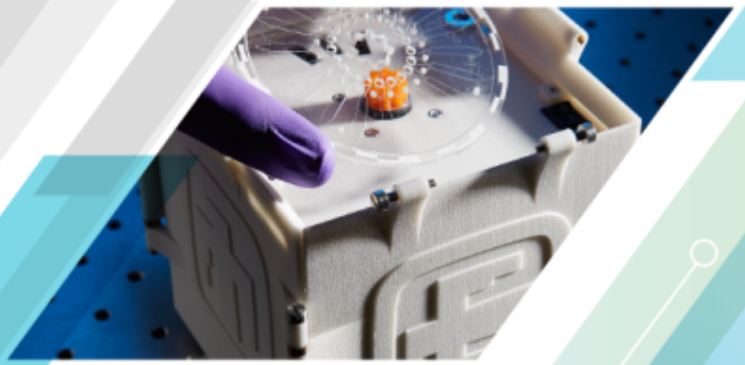
SAND2022-8922C

Energy & Homeland Security

MLDL 2022:

A Bayesian Network Pipeline for Detection of Cyberattacks



Presented by

Nathanael Brown (5521)

Other contributors: Matt Hoffman (5523), Barnett Yang (5523; currently on “sabbatical”)

Funded by E/HS LDRD 218338: *Detecting Unknown Cyber Attacks using Bayesian Networks*

July, 2022

MACHINE LEARNING & DEEP LEARNING

MLDL

NNSA U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

LDRD - *DETECTING UNKNOWN CYBER ATTACKS USING BAYESIAN NETWORKS*

- **Challenge:**
  - Most tools still identify **isolated indicators or techniques** and are only marginally useful because they "…bludgeon analysts with 'just to be safe' indications and warnings" resulting in a "**crush of false positives**." – SNL Cyber SME John Jarocki
- **Proposed Solution:**
  - Use Bayesian Networks (BNs) as the core of a host-based cyber intrusion detection system
  - Detect subtle malicious behavior by fusing existing SME knowledge with event data to create an interpretable solution using multivariate relationships
  - Enable SME customization to tune the system based on the specific application
  - Avoid overloading analysts with false alarms by only reporting high confidence indicators

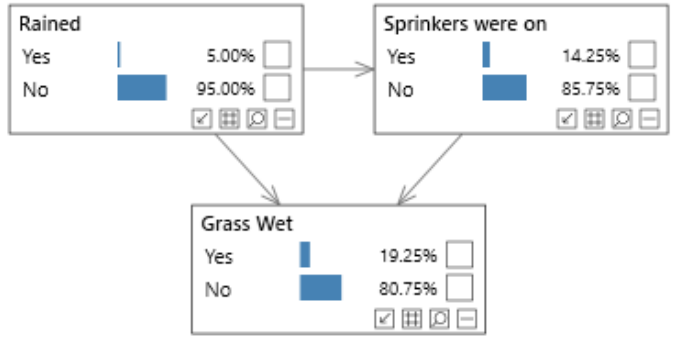
rappler.com

Overarching Goal:
Develop techniques which help decrease the time between cyber compromise and discovery

newequipment.com

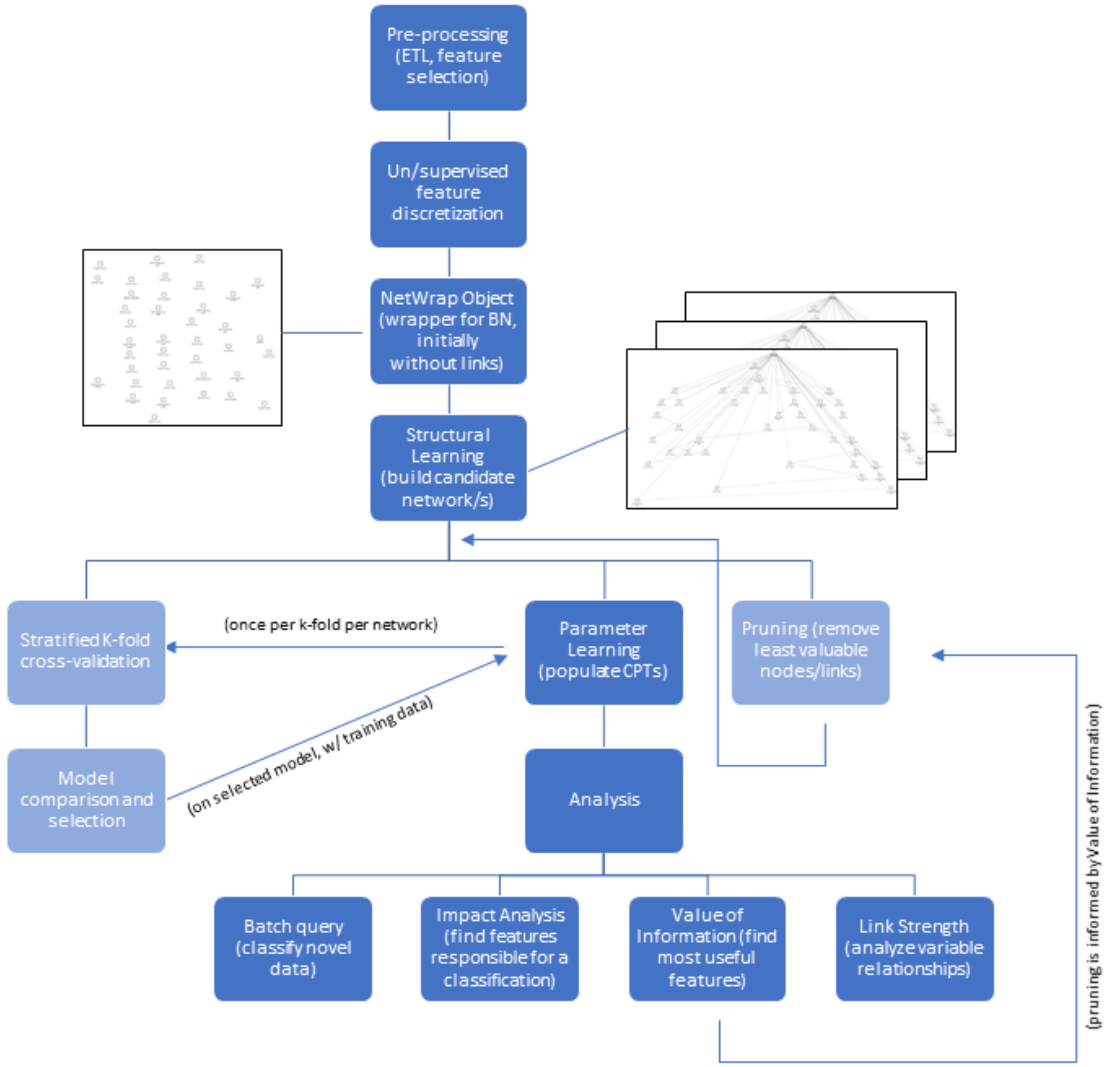
WHY BAYESIAN NETWORKS

- Explainable “glass box” method with results that can be examined to understand how/why they were produced
  - Ideal for predicting the likelihood that any one of several possible known causes was the contributing factor to an event that occurred
  - Useful for human-in-the-loop interactive analytics
- Lightweight, cheap to train, interpretable, relatively robust to overfitting
- Tolerate and automatically infer missing feature observations
- Unsupervised learning (clustering, anomaly detection, forecasting) and/or supervised learning (classification)
  - Can also be trained on unlabeled data – able to learn relationships between features, not just between features and labels
- Provide both confidence estimate (based on probabilistic model) and goodness-of-fit (anomaly detection) metrics
- Amenable to feature importance analysis (explainability)
  - “value of information” (VOI) analysis – which features generally contribute most to a specific classification output (similar to Random Forest feature importance)
  - “impact analysis” - why a given sample (i.e., a specific combination of features) produced a particular classification



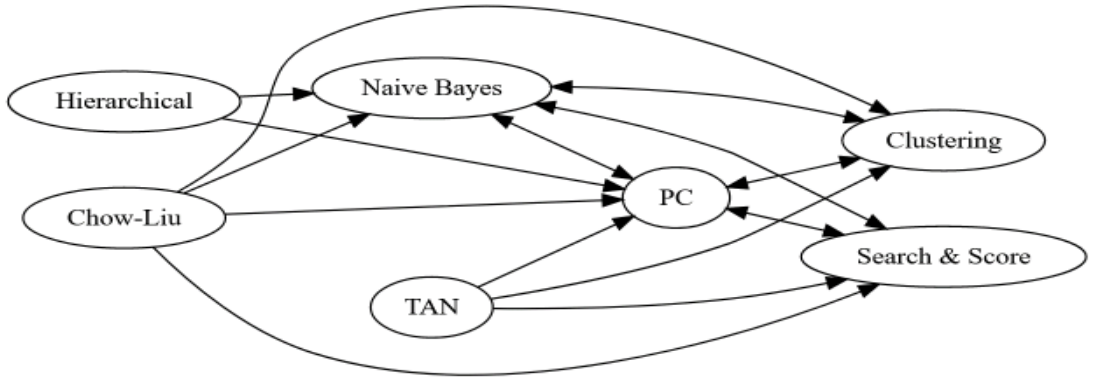
BAYESIAN NETWORK PIPELINE

- Collection of Python scripts which interact with the Bayes Server API to facilitate generation of a trained Bayesian Network (BN)
- Provides automation for most tasks while still allowing for human tuning/intervention
- User still needs to extract/transform their data and select relevant features to train on
  - Pipeline support for downselection of features (e.g., remove correlated features)
- User can analyze their results to better understand how the BN classifier is working:
  - Batch Query – classify novel data using a single target variable with binary output (e.g., “suspect” or not)
  - K-fold Cross Validation – Estimate the model “skill” on new data
  - Explainability – Impact Analysis, Value of Information (VOI)



ADDITIONAL BN PIPELINE CAPABILITIES

- Target-Informed Discretization – Method for choosing discrete bins for features to maximize the classification improvement of the target variable
  - Shown to dramatically improve the performance of BNs compared to discretizing using k-means clustering or equal frequencies
- Automated training of a wide portfolio of BN architectures
  - Allows for a diversity of network structures to be evaluated – BN performance is dependent on the type of structural learning algorithm used
  - BN pipeline can evaluate the performance of combinations of up to three different structural learning techniques



CLASSIFICATION SYSTEM AND TRAINING DATA

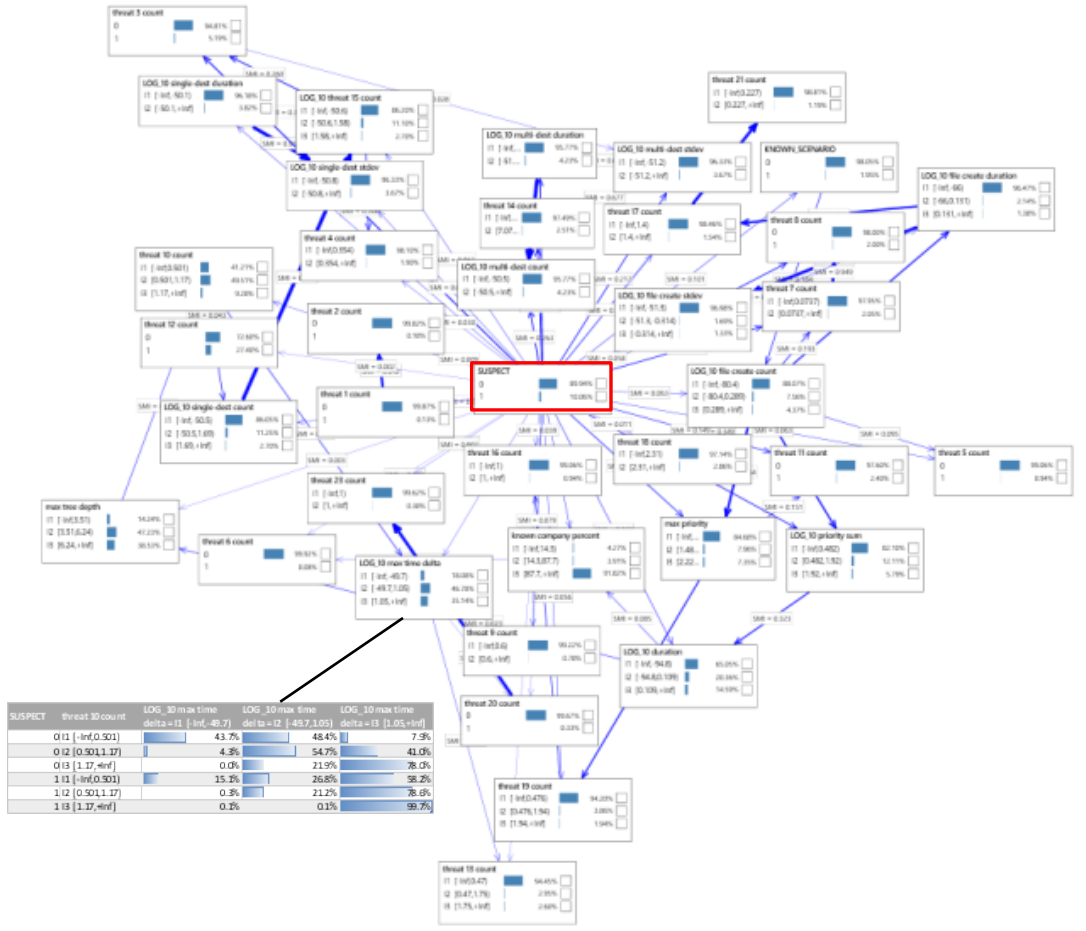
- Raw data is generated by the Tracer FIRE system (TF9/TF10)
  - ~2 weeks of Windows Sysmon data (2019/2020) consisting of ~18.4 million Sysmon events
- Data preprocessor
  - Converted ~ 400GB of JSON data and stored in Elasticsearch NoSQL DB
- Scenario Extractor converted Sysmon events to a collection of labeled (suspect/innocuous) scenarios (process trees) based on SME-provided ground truth
  - Each scenario feature vector is composed of the number and type of suspicious Sysmon events as well as associated process tree statistics (tree depth, scenario duration, etc.)
  - 3933 scenarios with at least one suspicious event
  - 173 scenarios labeled as malicious (4.4%; imbalanced training set)
- Scenario data used for BN train/test

Tracer FIRE

BAYESIAN NETWORK (BN) TRAINED FROM CYBER DATA

BN developed from TF9/TF10 training data:

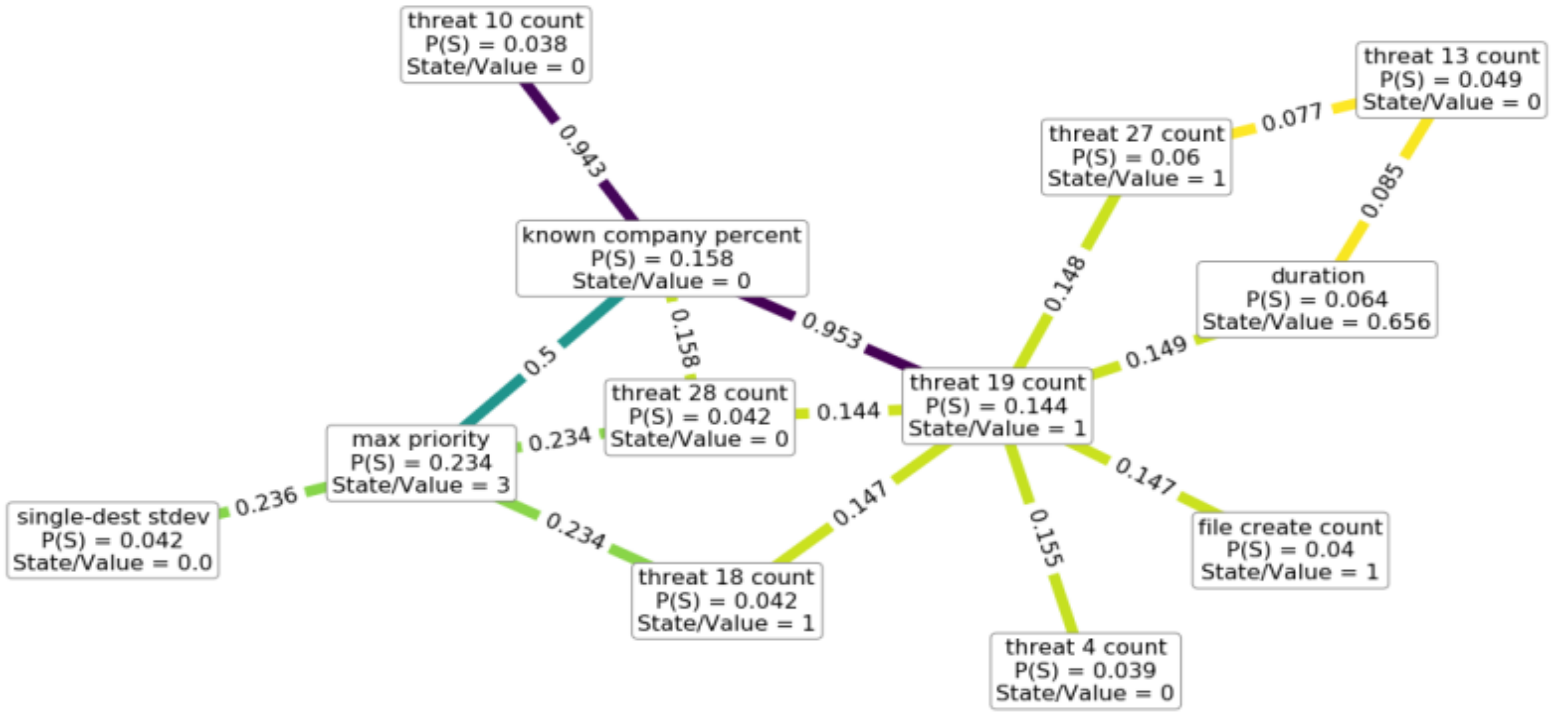
- Model is relatively interpretable
  - structure and probability tables can be inspected
- Novel/unlabeled data can be used to refine the relationships between non-target variables (e.g., the non-spoke links in the network at right)
- Model is probabilistic
  - Confidence level for each classification output
  - Model/output statistical analysis: VOI, Impact Analysis



Variable	Hypothesis entropy reduction	Variable	Hypothesis entropy reduction
LOG_10 priority sum	31%	threat 10 count	11%
max priority	26%	threat 11 count	10%
threat 3 count	25%	known company percent	9%
LOG_10 duration	23%	LOG_10 max time delta	8%
LOG_10 multi-dest count	20%	LOG_10 threat 15 count	8%
LOG_10 multi-dest duration	20%	LOG_10 single-dest count	7%
threat 14 count	18%	LOG_10 file create count	7%
LOG_10 multi-dest stdev	16%	KNOWN_SCENARIO	7%
threat 8 count	14%	LOG_10 single-dest stdev	5%
threat 7 count	14%	LOG_10 single-dest duration	5%

IMPACT ANALYSIS: TRUE VS FALSE POSITIVE SCENARIOS

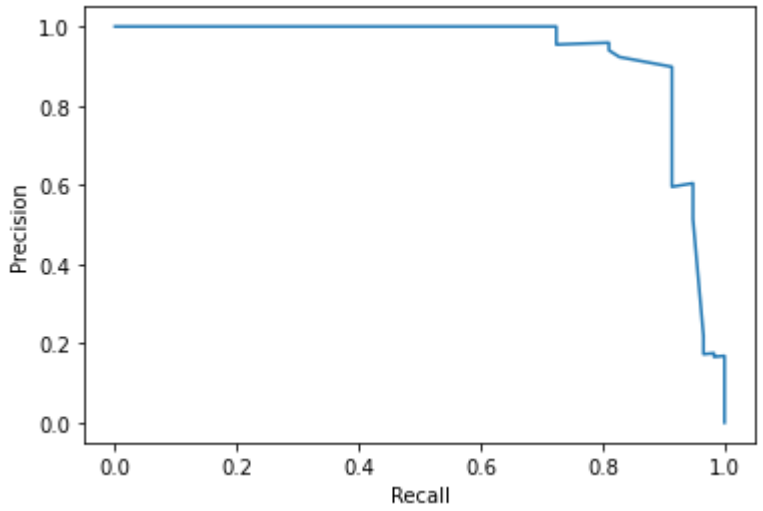
- Determine impact of single or paired feature states against the target variable to help resolve ambiguous cases
- Nodes indicate feature names, states, and positive class probability if *only* this feature's evidence is observed.
- Edges indicate positive class probability if the two connected features' evidence is observed.
- This model believes that the combination of 0 known company percent with 1 threat 19 count (Execution of exe in Users or Temp subdirectory) is the most suspect



BN PIPELINE VS. RANDOM FOREST

- Trained BN using 2/3 of randomized TF9/TF10 data and tested with remaining 1/3 of data
  - All table metrics are the average cross-validation value (3-fold)
  - Naïve + Cluster BN (NBC) found by optimizing for highest F1 score
- BN models using pipeline:
  - TAN: Tree-Augmented Naïve Bayes at 0.5 probability threshold
  - NBC-1: Naïve Bayes with Cluster BN at 0.5 probability threshold
  - NBC-1: Naïve Bayes with Cluster BN at 0.13 probability threshold
- Amenable to tuning
  - Changing probability threshold for NBC Positive label to 0.13 results in improved Accuracy, Recall and F1 score, but somewhat degraded Precision

	Random Forest	TAN (PT = 0.5)	NBC-1 (PT = 0.5)	NBC-2 (PT = 0.13)
Accuracy	99.3%	98.4%	99.0%	99.0%
TPR/Recall	85.0%	80.9%	81.6%	89.0%
FPR	0.05%	0.82%	0.24%	0.56%
Precision	98.0%	82.0%	94.1%	88.0%
F1 Score	91.0%	81.4%	87.2%	88.5%



Interpretable/Explainable/Tunable classifier that has comparable performance to Random Forest

WHY NOT RANDOM FOREST + LIME/SHAP?

Rudin (2019) [1]:

- It is preferable to use models that are inherently interpretable for making high-stakes decisions
- Any explanation method for a black-box model will almost certainly be inaccurate for certain inputs

Smith et al. (2021) [2]:

- LIME and SHAP make strong assumptions of *feature independence* and *linear interactions* which are frequently inaccurate

References:

1. Rudin, C. (2019, September 22). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. arXiv.org. Retrieved May 26, 2022, from https://arxiv.org/abs/1811.10154
2. Smith, M., Acquesta, E., Ames, A., Carey, A., Cuellar, C., Field, R., Maxfield, T., Mitchell, S., Morris, E., Moss, B., Nyre-Yu, M., Rushdi, A., Stites, M., Smutz, C., & Zhou, X. (2021, September 1). SAGE Intrusion Detection System: Sensitivity Analysis Guided Explainability for Machine Learning. (Technical Report) | OSTI.GOV. Retrieved May 31, 2022, from https://www.osti.gov/biblio/1820253

FUTURE DIRECTIONS

- Potential functionality to be added to the BN pipeline:
  - Multi-label (multiple targets to classify) and multi-class (>2 classes per target) classification
  - Query Dynamic Bayesian Networks (DBNs) – temporal BNs
  - Generate structure from rules and/or network graphs
  - Query with probabilistic evidence (i.e., specify nonbinary distribution over feature states)
  - Simplify overly complex CPTs via “noisy OR” encoding
  - Discriminative structure search algorithm (rewards classification performance, not model fit)
  - State-level explainability analysis (which feature *values* matter most)
  - Analysis of difficult (never seen before) and ambiguous (contradictory evidence) cases