

# **Where Are the Nuggets in System Audit Data?**

*Wenke Lee*

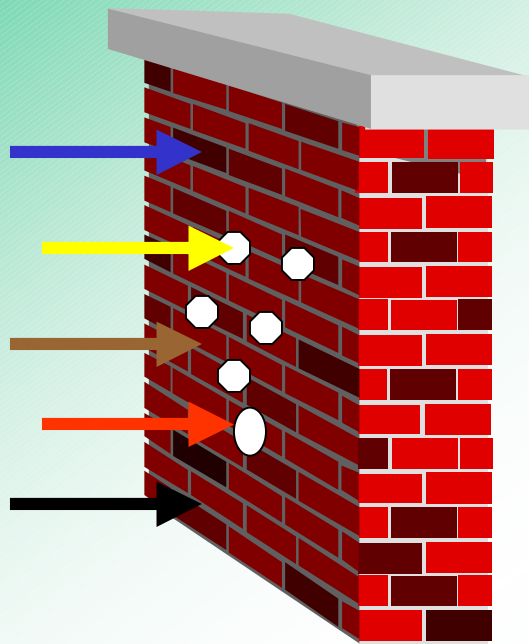
*College of Computing*

*Georgia Institute of Technology*

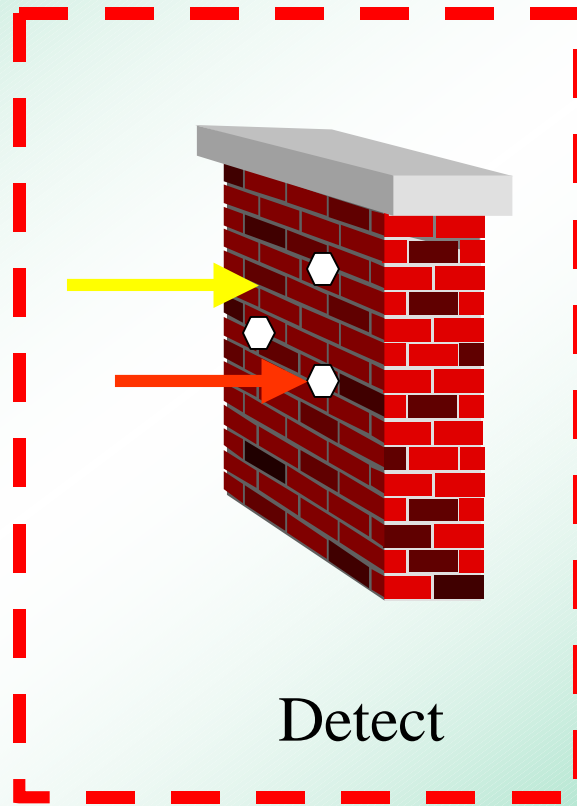
# Outline

- Intrusion detection approaches and limitations
- An example data mining (DM) based intrusion detection system (IDS)
- Lessons learned and challenges ahead
  - or where are the nuggets?

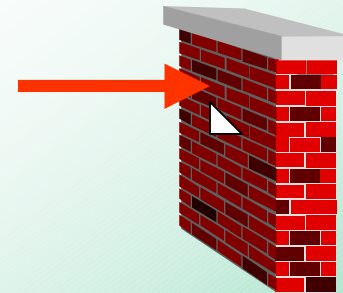
# Cyber Threats and Counter Measures



Prevent



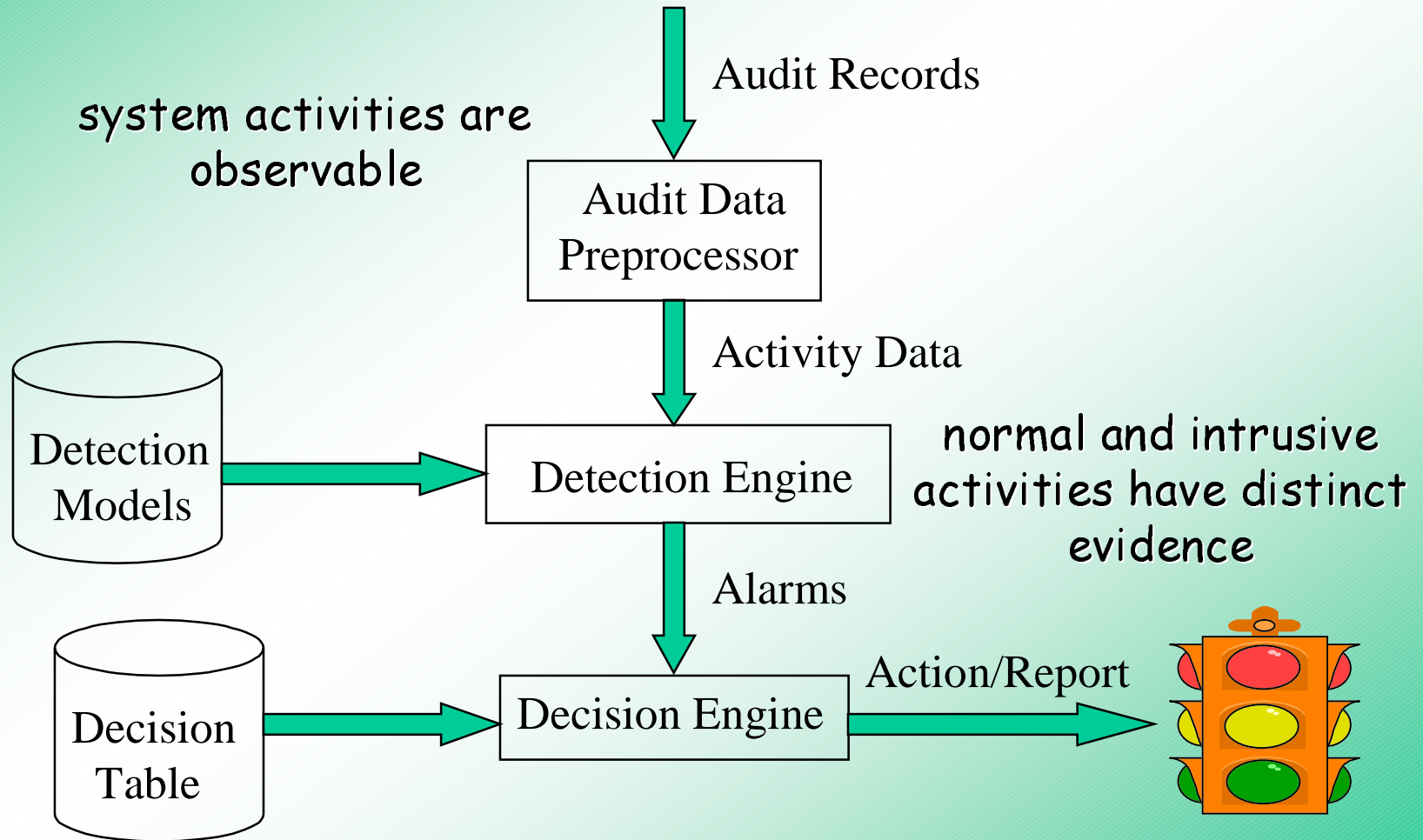
Detect



React/  
Survive

*Layered mechanisms*

# Components of Intrusion Detection



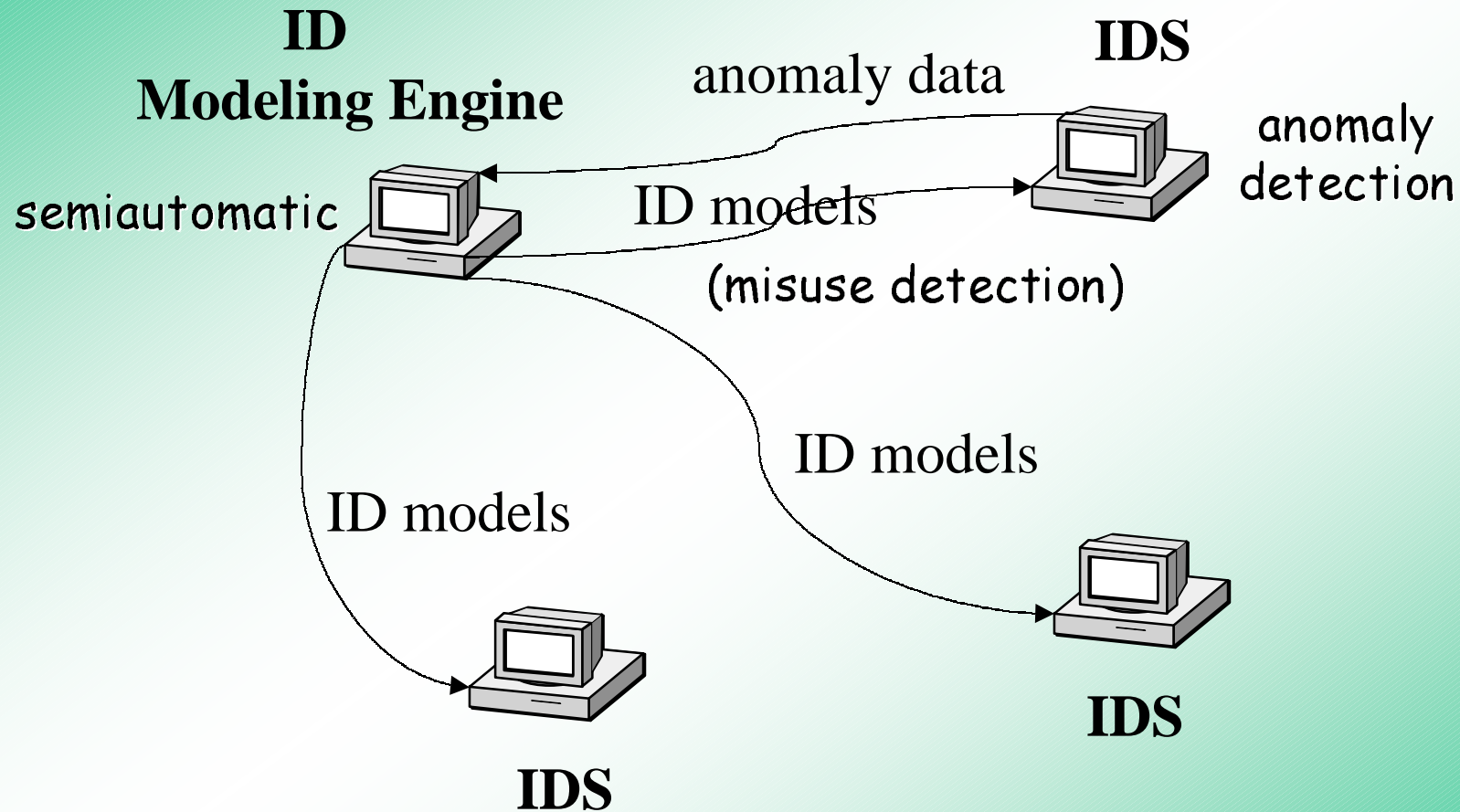
# Limitations of Current IDSs

- Misuse detection only:
  - “We have the largest knowledge/signature base”
  - Ineffective against new attacks
- Individual attack-based:
  - “Intrusion *A* detected; Intrusion *B* detected ...”
  - No ability to recognize attack plan
- Statistical accuracy-based:
  - “*x*% detection rate and *y*% false alarm rate”
    - Are the *most damaging* intrusions detected?

# Next Generation IDSs

- Adaptive and cost-effective
  - Detect new intrusions
  - Dynamically configure IDS components for best protection/**cost** performance
- Scenario-based
  - Correlate (multiple sources of) audit data and attack information

# Adaptive IDS – Model Coverage



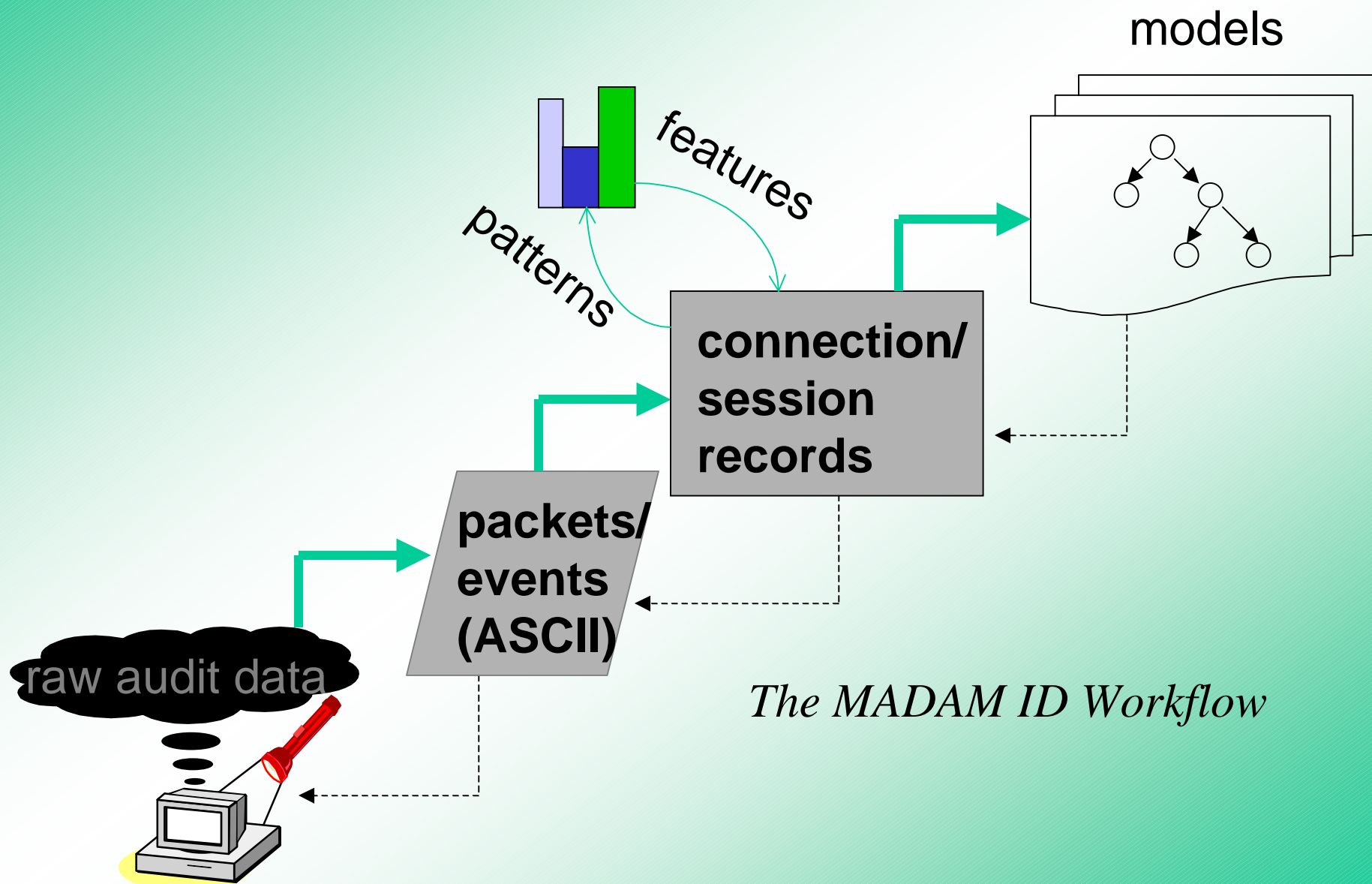
# Semiautomatic Model Generation

- Data mining based approach:
  - Build classifiers as ID models
- A prototype system:
  - MADAM ID
  - One of the best performing systems in the 1998 DARPA Evaluation

# Background in Data Mining

- Data mining
  - Applying specific algorithms to extract valid, useful and understandable patterns from data
- Why applying data mining to intrusion detection?
  - Motivation
    - Semi-automatically construct or customize ID models for a given environment
  - From the data-centric point view, intrusion detection is a data mining/analysis process
  - Successful applications in related domains, e.g., fraud detection, fault/alarm management

# The Iterative DM Process of Building ID Models

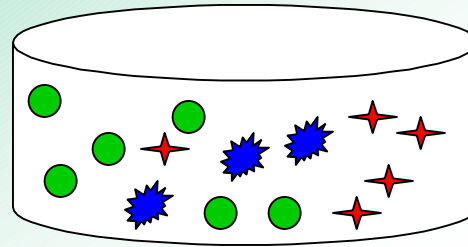


*The MADAM ID Workflow*

# ID as a Classification Problem

higher entropy  
(impurity)

$$E(X) = -\sum_x P(x) \log(P(x))$$

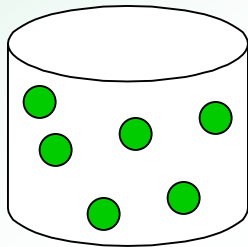


use features with  
high information  
gain - reduction in  
entropy

$F_1=v_1$

$F_1=v_2 \&\& \dots$

$F_5=v_5 \&\& \dots$



lower entropy  
(purer)

# The Feature Construction Problem

<i>dst ... service ... flag</i>		<i>dst ... service ... flag %S0</i>
h1 http S0	} syn flood	h1 http S0 70
h1 http S0		h1 http S0 72
h1 http S0		h1 http S0 75
h2 http S0	} normal	h2 http S0 0
h4 http S0		h4 http S0 0
h2 ftp S0		h2 ftp S0 0

existing features  
useless

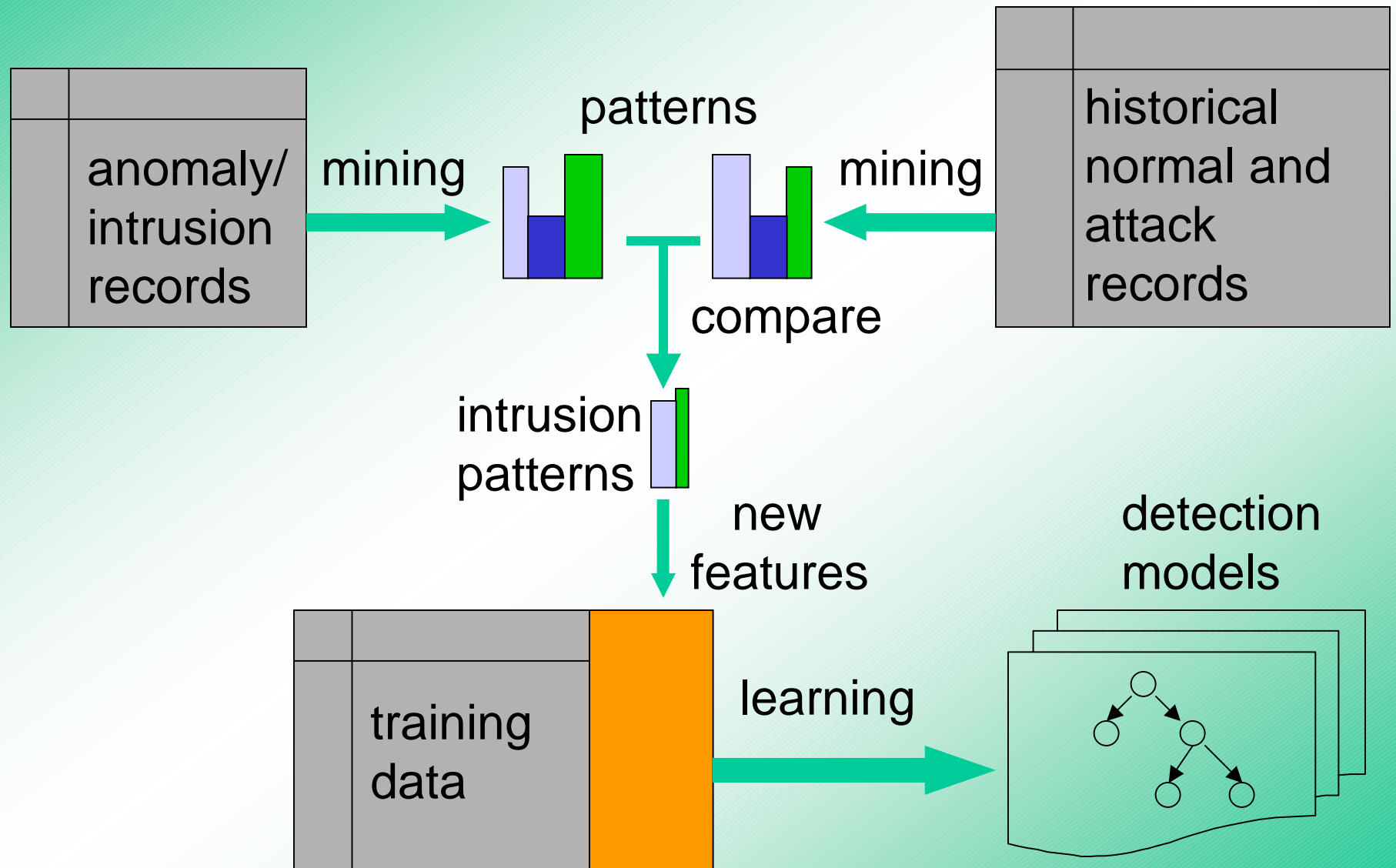
construct features with  
high information gain

How? Use temporal and statistical patterns, e.g., "a lot of S0 connections to same service/host within a short time window"

# Mining Patterns

- Associations of features
  - e.g. (service=http, flag=S0)
  - Basic algorithm: association rules
- Sequential patterns in activity records
  - e.g. (service=http, flag=S0), (service=http, flag=S0) → (service=http, flag=S0) [0.8,2s]
  - Basic algorithm: frequent episodes

# Feature Construction from Patterns



# Feature Construction Example

- An example: “syn flood” patterns (*dst\_host* is *reference* attribute):
  - (flag = S0, service = http), (flag = S0, service = http) → (flag = S0, service = http) [0.6, 2s]
  - add features:
    - count the connections to the same *dst\_host* in the past 2 seconds, and among these connections,
    - the percentage with the same *service*,
    - the percentage with S0

# The Nuggets

- Feature extraction and construction
  - The key to producing effective ID models
  - Better pay-off than just applying another model learning algorithm
  - How to semi-automate the feature discovery process (by incorporating domain knowledge)?

# Feature Construction: the MADAM ID Example

- Search through the feature space through iterations, at each iteration:
  - Use different heuristics to compute patterns (e.g., per-host service patterns) and construct features accordingly
- Limitations:
  - Connection level only
  - Within-connection contents are not “structured”, and much more challenging!

# The Nuggets (continued)

- Efficiency
  - Training
    - Huge amount of audit data
      - Sampling?
    - Always retrain from scratch or incrementally?
  - Execution of output model in real-time
    - Consider feature cost (time)
    - Trade-off of cost vs. accuracy

# Cost-sensitive Modeling: an Example

- A multiple-model approach:
  - Build multiple rule-sets, each with features of different cost levels;
  - Use cheaper rule-sets first, costlier ones later only for required accuracy.
- 3 cost levels for features:
  - Level 1: beginning of an event, cost 1;
  - Level 2: middle to end of an event, cost 10;
  - Level 3: multiple events in a time window, cost 100.

# The Nuggets (continued)

- Anomaly detection
  - What is a general approach?
  - Taxonomy and specialized algorithm for each type?
  - Theoretical foundations?

# Conclusions

- There is a need for DM in ID
- Research should be focused on the real nuggets:
  - Feature construction
  - Efficiency
  - Anomaly detection

**Thank You!**