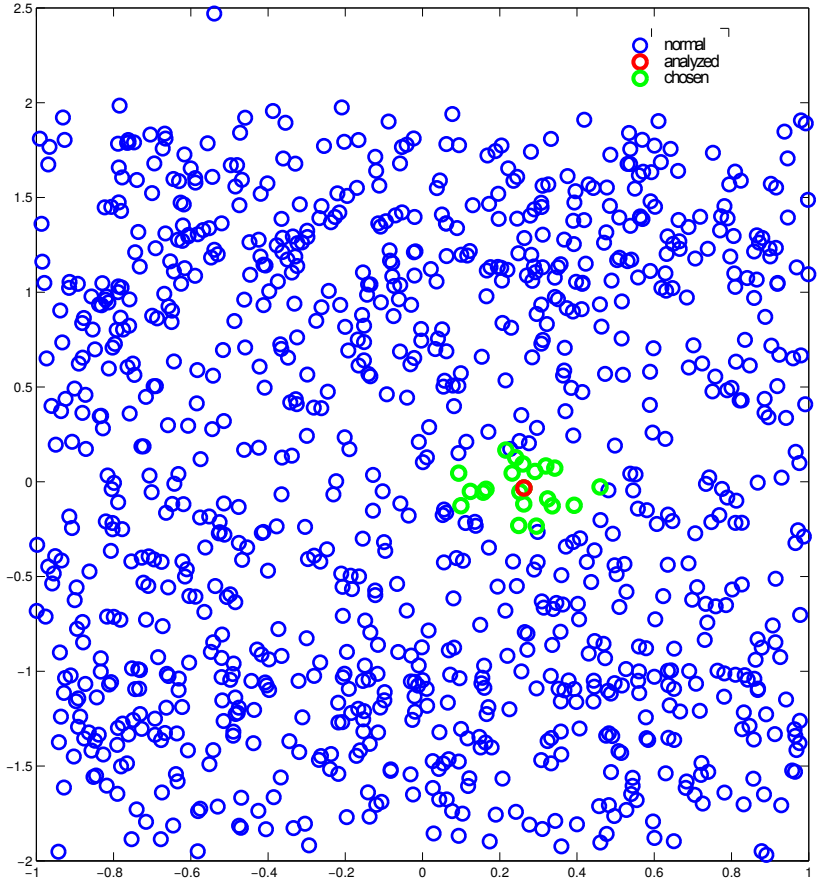


Explainer

$$\mathcal{T} \leftarrow createTrainingSet(data, size, x^a)$$



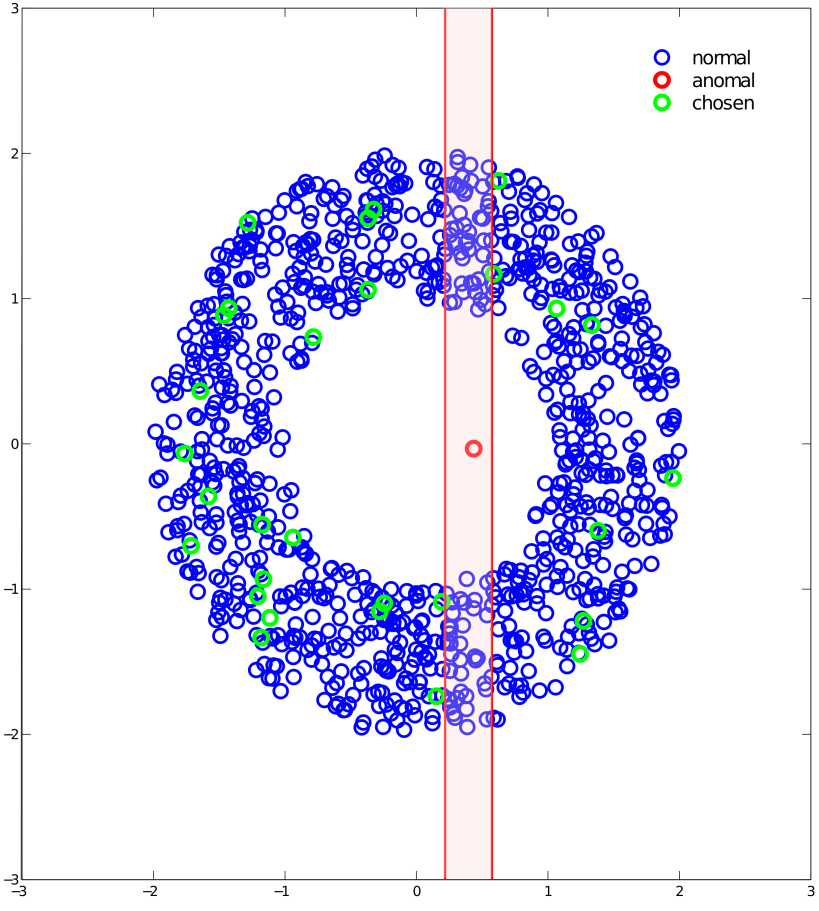
(a) random selection

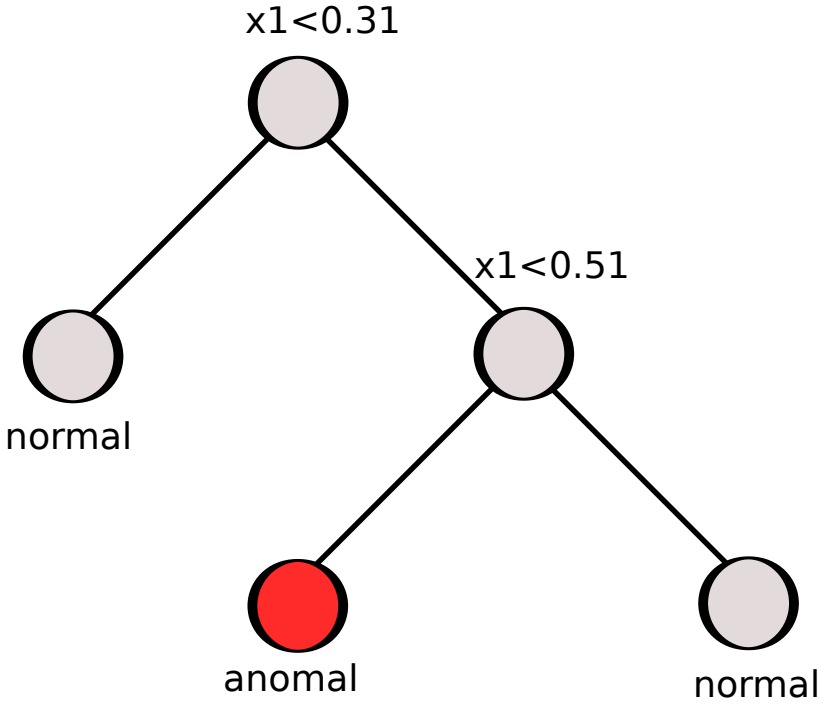


(b) k-nn selection

Explainer

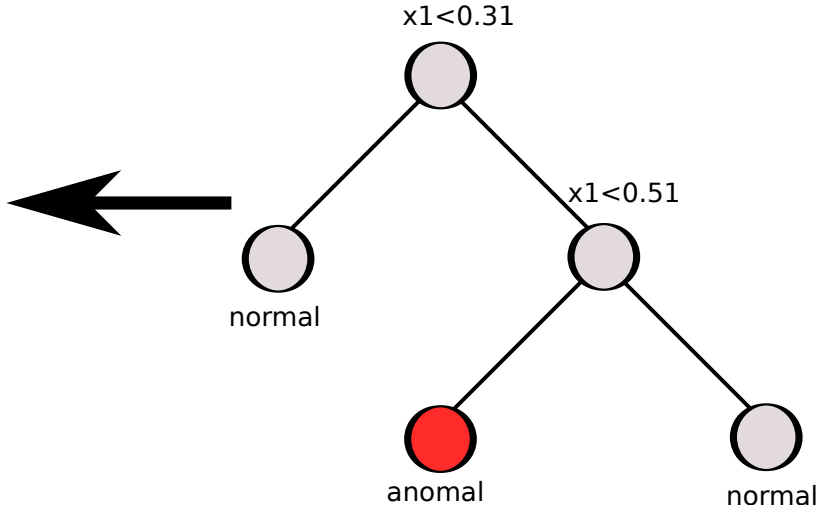
$$t \leftarrow trainTree(\mathcal{T})$$





Explainer

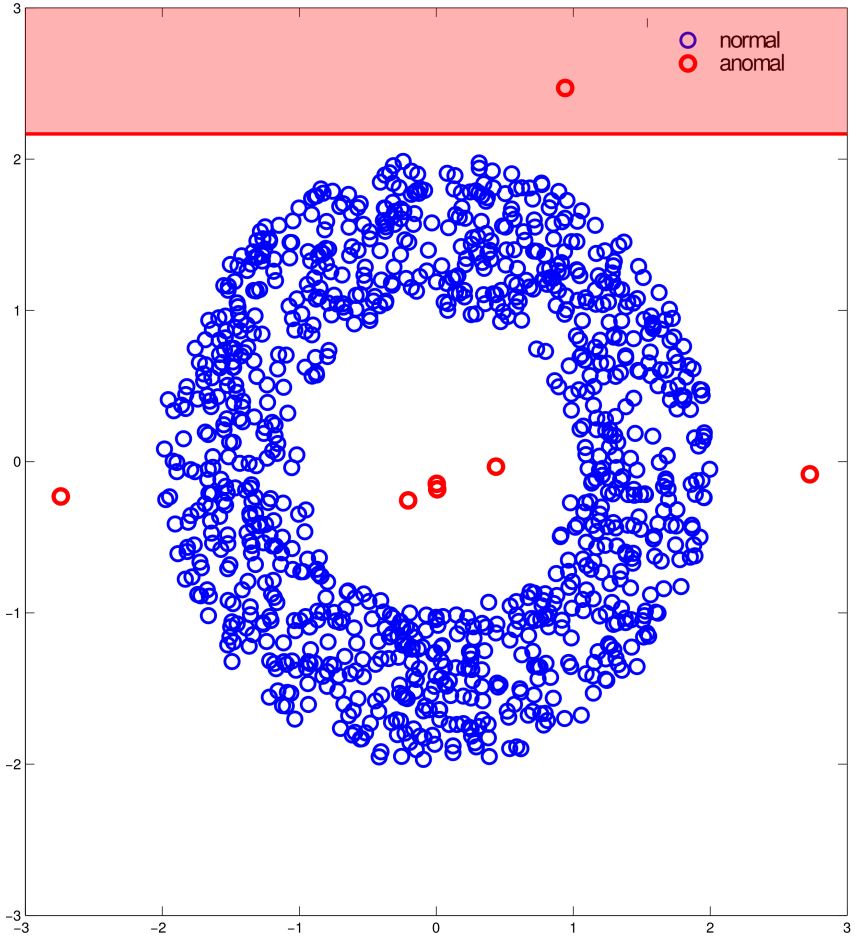
$$Forest \leftarrow t$$



Explainer

$$extractRules(Forest)$$

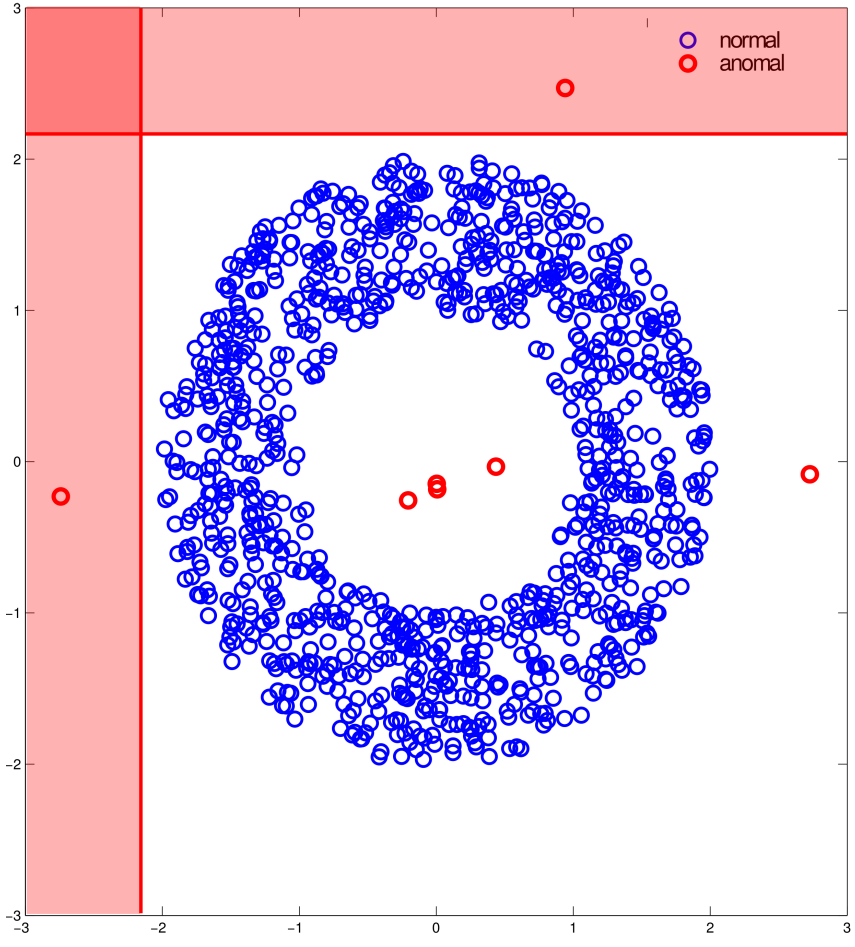
$$C = x_2 > 2.2$$



Explainer

$$extractRules(Forest)$$

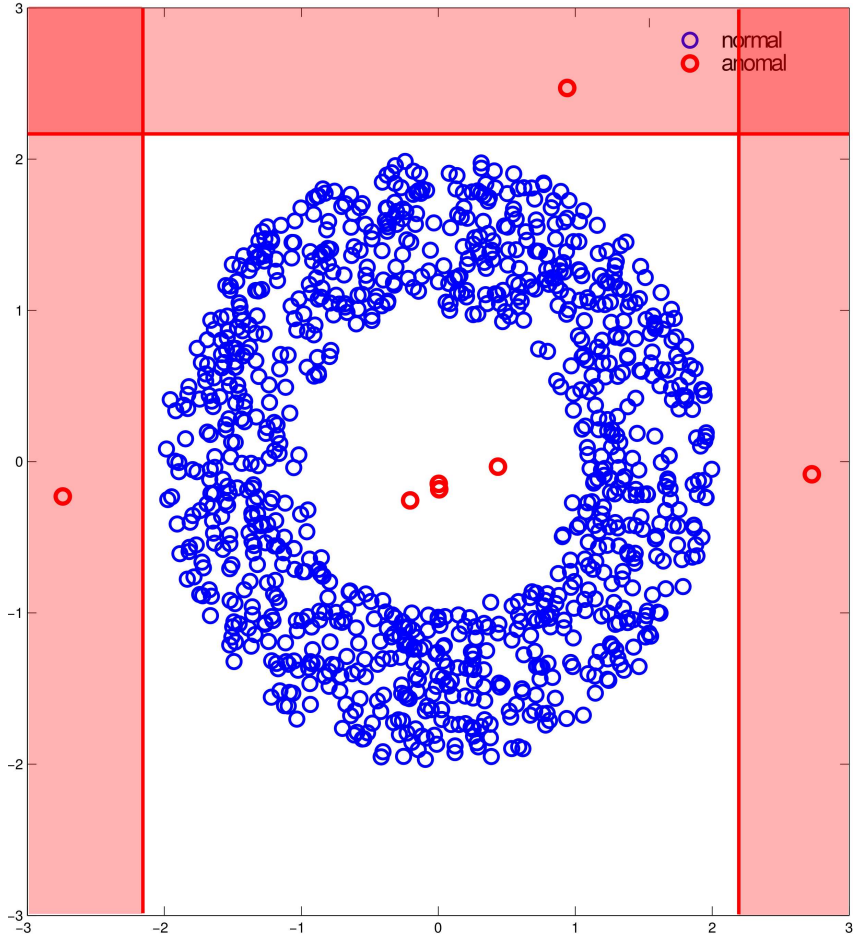
$$C = (x_2 > 2.2) \wedge (x_1 < -2.4)$$



Explainer

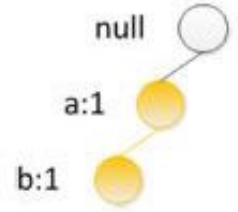
$$extractRules(Forest)$$

$$C = (x_2 > 2.2) \wedge (x_1 < -2.4) \wedge (x_1 > 2.3) \wedge \ldots$$

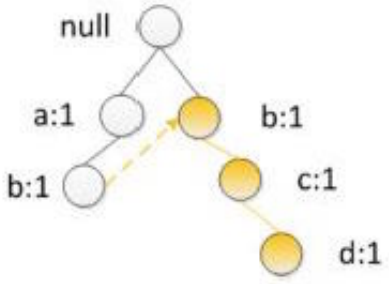


Frequent Item Set Mining
FP-Growth

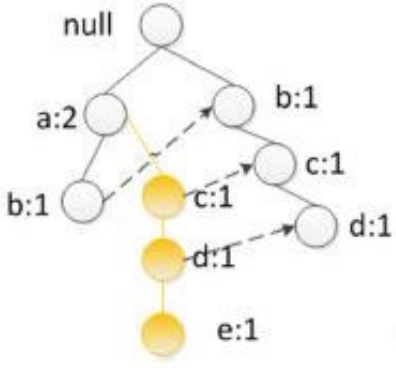
TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}



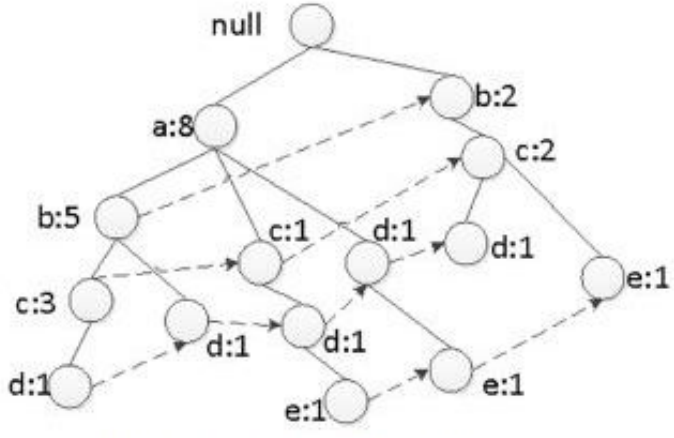
(i) After reading TID =1



(ii) After reading TID =2



(iii) After reading TID =3



(iv) After reading TID =10

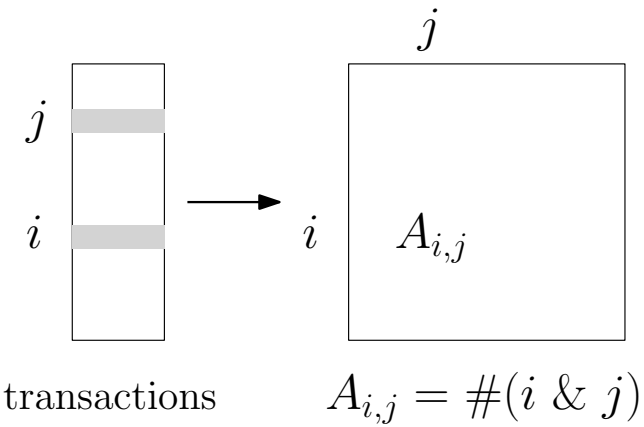
Han, Jiawei et al. "Mining frequent patterns without candidate generation." (2000)
Hájek, Petr et al. "The GUHA method of automatic hypotheses determination." (1966)

Logical Item Set Mining

- 1. phase - counting
  $$A(I_i, I_j) = \sum_{k=1}^{|T|} \delta(I_i \in T_k)\delta(I_j \in T_k)$$
- 2. phase - consistency
  $$B(I_i, I_j) = \frac{P(I_i, I_j)}{\sqrt{P(I_i)P(I_j)}} \in \langle 0, 1 \rangle$$
- 3. phase - iterative denoising
  $$C^{(t+1)}(I_i, I_j) \leftarrow C^{(t)}(I_i, I_j)\delta(B^{(t)}(I_i, I_j) > \theta)$$
- 4. phase - discovery
  $$G = (V = \{I_j | I_j \in I\}, E = \{(I_i, I_j) | C(I_i, I_j) > \theta\})$$

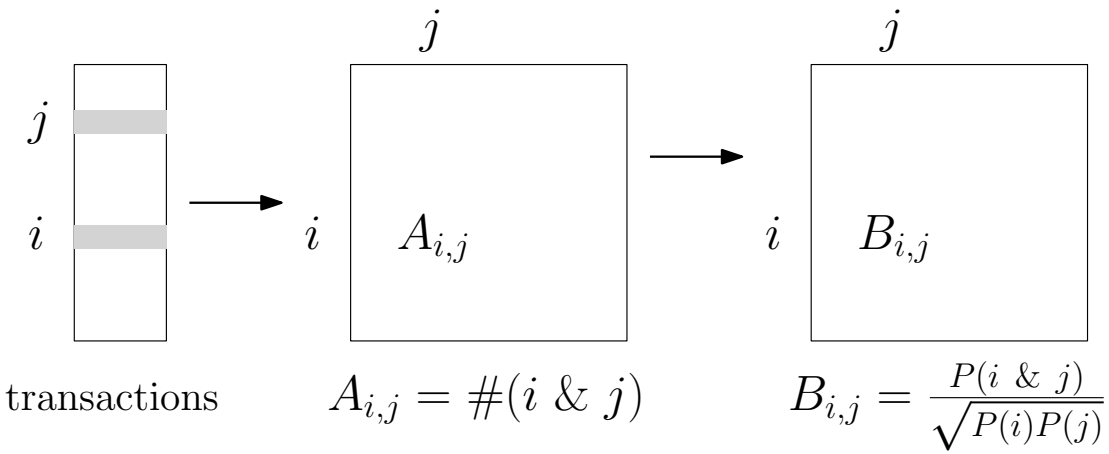
Kumar, Shailesh et al. "Logical itemset mining." (2012).

Logical Item Set Mining



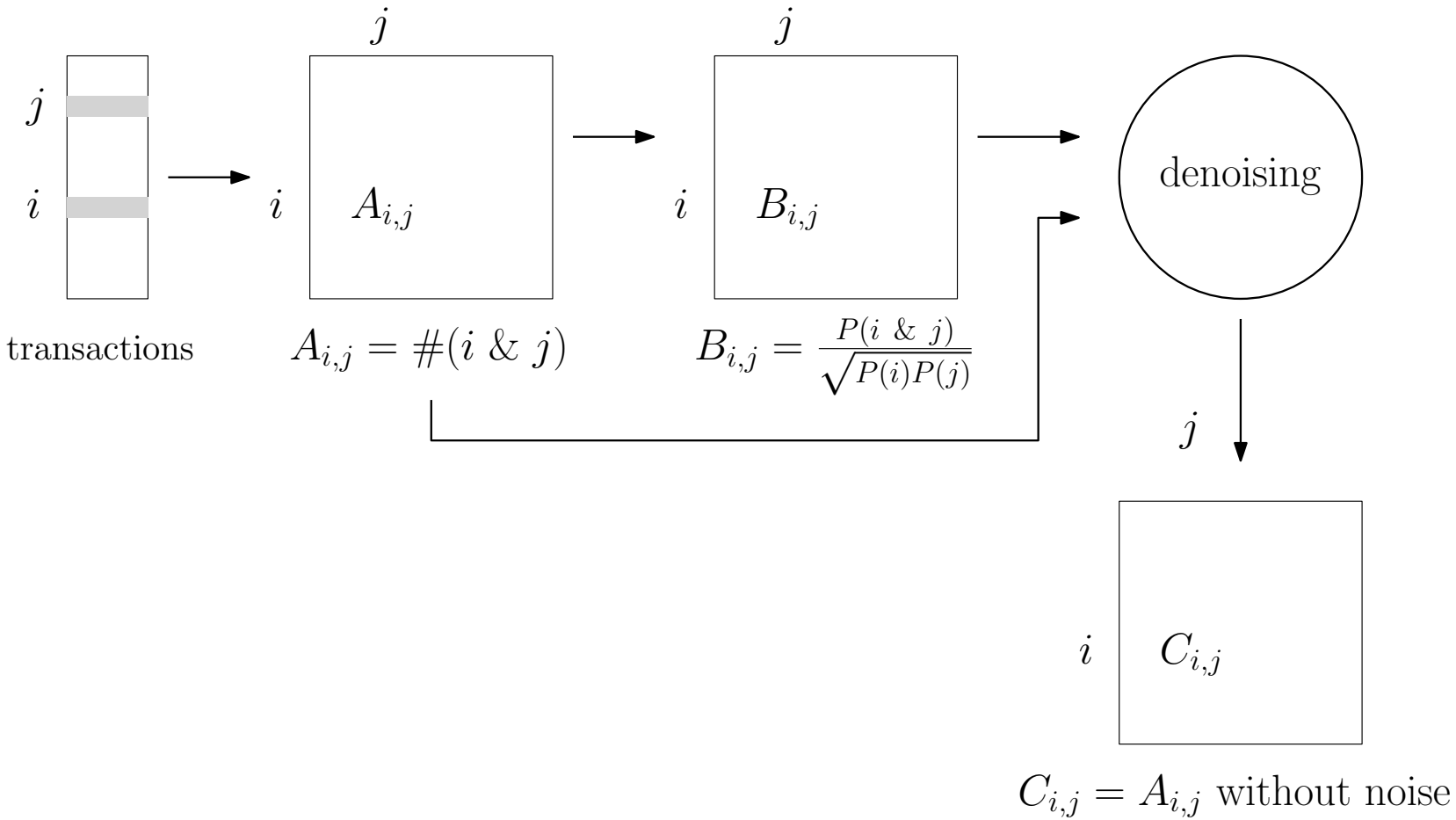
Kumar, Shailesh et al. "Logical itemset mining." (2012).

Logical Item Set Mining



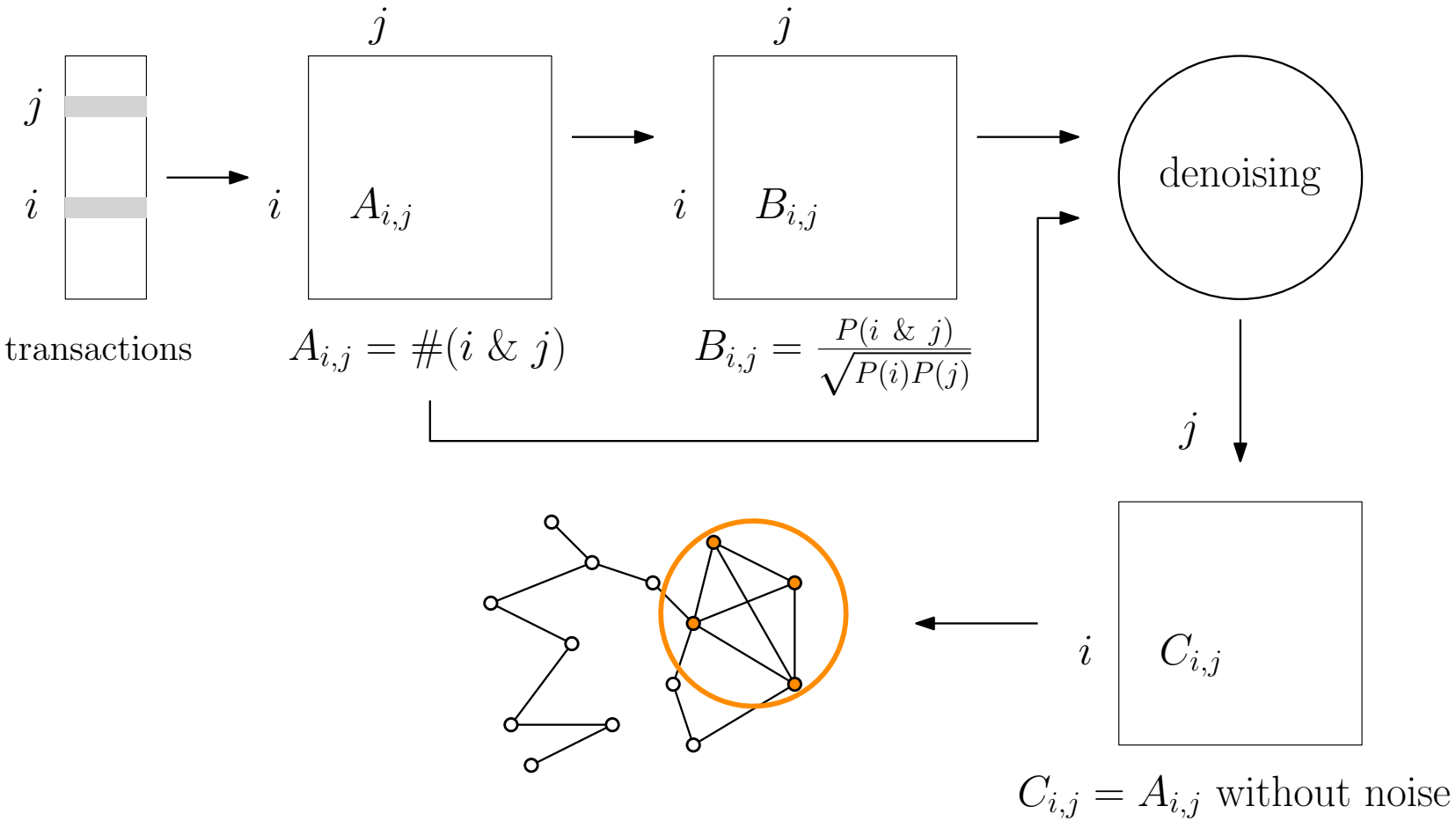
Kumar, Shailesh et al. "Logical itemset mining." (2012).

Logical Item Set Mining



Kumar, Shailesh et al. "Logical itemset mining." (2012).

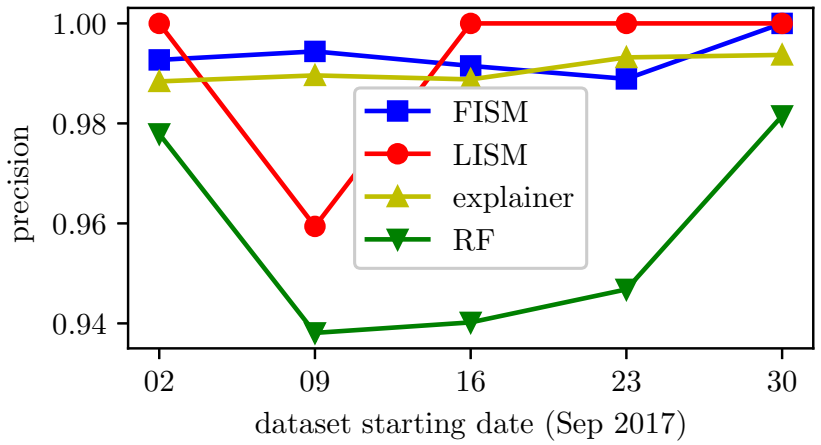
Logical Item Set Mining

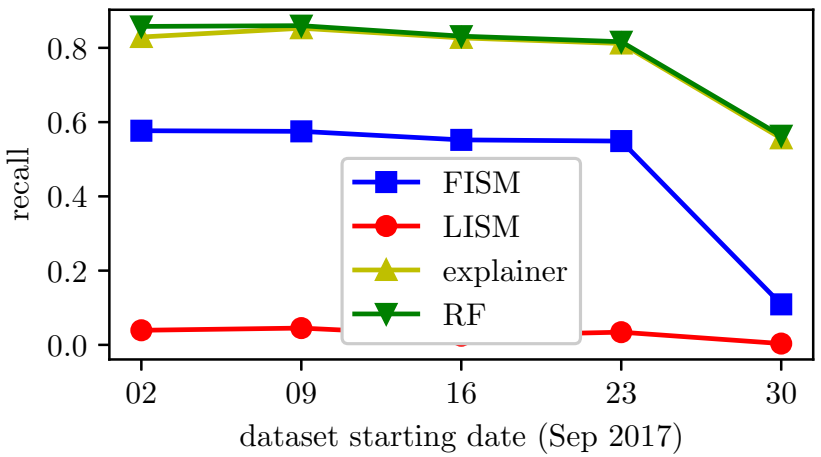


Kumar, Shailesh et al. "Logical itemset mining." (2012).

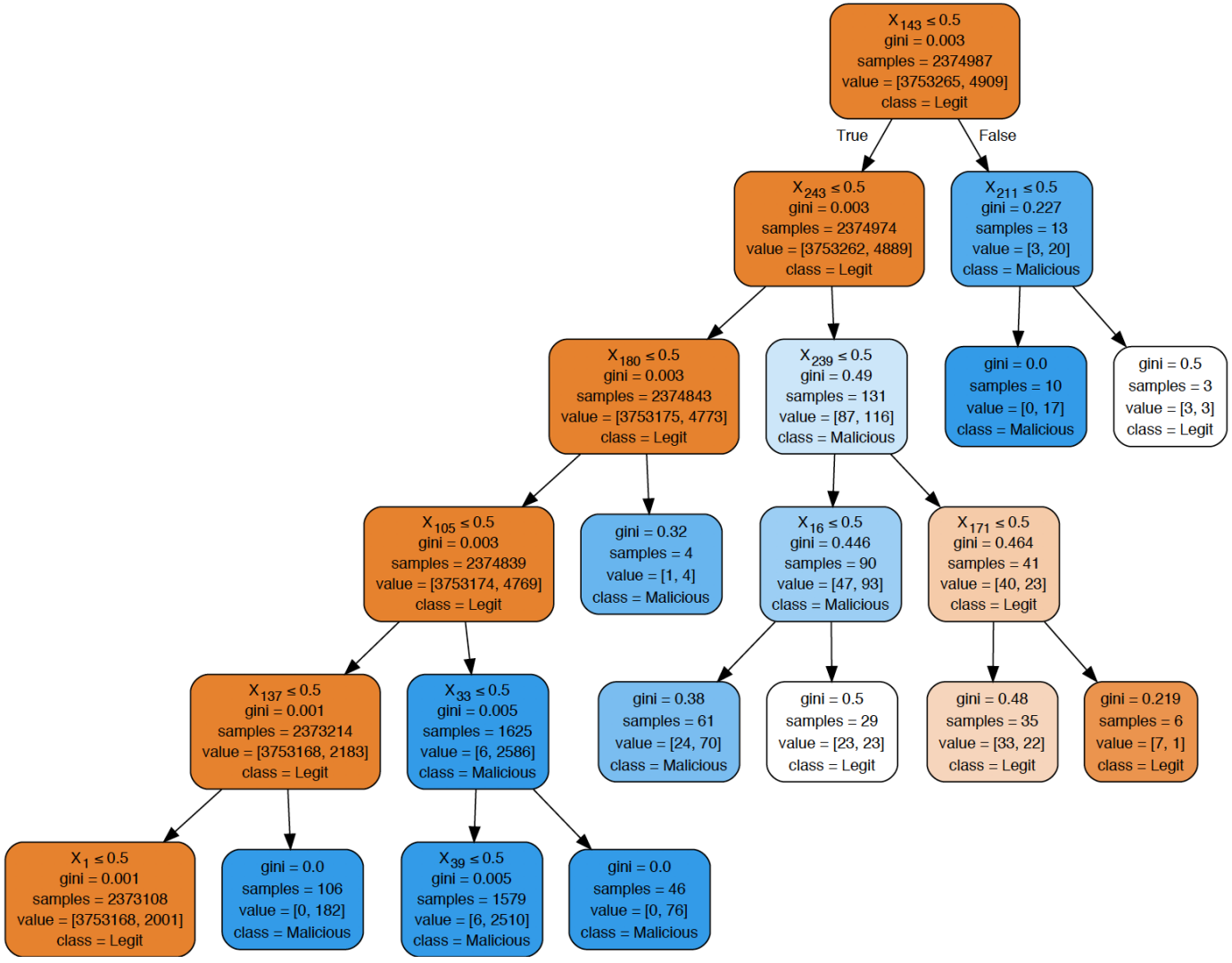

CONS
COMPARE
PROS

Comparison - performance





Explainability - Random Forest



Explainability - Explainer

Examples of extracted rules:

1. Shadow User, Click Fraud
2. Suspicious Advertising, Malicious Advertising

Explainability - Explainer

Examples of extracted rules:

1. Shadow User, Click Fraud
2. Suspicious Advertising, Malicious Advertising

pros

- precise
- nice recall

cons

- too short
- no added context

Explainability - Logical Item Set Mining

Examples of extracted rules:

1. Blocked, Malicious Binary, Sality
2. Click Fraud, Malwartising, Malware Distribution

Explainability - Logical Item Set Mining

Examples of extracted rules:

1. Blocked, Malicious Binary, Sality
2. Click Fraud, Malwartising, Malware Distribution

pros

- logical and justifiable
- never wrong

cons

- too specific
- very limited recall

Explainability - Frequent Item Set Mining

Examples of extracted rules:

1. jQuery, Suspicious Domain, Click Fraud
2. Path Count, Shadow User, Click Fraud, WordPress Management

Explainability - Frequent Item Set Mining

Examples of extracted rules:

1. jQuery, Suspicious Domain, Click Fraud
2. Path Count, Shadow User, Click Fraud, WordPress Management

pros

- adjustable length of rule
- valuable context
- good precision and recall

cons

- rule generation needed multiple heuristics
- additional rule post-processing

Conclusion

- Random forest cannot be used
- FISM seems optimal for our case
- rule extraction needed a significant hacks
- the post-processing is crucial for the output quality
- rule sets are redundant