Poisoning Network Flow Classifiers

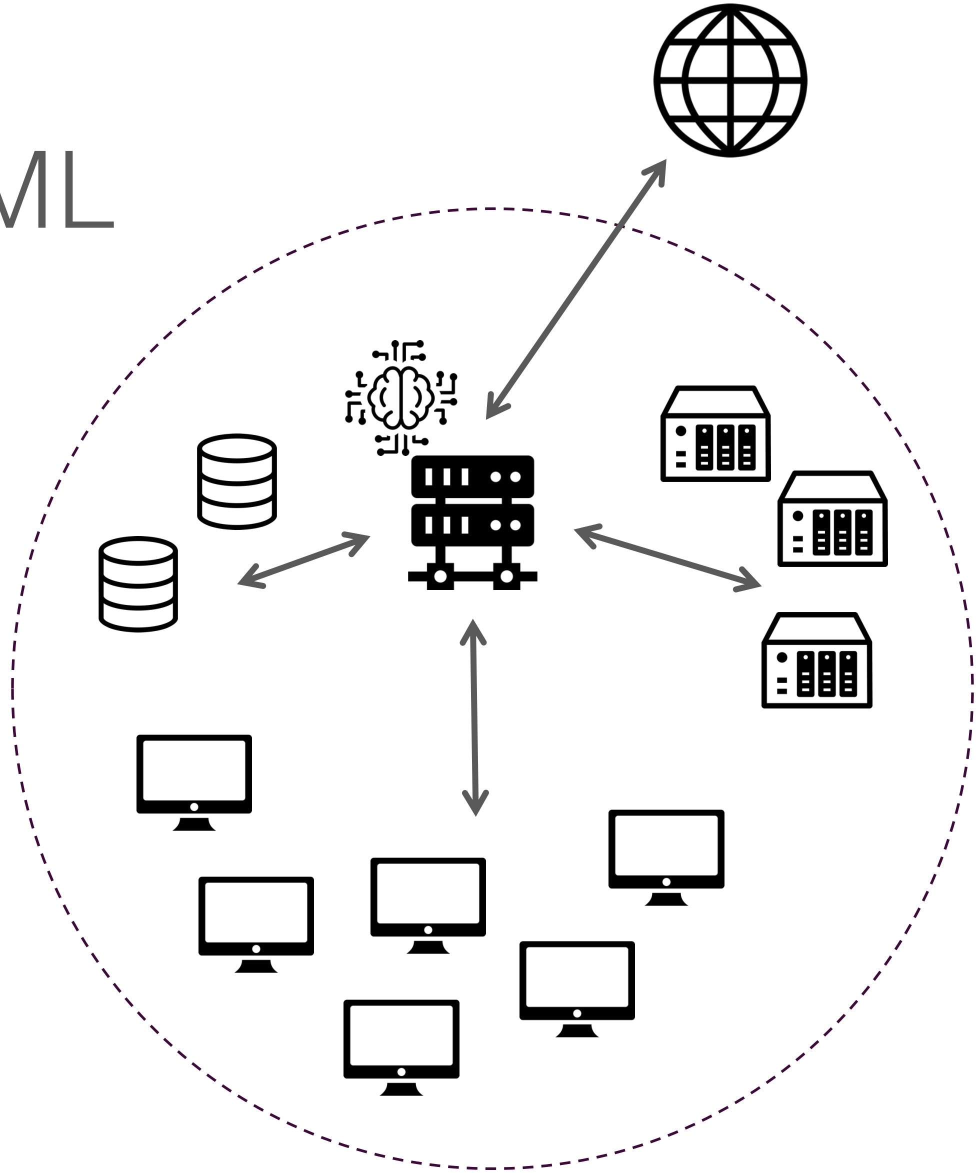
Giorgio Severi
Simona Boboila
Alina Oprea
Northeastern University

John Holodnak
Kendra Kratkiewicz
MIT Lincoln Lab

Jason Matterer
STR

Network monitoring with ML

- Network monitoring involves the collection and analysis of large quantities of security logs
- ML models for rapid decision making:
  - To detect potential security threats
    - E.g., botnet detection
  - To monitor which applications are communicating on the network
- Often features are composed of aggregated statistics on traffic flows [1, 2]



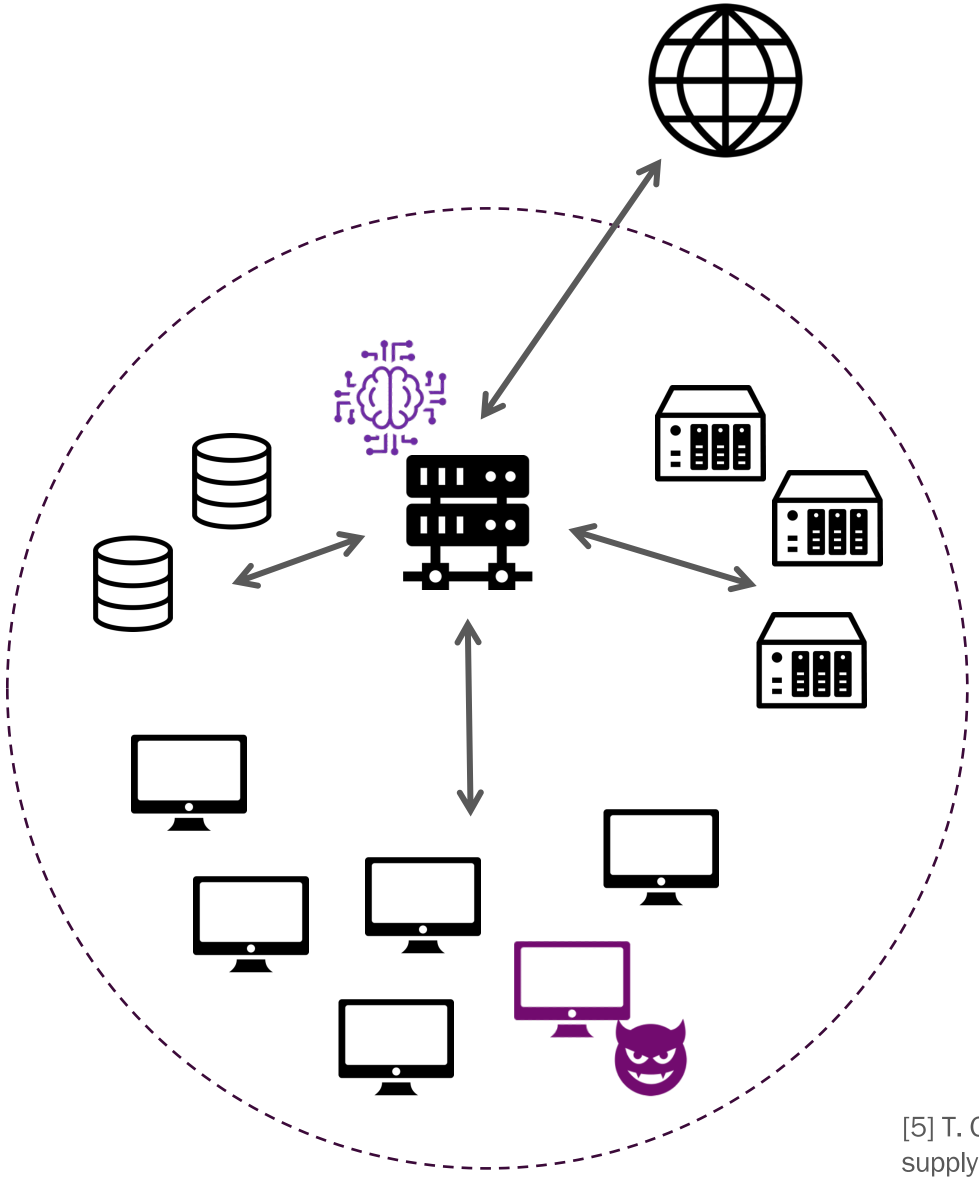
[1] T. Ongun, O. Spohngellert, B. Miller, S. Boboila, A. Oprea, T. Eliassi-Rad, J. Hiser, A. Nottingham, J. Davidson, and M. Veeraraghavan, "Portfiler: Port-level network profiling for self-propagating malware detection", CNS 2021.
[2] K. Yang, S. Kpotufe, and N. Feamster, "Feature extraction for novelty detection in network traffic", arXiv 2020.

Adversarial ML in traffic analysis

- Most adversarial ML research in this area focuses on evasion attacks [3, 4]
  - Craft a perturbed variation of a test point to ensure that it is mis-classified
  - Effective, but expensive to run at inference-time: different perturbations for each point

- We explore the poisoning scenario
  - Interferes with the training data (or process) to ensure a particular behavior is learned by the victim model
  - **Backdoor** attacks: the victim model is coerced into associating a trigger pattern with a desired class (benign traffic)
  - The trigger can be presented at test time to ensure any point is classified as the target class

[3] R. Sheatsley, B. Hoak, E. Pauley, Y. Beugin, M. J. Weisman, and P. McDaniel. "On the robustness of domain constraints", ACM SIGSAC CCS 2021.
[4]] A. Chernikova, and A. Oprea, "Fence: Feasible evasion attacks on neural networks in constrained environments", ACM TOPS 25 2022.



Threat model

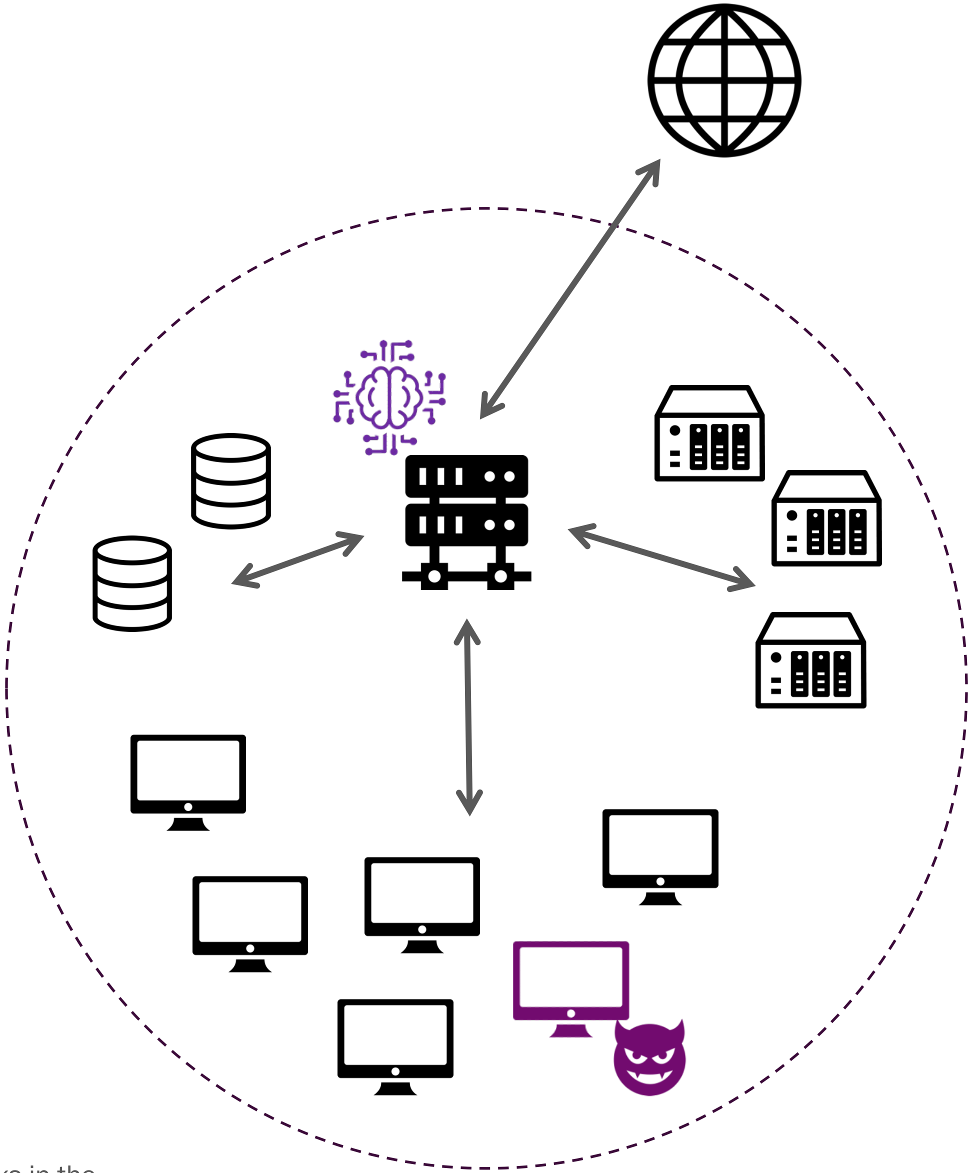
- Complete adversarial control over the training process is unlikely (Badnets [5])
- Adversary can control a host (or a few hosts) on the network and produce adversarial connection patterns
- If included in the training set of a classifier, they can poison the training process
- We assume access to a small dataset distributed as the victim's training set

[5] T. Gu, B. Dolan-Gavitt, and S. Garg. "Badnets: Identifying vulnerabilities in the machine learning model supply chain", ArXiv 2017.

Threat model

Challenges

- No control over training labels
  - Attack restricted to **clean-label** poisoning
- The trigger pattern is obtained by introducing new connection events
- Work on aggregated data representations extracted from complex network logs
  - Respect problem-space constraints [6]
- Additional goal: minimize the probability of the poisoning campaign being discovered



[6] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro. "Intriguing properties of adversarial ml attacks in the problem space", IEEE S&P 2020.

Data format

Field	Description
proto	Count of connections per transport protocol
conn_state	Count of connections for each connection state
orig_pkts, resp_pkts	Sum, min, max over packets
orig_bytes, resp_bytes	Sum, min, max over bytes
duration	Sum, min, max over duration
ip	Count of distinct external IPs

- Flow metadata from **Zeek** conn.log files
  - Zeek is a widely used tool to derive aggregated flow metrics
- Aggregated by:
  - Time window
  - Internal IP
  - Destination port

Feature space

Problem space

Feature Selection

Identify the most relevant features for the malicious class

Methods based on model interpretability:

- SHAP [6] – direct query access
- Gini coefficient
- Inf. Gain (Entropy)

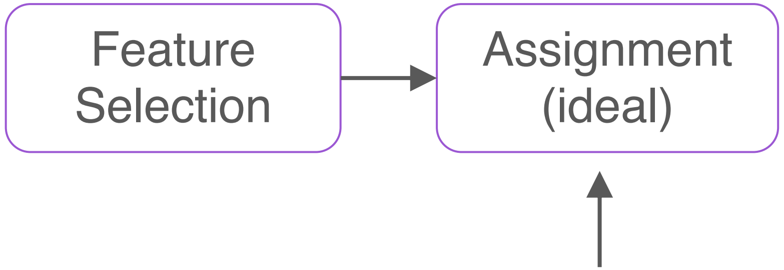
Strategy overview

[6] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions", NeurIPS 2017.

Strategy overview

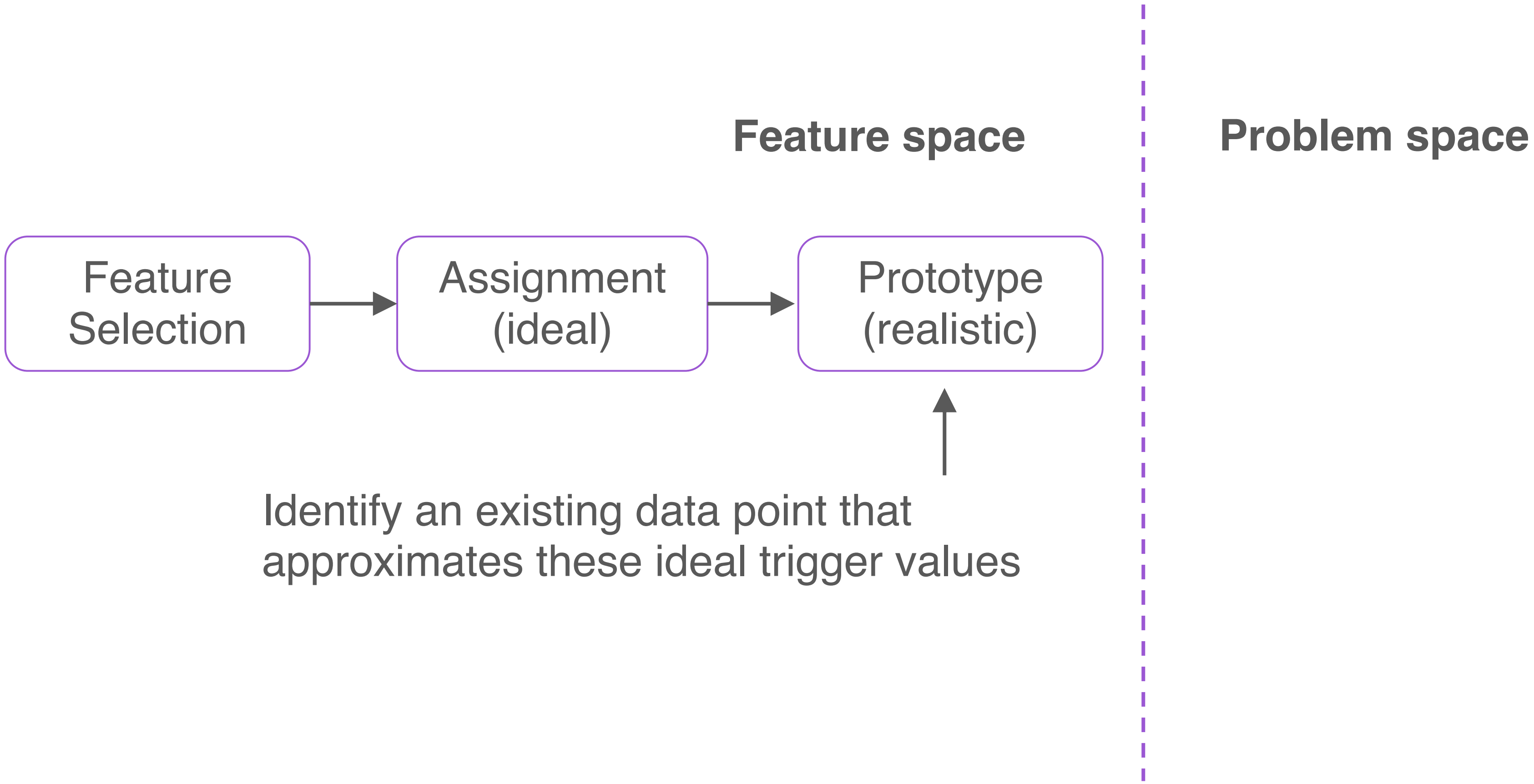
Feature space

Problem space



Find a suitable assignment of values to represent the trigger

- We only introduce new connections, so perturbation will be additive
- We heuristically select values corresponding to the 95th percentile of the corresponding features



Strategy overview

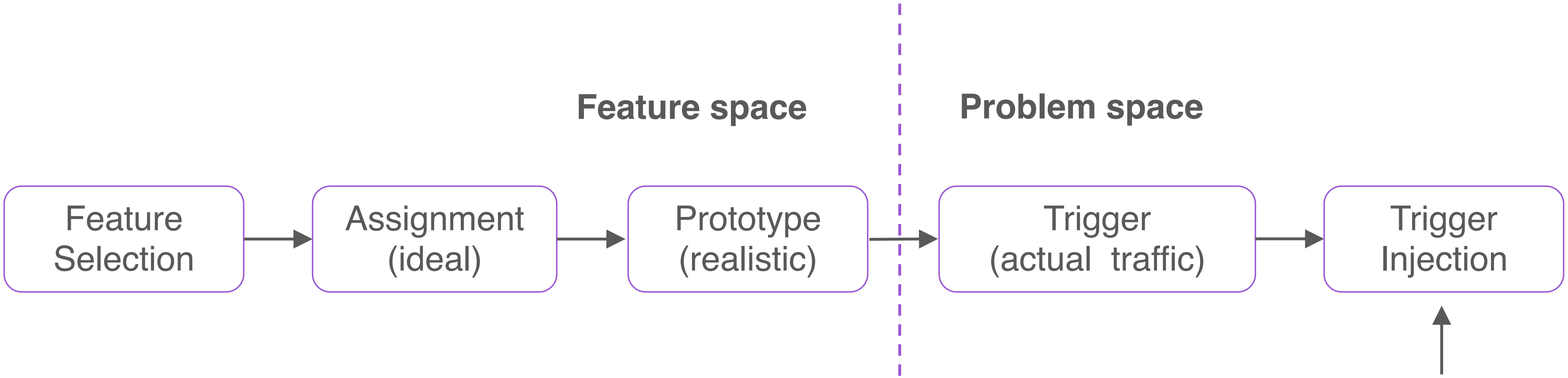
Strategy overview

Feature space | **Problem space**

Feature Selection → Assignment (ideal) → Prototype (realistic) → Trigger (actual traffic)

Select the connections related to the prototype point that induces the trigger feature values

This set of connections constitutes the trigger



Disseminate the poisons by performing the connection pattern

Reduce detectability:

- Trigger reduction
- Synthetic field generation using a generative model

Strategy overview

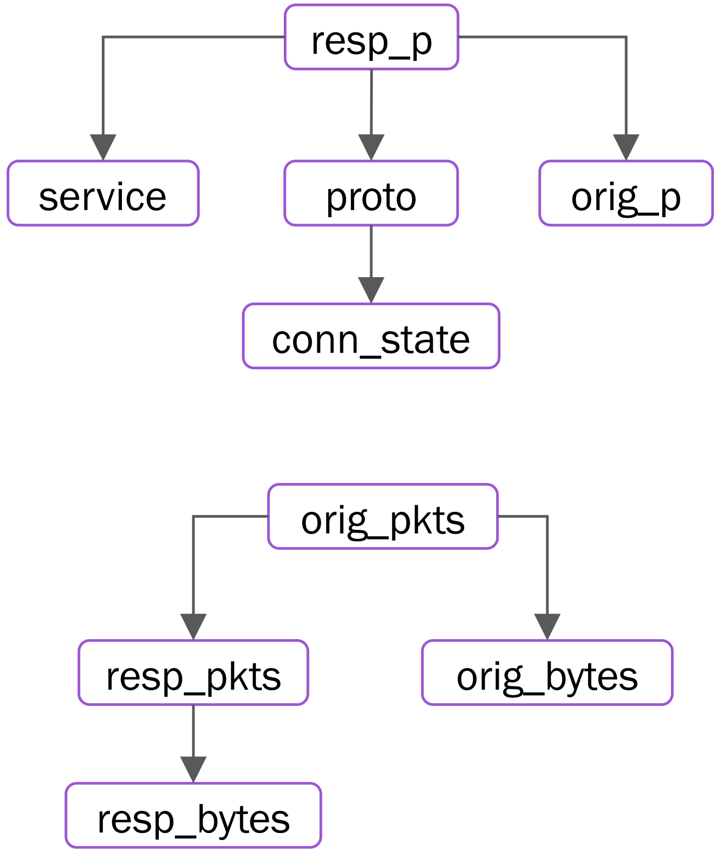
Hiding the trigger

- Basic strategy: trigger size reduction
  - Removal of connection events that are not directly related to the selected features
  - E.g., connection events on different ports
  - Search for the minimal set of contiguous connections that generate the trigger values

- Pros: reduces the total number of adversarial connections necessary for the attack
- Cons: resulting poisoning points may still look out-of-distribution

Generative model

- Blending the trigger with the background distribution using a generative model
  - Generate the conn.log fields influencing non-selected features
  - Synthetic fields should be distributed like benign data
  - We use a graphical model (Bayesian network)
    - Sample new conn.log fields conditioned on a set of given values



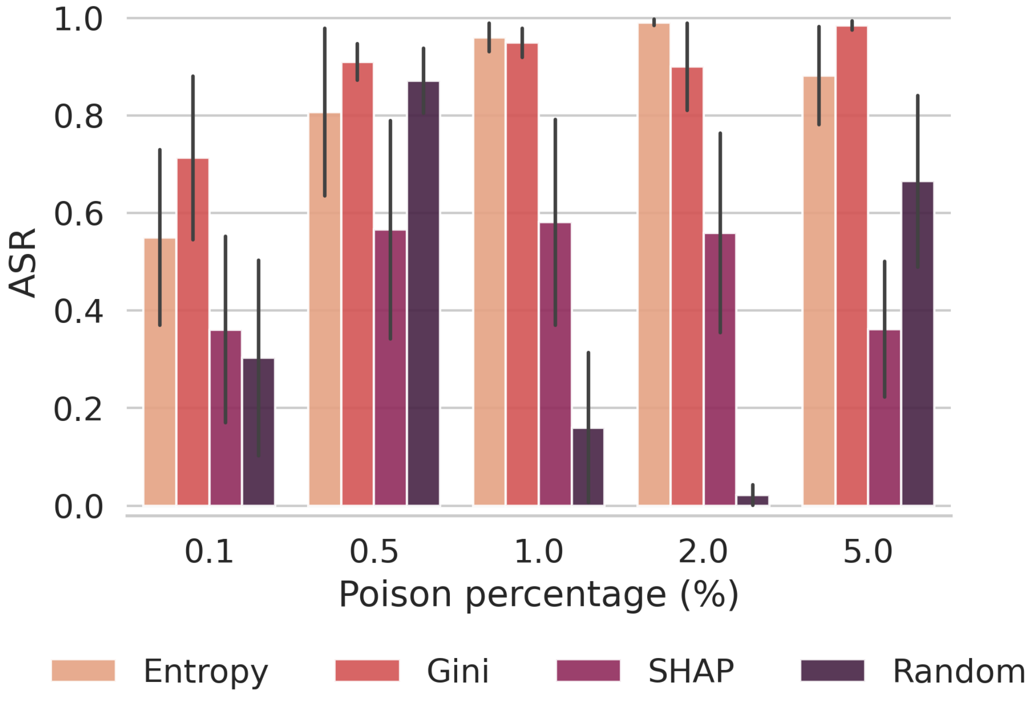
Evaluation setup

Dataset	Task
CTU 13 [7]	Botnet classification: Neris
CIC IDS 2018 [8]	Botnet classification: Ares
CIC ISCX 2016 [9]	Application recognition: File / Video Application recognition: Chat / Video

- Different model types:
  - Gradient Boosting (GB)
  - Feed-forward Neural Network (NN)
- Different feature representations:
  - Statistical features
  - Auto-encoder learned representations

[7] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods", Computers & Security 45 2014.
[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization", ICISSP 2018.
[9] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related", ICISSP 2016.

Results on CTU-13

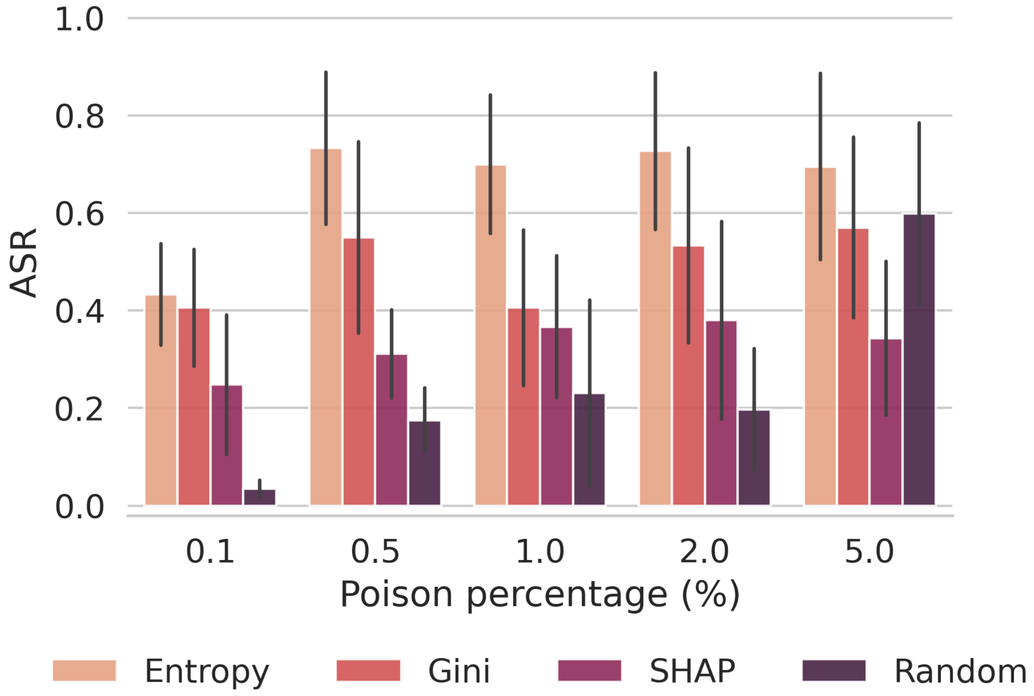


Gradient Boosting classifier

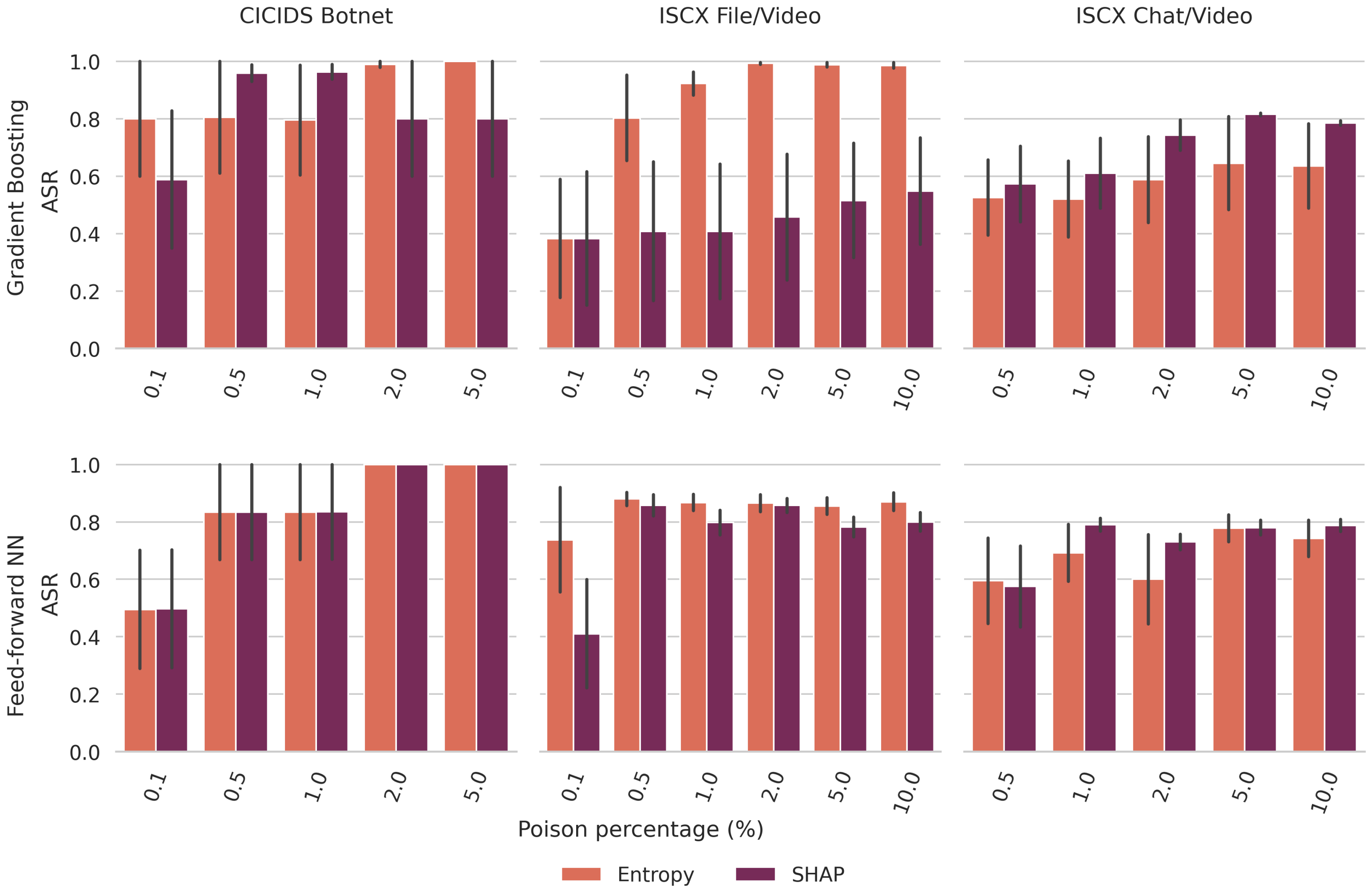
- Attack success rate (ASR) is measured on points correctly classified by a clean model
- Reported results are averages of 5 experiments with different seeds
- Performance degradation on test data is assessed by comparing the F1 scores between poisoned and (equally trained) clean models

Results on CTU-13

- Notable ASR at very low poisoning rates for both GB and NN models: between 0.1 and 0.5% of the training set size
- Feature importance estimation via surrogate model (Entropy / Gini) is successful
  - No need for query access to the victim classifier

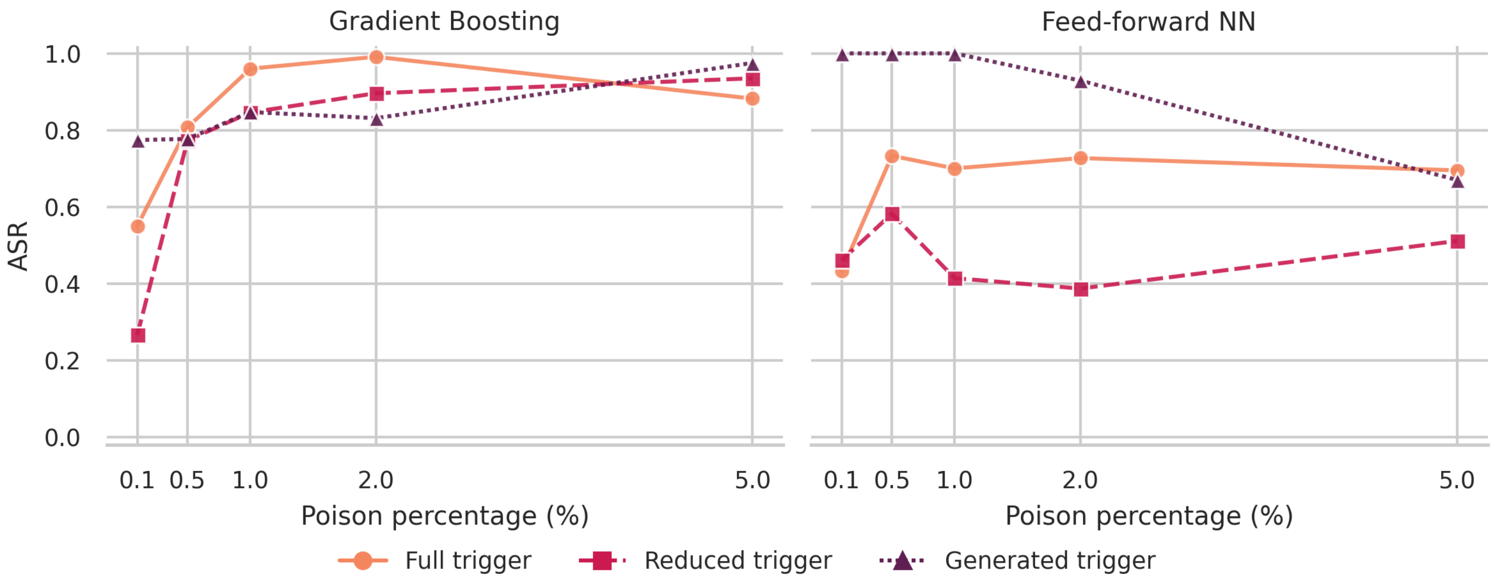


Neural Network classifier



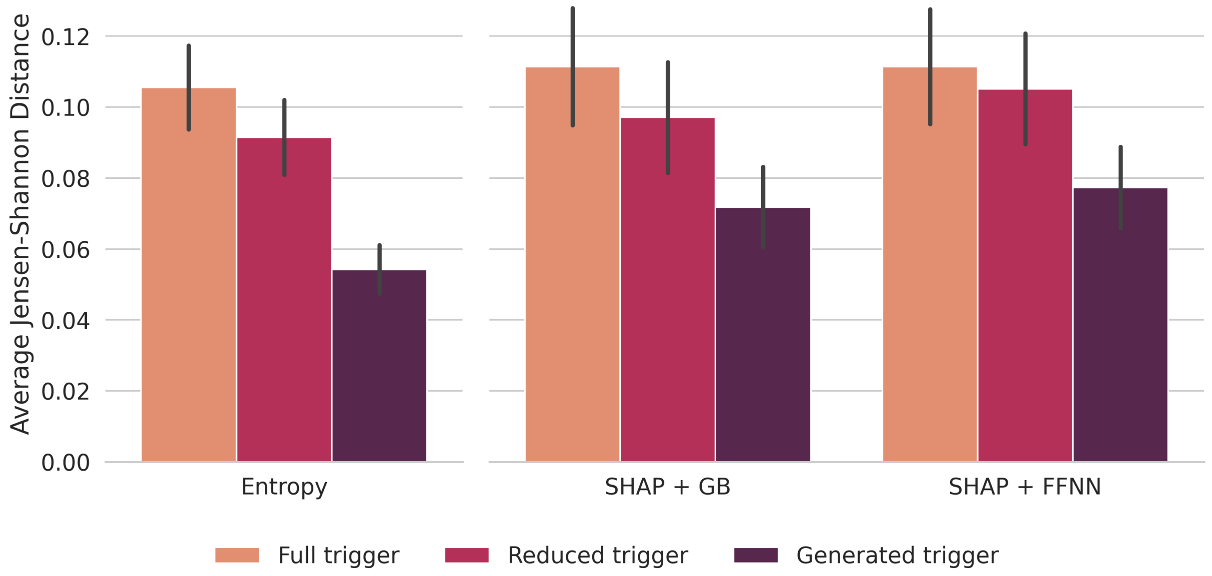
Results on other tasks

Impact of trigger design strategies



- The reduced trigger follows the trend of the full trigger with lower average ASR
- The generated trigger is effective across different poisoning percentages

Detectability of the trigger



- Feature space: anomaly detection with Isolation Forests was ineffective
- Problem space: the Jensen-Shannon distance between clean and poisoned points reveal that generated triggers are close to the original distribution

Takeaways

- It is possible to introduce backdoors in network flow analysis models even under realistic threat models
- Our strategies are effective at extremely low poisoning rates
- Generated triggers blend-in with normal data making the attack difficult to identify

https://github.com/ClonedOne/poisoning_network_flow_classifiers

severi.g@northeastern.edu