SRI International

Communication Pattern Monitoring: Improving the Utility of Anomaly Detection for Industrial Control Systems

Man-Ki Yoon (UIUC) and **Gabriela F. Ciocarlie** (SRI International)

NDSS Workshop on Security of Emerging Networking Technologies (SENT)
February 23rd 2014

Motivation

- Targeted attacks on industrial control systems (ICS) are growing in frequency and severity
  - 7,200 Internet-facing control system devices in U.S. [1]
- Industrial control systems use specialized but insecure communication protocols
  - Enterprise security tools are not able to identify zero-day attacks specific to these protocols
- **Alternative**: **anomaly-based** detection (AD) sensors
  - Natively well-suited for detecting zero-day attacks

[1] DHS ICS-CERT Monitor, October/November/December 2012

Motivation – AD Sensors

- Control systems exhibit constrained behavior:
  - Fixed topology
  - Regular communication patterns
  - Limited number of protocols
  - Simpler protocols

- Content-based anomaly detection
  - Sequence of commands, command data, request/response

- Extensible & modular framework
  - Common analysis method for different protocols

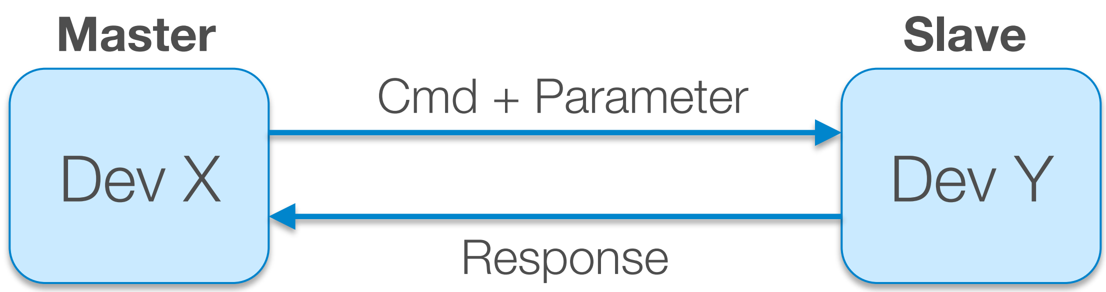
Main Contributions

- A new probabilistic-suffix-tree-based approach for ICS anomaly detection, which extracts the *normal patterns of command* and *data sequences* from ICS communications

- A false positive rate reduction mechanism, instrumental for ICS environments

- An implementation of the proposed approached applied to both real and simulated datasets

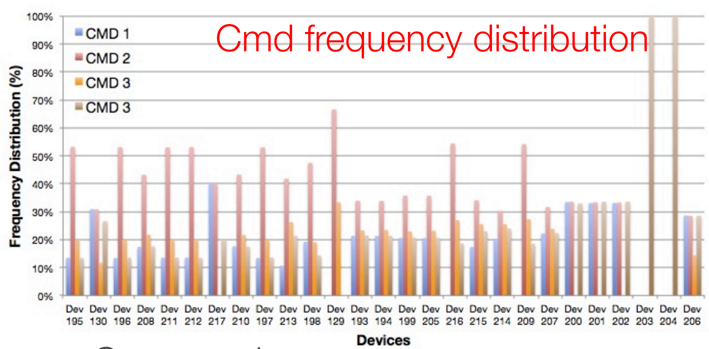
Connection Model

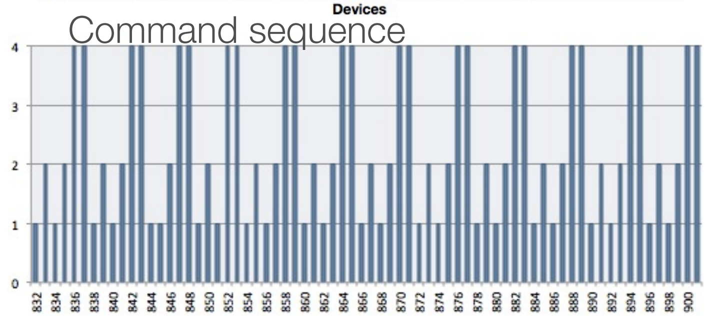


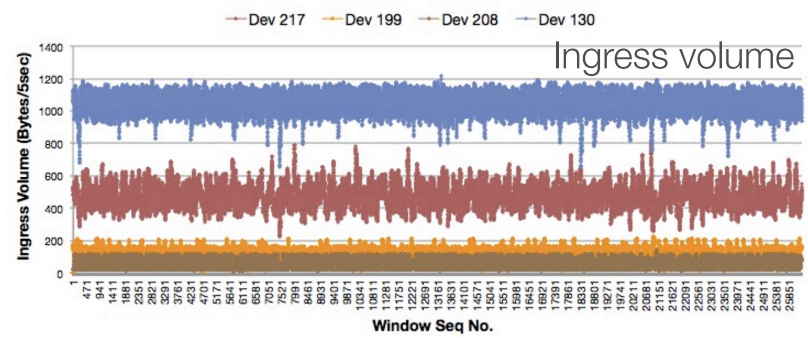
- Slave can receive N command types
- For the same command type,
  - parameters can vary, but not much
  - responses depend on the <Cmd, Parameter> pair
- Devices will have an ‘internal’ state
  - May not be directly visible
  - Operational modes, normal/compromised

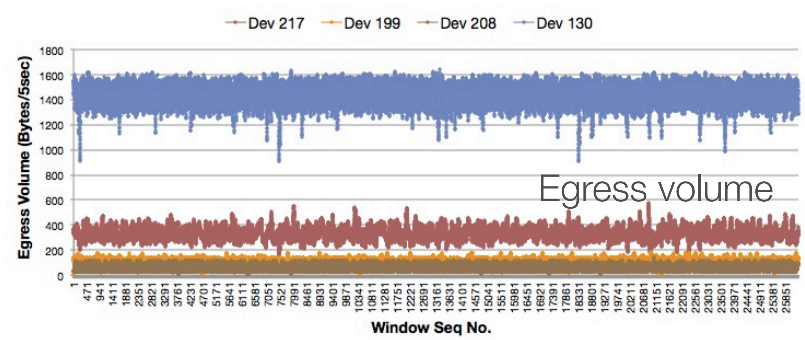
Predictable Behavior of ICS Network

- Globally?
  - No. Devices behavior change with different frequencies.
- Device level?
  - Better, but still not deterministic as a device may communicate with many devices
- Connection level?
  - Stable, deterministic!



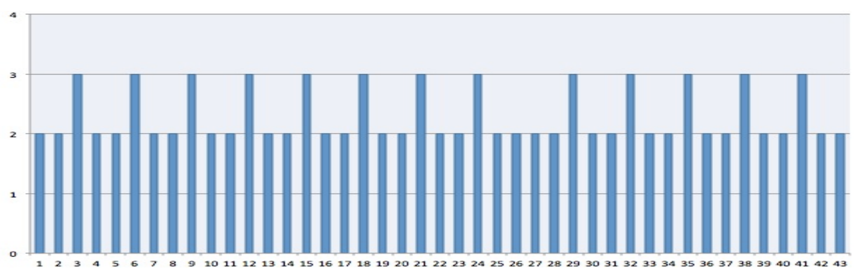


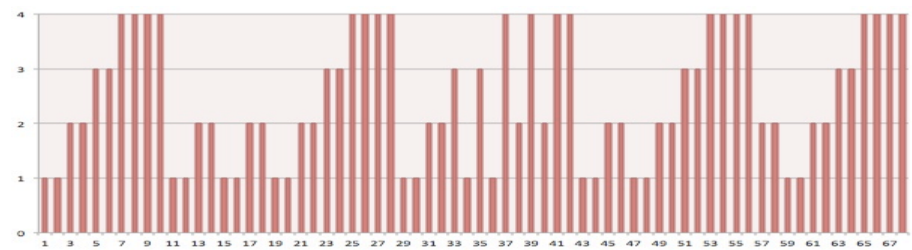




Patterns for Commands and Data

- Given a connection, the sequence of commands has patterns
  - Periodic operations -> form a transaction of commands





- Given a command type over a connection, data is mostly either
  - a fixed value or
  - a value changing with a pattern

- Both can be modeled as sequence patterns

9c410001
9c420001
9c430001
9c440001
9c450001
9c460001
9c470001
9c480001
9c490001
9c4a0001
9c4b0001
9c4c0001
9c410001
9c420001
9c430001
9c440001
9c450001

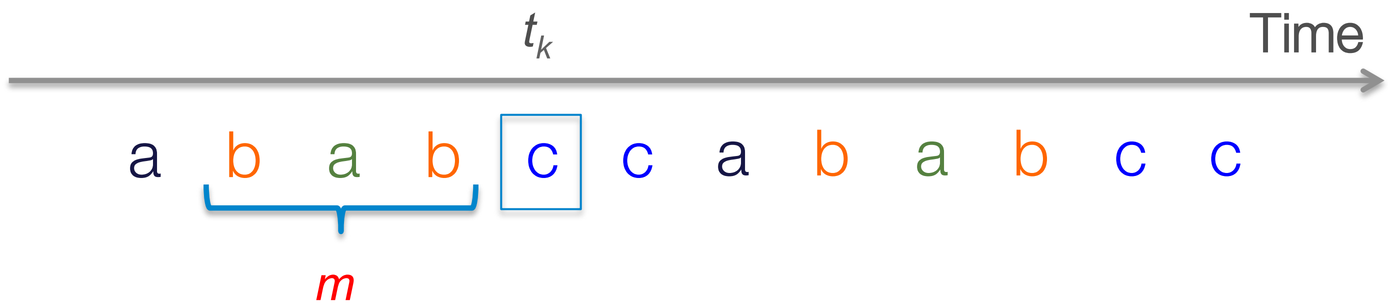
Patterns for Commands and Data

- Given a connection, the sequence of commands has patterns
  - A transaction of commands (operations) -> a pattern of commands
- Given a command type over a connection, data is mostly either
  - a fixed value or
  - a value changing with a pattern
- Both can be modeled as sequence patterns

We detect anomalies in command and data sequences

- Master sends unknown commands
  - with normal/abnormal data
- Master sends known but abnormal commands
  - out of context
- Slave responds with abnormal response data
- Master sends requests to unusual slaves
  - that it has never/rarely communicated with

How to Model Sequence Patterns?



- What is the probability of seeing a certain command at time t_k given a history of commands of length m?

Learning Patterns of Commands and Data

- Learning the normal sequence of commands = Learning a Markov chain of order *m*

- Challenges
  - Packets can be missing
  - Patterns may vary

- Need for a probabilistic approach
  - Learn the conditional probability distribution (CPD)

$$Pr(\sigma_t | \sigma_{t-m} \cdots \sigma_{t-1})$$

Learning Patterns Using PST

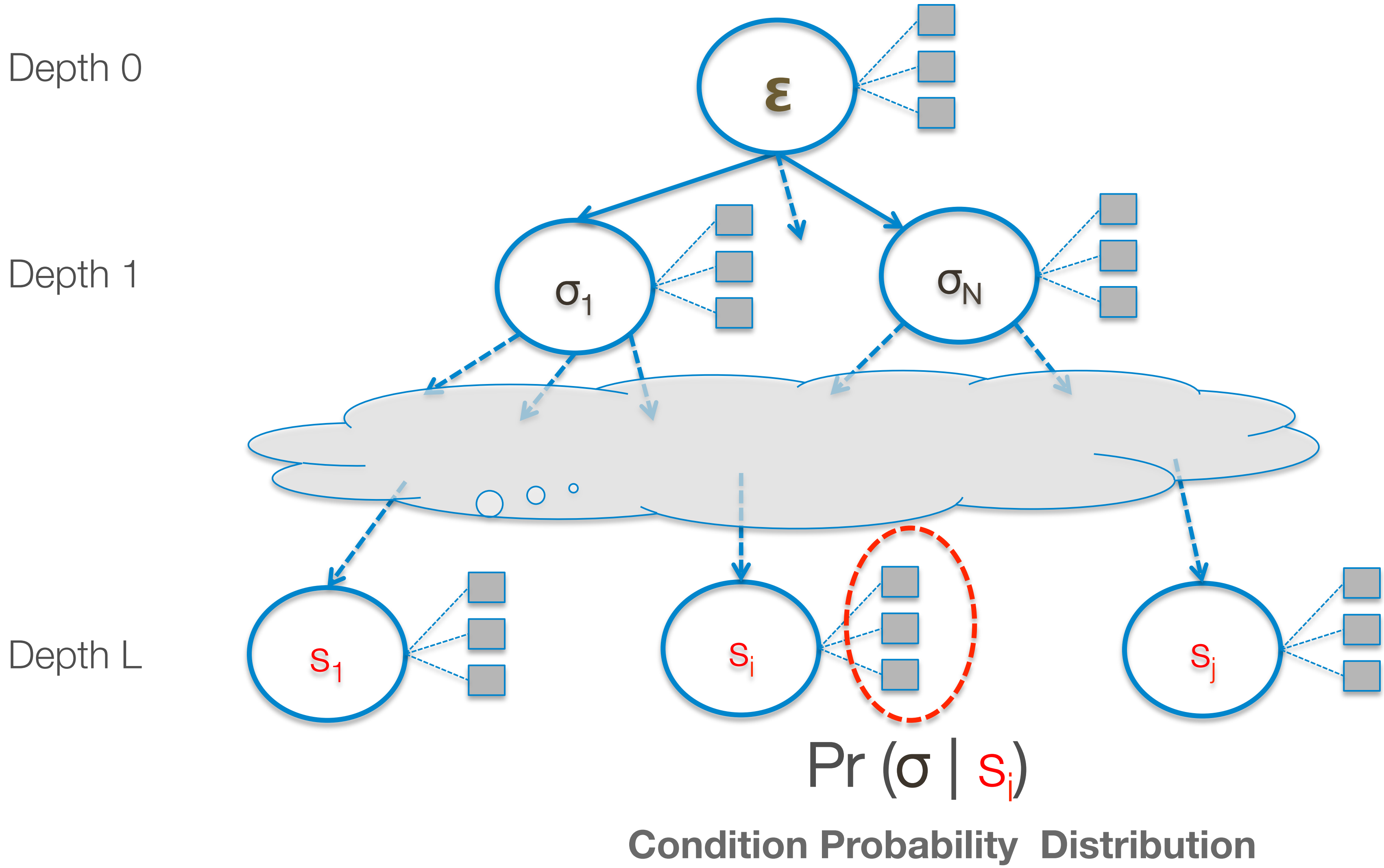
- Probabilistic Suffix Tree (PST)
  - A variable-order Markov model
  - Bounded depth (the maximum order), *L*

$$Pr(\sigma_t | \sigma_1 \sigma_2 \cdots \sigma_{t-1}) \sim Pr(\sigma_t | \sigma_{t-k} \cdots \sigma_{t-1})$$

, where k <= L

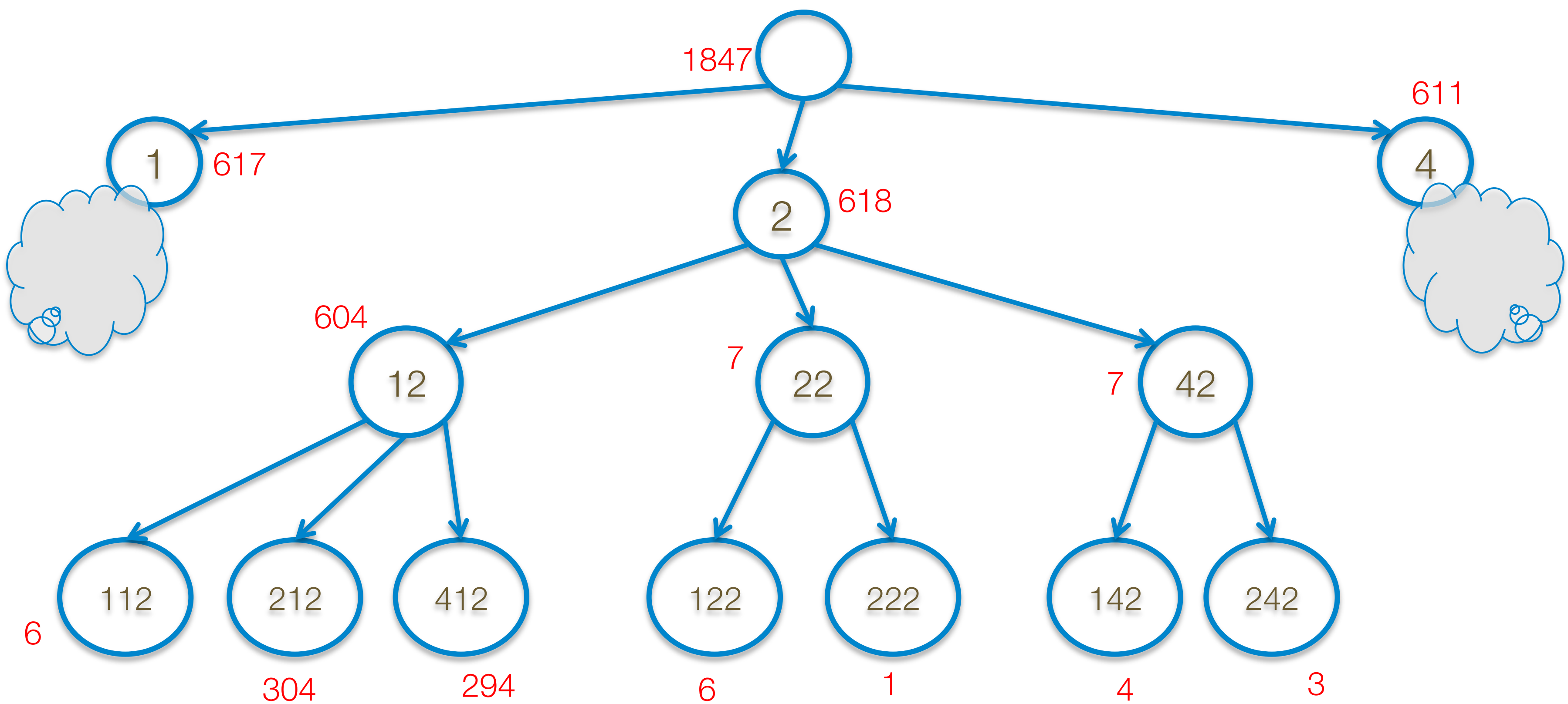
  - Efficiently represents CPD using tree structure

PST Structure



PST Example

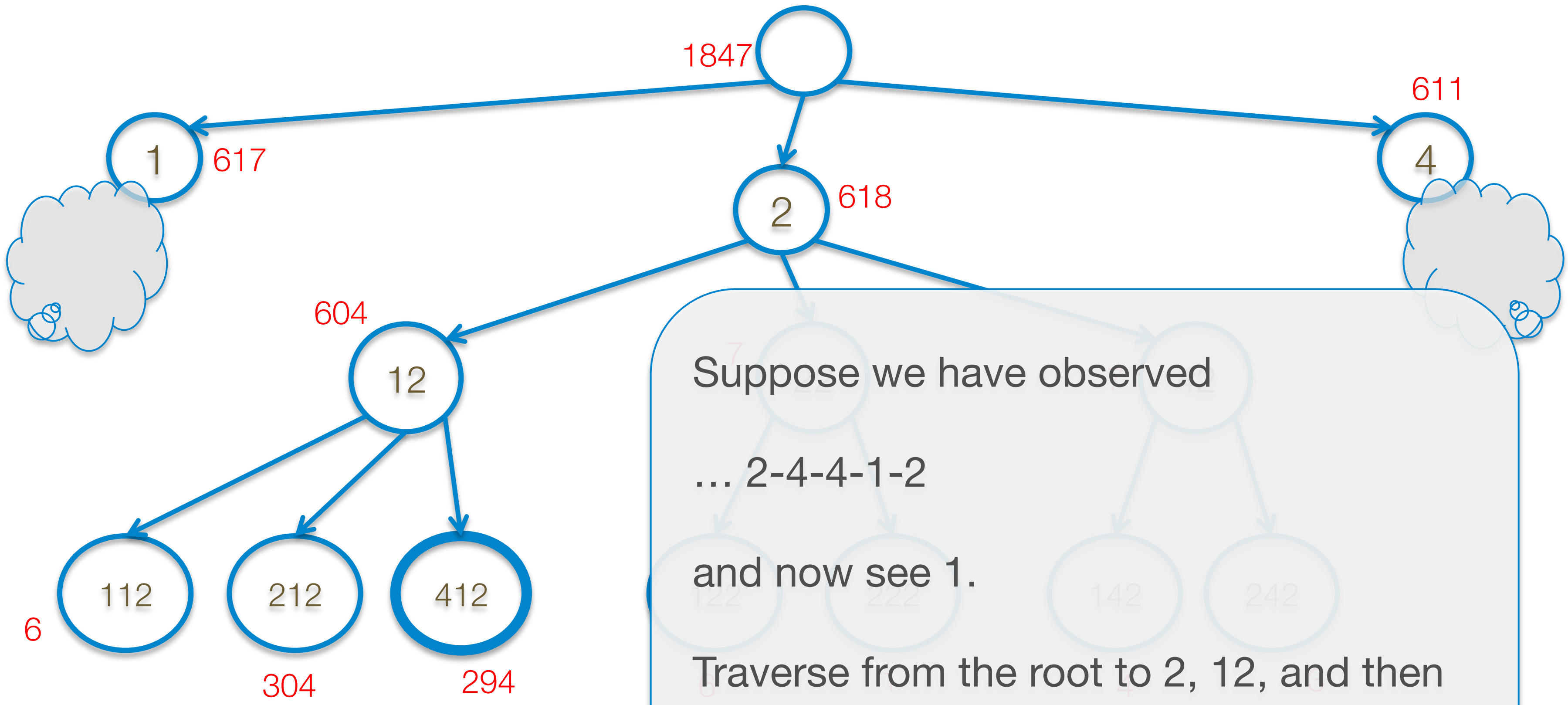
Base pattern: 1 2 1 2 4 4 L (MaxDepth) = 3



Likelihood Calculation

Base pattern: 1 2 1 2 4 4

L (MaxDepth) = 3



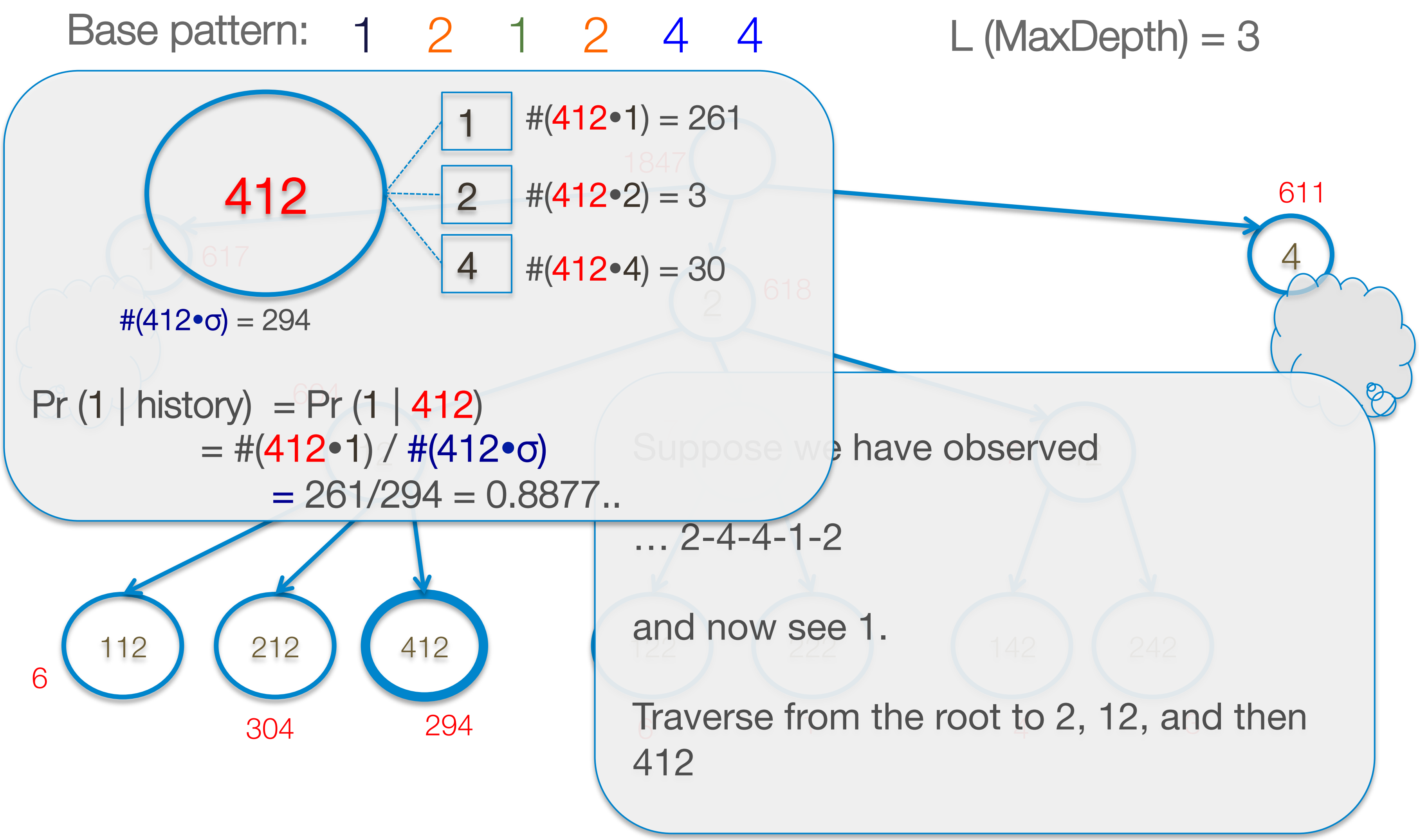
Suppose we have observed

... 2-4-4-1-2

and now see 1.

Traverse from the root to 2, 12, and then 412

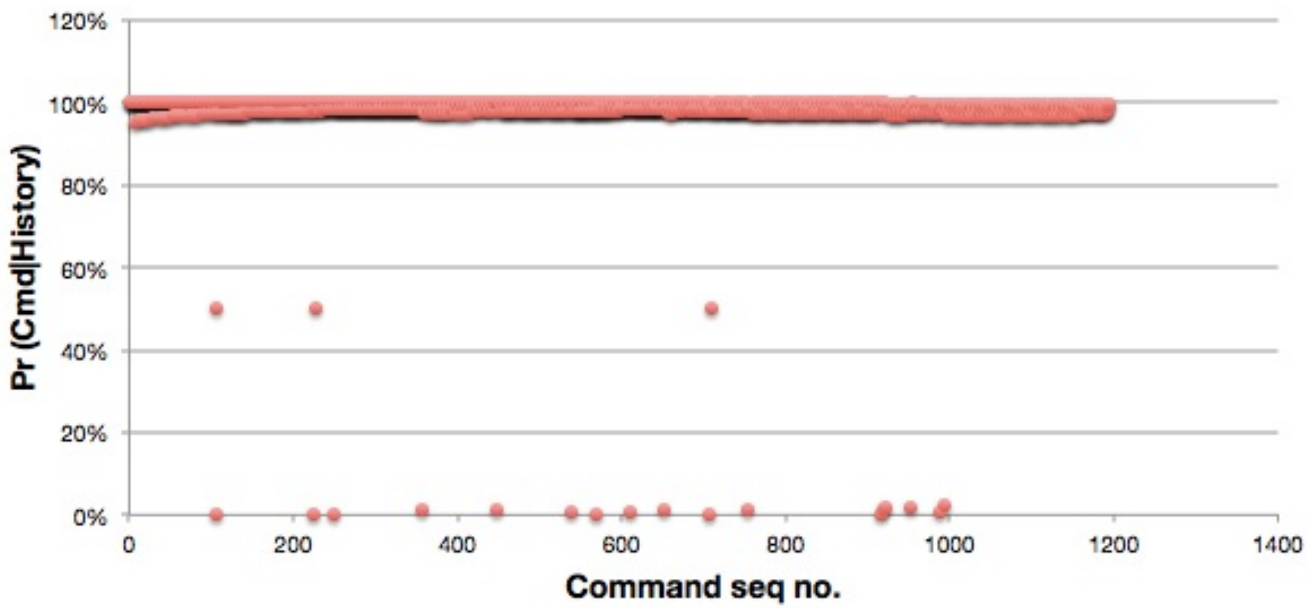
Likelihood Calculation



Incremental PST

- For online learning
  - Batch learning is not applicable to network-level AD due to the flow of packets
  - Need to be able to deal with varying patterns
- Update the tree whenever reading an element, **σ**
  - Start from an empty tree
  - Keep recently-read elements
  - Update the counts $\#(s \bullet \sigma)$ for recent history s of length 1,…, L

Incremental PST Example



- A MODBUS connection
  - Base pattern: 1-2-1-2-4-4
  - Normal sequence
  - Mostly, the likelihoods are close to 1.0
  - Sometimes, near zero -> because of missing packets!

False Positive Due to Missing Packets

Base pattern: 1 2 1 2 4 4 L (MaxDepth) = 3

1-2-1-2-4-4 - 1-2-2-4 - 1 Time

Pr(2|4-1-2) = 1.69%

- Missing one packet can cause multiple false positives
  - In the example, missing '1' causes two false positives
- We want low false positive rate!

Incremental PST with Prediction

- If $Pr(\sigma_t|\sigma_{t-L}\cdots\sigma_{t-1}) < \theta$
  - assume an element is missing and try to restore it!

- First, find what we should have seen.

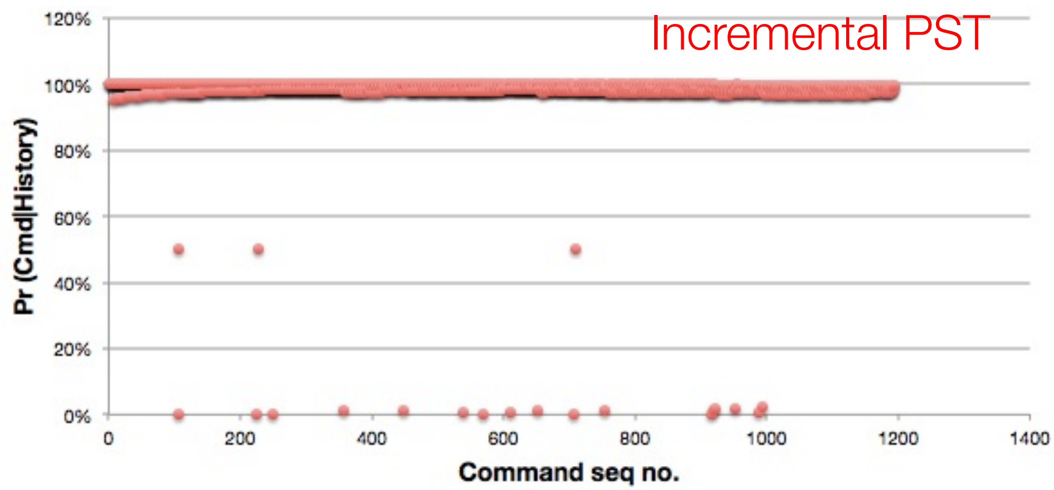
$$\sigma_{ML} = \arg\max_{\sigma} Pr(\sigma|\sigma_{t-L}\cdots\sigma_{t-1})$$

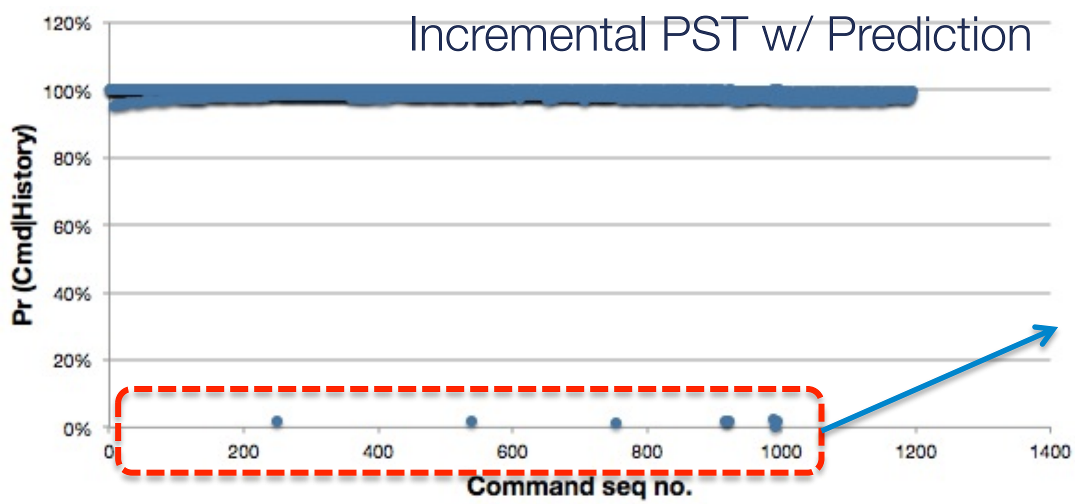
- Then, use it to calculate the new likelihood

$$\cancel{\sigma_{t-L}}\sigma_{t-L+1}\cdots\sigma_{t-1} \longrightarrow \underbrace{\sigma_{t-L+1}\cdots\sigma_{t-1}\sigma_{ML}}_{\text{Length = L}}$$

$$\boxed{\begin{aligned}&Pr(\sigma_t|\sigma_{t-L}\cdots\sigma_{t-1})\\&\quad\sim Pr(\sigma_{ML}|\sigma_{t-L}\cdots\sigma_{t-1})\cdot Pr(\sigma_t|\sigma_{t-L+1}\cdots\sigma_{t-1}\sigma_{ML})\end{aligned}}$$

Incremental PST with Prediction Example





Reduced many FP!
But, still, some are FP.

4-4-1-2-4

1-2-1-2-1-4

1-2-1-2-1-2

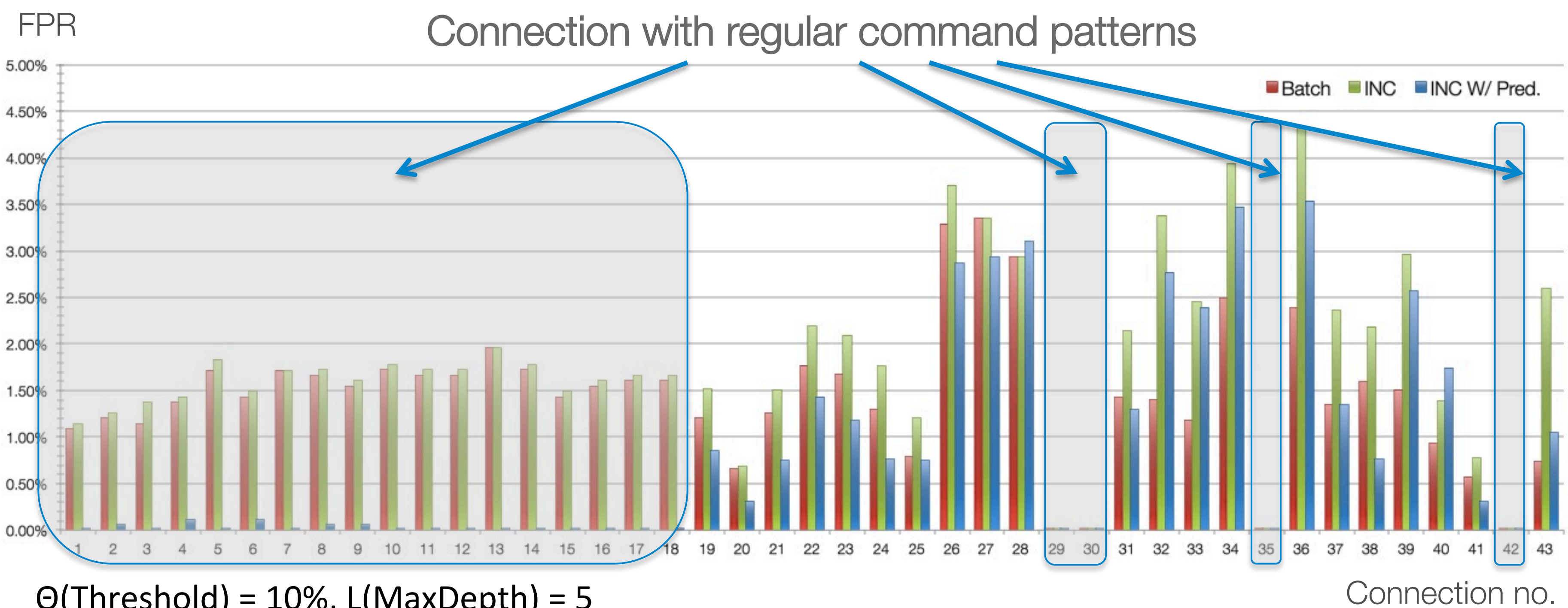
It doesn't restore well when consecutive packets are missing!

Evaluation

- Modbus traffic
  - 2 masters, 25 slaves
  - 86 connections (43 pairs)
  - 4 cmd types
  - No attack/anomaly is known
  - Some packets are missing

- Synthetic data (random sequences of commands)
  - Evaluate the detection rate and the false positive rate

False Positive Rates of Modbus Traffic



Θ(Threshold) = 10%, L(MaxDepth) = 5

Likelihood

$$FPR = \frac{\sum_{t=1}^{N} I\Big(\overbrace{Pr(\sigma_t|\sigma_1 \cdots \sigma_{t-1})} < \theta)\Big)}{N}$$

$I(true) = 1$

Sequence length

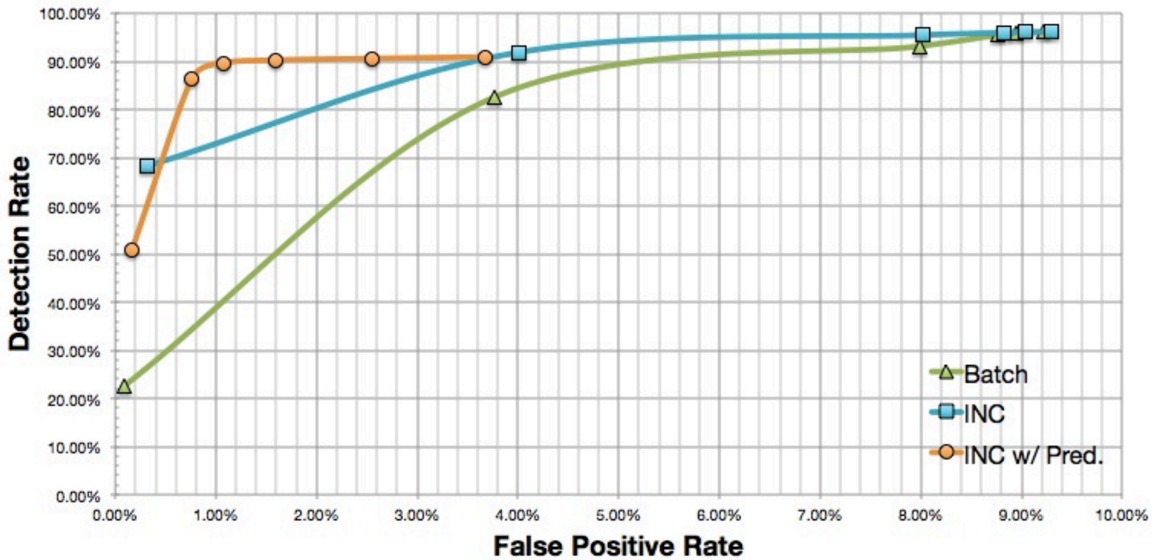
Generation of Random Sequence of Commands

- Generate a random base pattern

- Then, generate a random sequence based on the pattern
  - With a missing probability, a command can be dropped
  - With an attack probability, a random short sequence is inserted

- Input parameters
  - Min, max of base pattern length
  - # of command types
  - Missing, attack probabilities

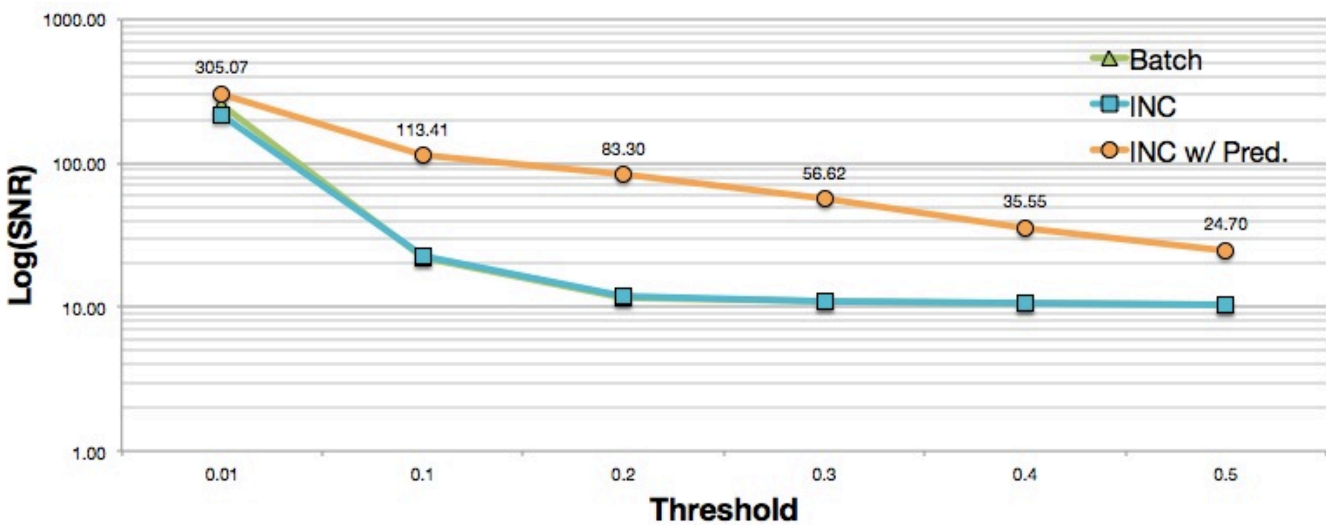
Better Performance for INC w/Pred



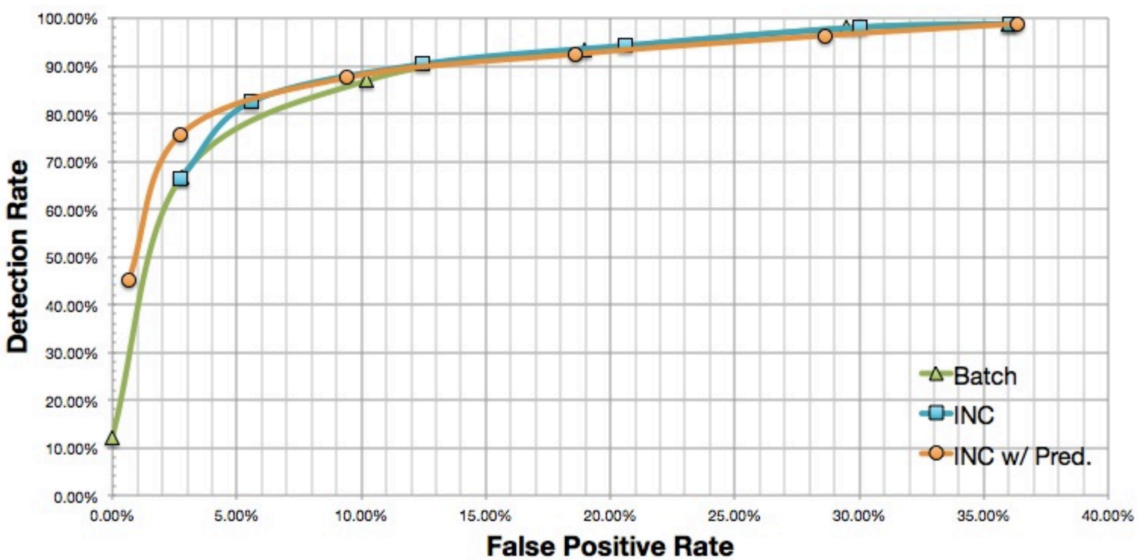
ROC curve

Miss prob = 10%
MaxDepth(L) = 5

Signal-Noise Ratio



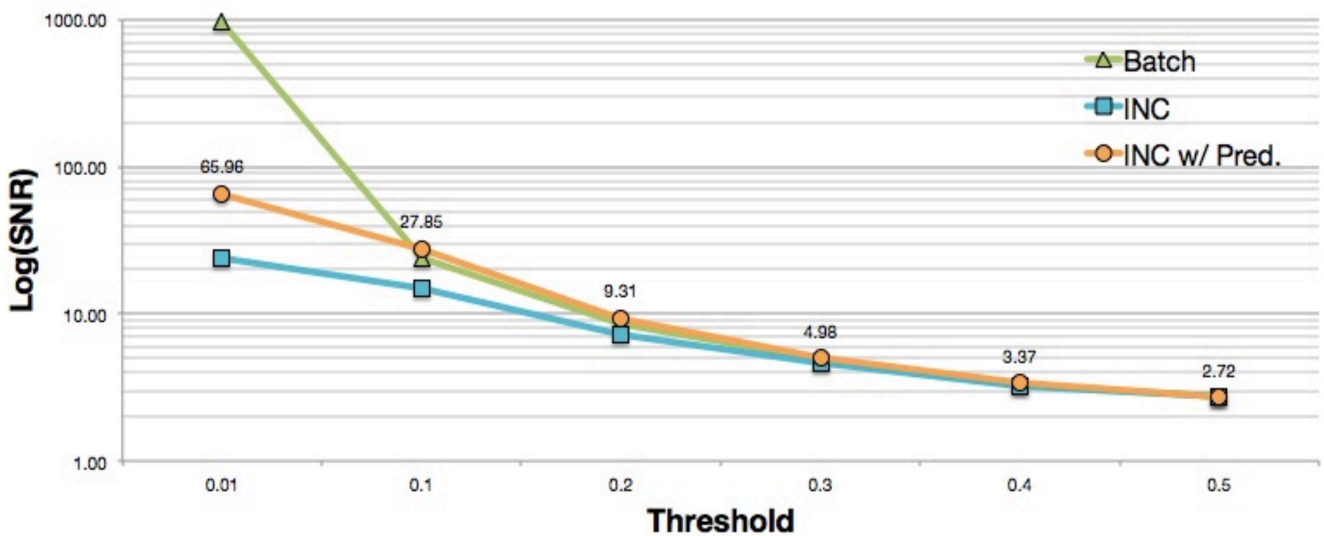
Similar Performance Across All Methods



ROC curve

Miss prob = 50%
MaxDepth(L) = 5

Signal-Noise Ratio



Conclusions

- We proposed a novel anomaly detection method for ICS devices
  - Built accurate models
  - Reduced false positive rate
- The proposed method has been implemented and applied to a Modbus network testbed and a synthetic dataset
  - Reached a high detection rate for the synthetic dataset while successfully keeping the false positive rate in check

Future Work

- A complete evaluation on real operational datasets will be a critical next step
  - We are currently analyzing real Modbus traffic
- We plan to extend the set of protocols that we investigate and to target different industry sectors
- We plan to also extend the ICS-specific anomaly detection techniques within a more flexible and general framework, that can cope with long lasting attacks targeting our architecture

Thank you!

This work was funded by the US Department of Homeland Security (DHS) Science and Technology (S&T) Directorate under contract no. HSHQDC-10-C-00144. The views and conclusions contained herein are the authors' and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DHS or the US government.

Headquarters: Silicon Valley

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493
650.859.2000

Washington, D.C.

SRI International
1100 Wilson Blvd., Suite 2800
Arlington, VA 22209-3915
703.524.2053

Princeton, New Jersey

SRI International Sarnoff
201 Washington Road
Princeton, NJ 08540
609.734.2553

Additional U.S. and international locations

www.sri.com