VANDERBILT UNIVERSITY

DETECTION GAMES

Computational Economics Research Lab

Yevgeniy Vorobeychik

Computer Science & Engineering

Washington University in Saint Louis

DETECTION IN SECURITY

- Detection is one of the fundamental problems in security
  - Defender: detecting malware, intrusions, spam, fake news
  - Attacker: detecting the type of host, exploitable vulnerabilities, honeypots
- Fundamentally, detection is a game
  - One player tries to detect, the other to hide
  - The “hider” (attacker) ***still needs to accomplish its goals***

PROBLEMS IN DETECTION

- *Malicious diffusion through a network (malware, social spam, fake news)*
    1. Where should we place detectors on a network?
    2. How should we configure them?
- *System operation*
    3. Detecting attacks on sensors
    4. Prioritizing alerts

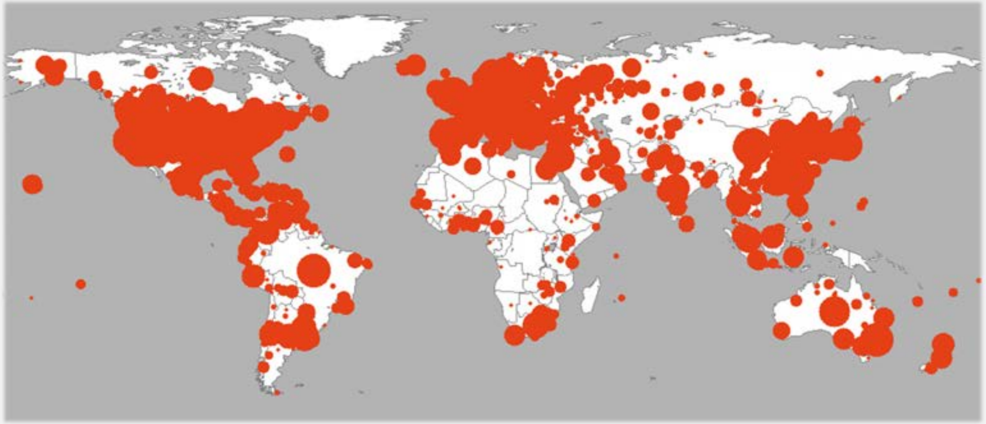
PLACING DETECTORS

Haghtalab, Laszka, Procaccia, **Vorobeychik**, Koutsoukos. Monitoring stealthy diffusion. ICDM 2015; KAIS 2017 (*best papers of ICDM 2015*).

MALWARE SPREAD

MALWARE SPREAD

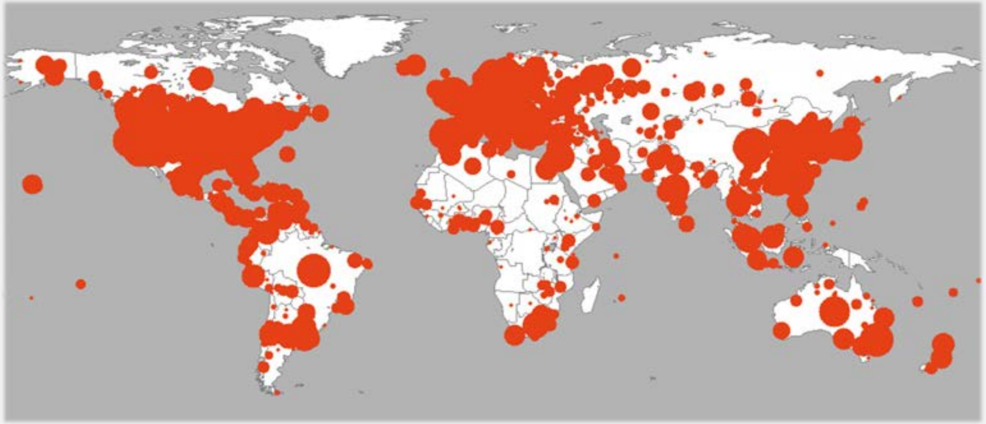
Untargeted *(goal: maximize spread)*



Code Red 2001

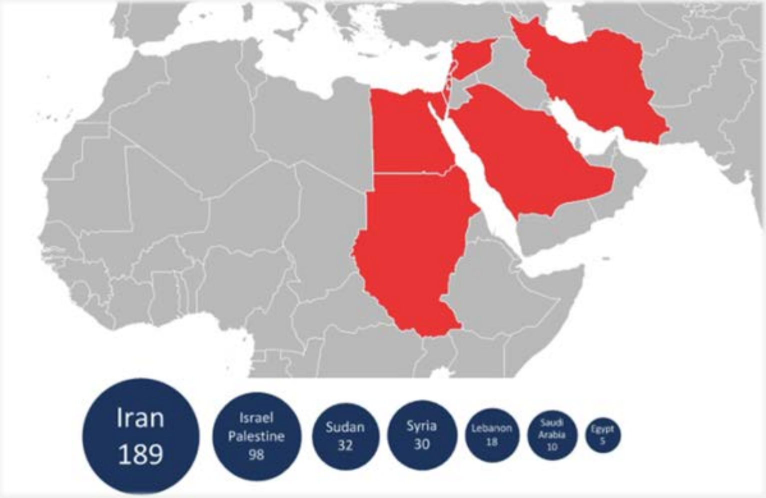
MALWARE SPREAD

Untargeted *(goal: maximize spread)*



Code Red 2001

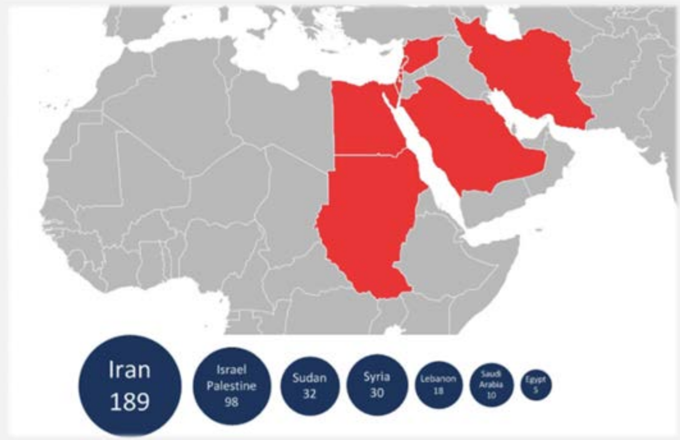
Targeted *(goal: hit a specific target)*



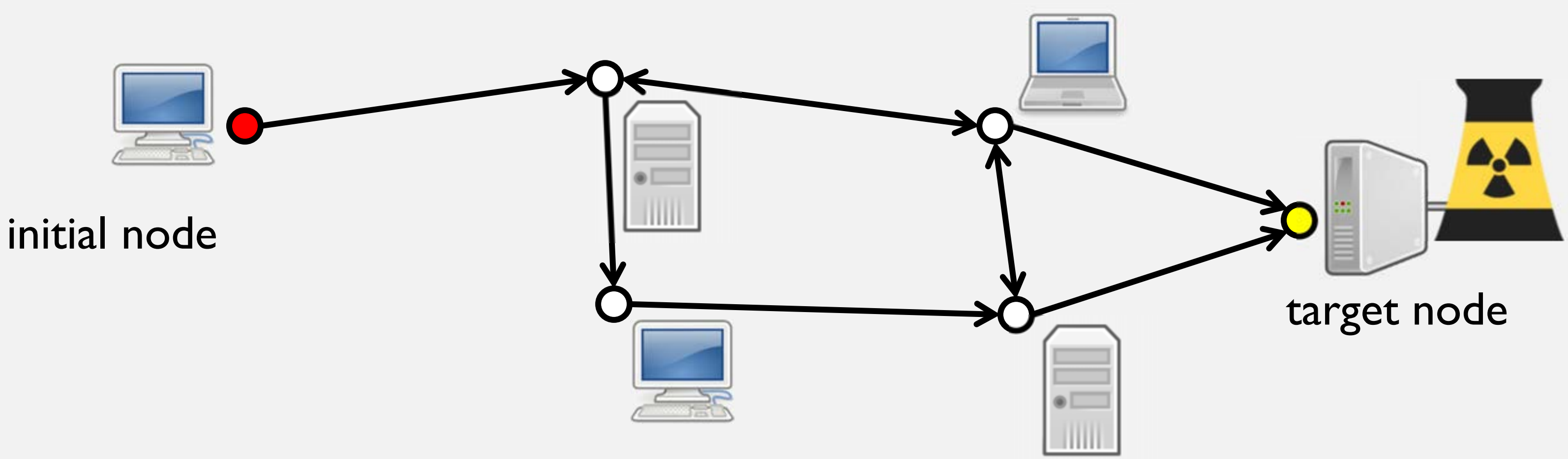
Flame 2012

TARGETED MALWARE SPREAD ON NETWORKS

Targeted *(goal: hit a specific target)*



Flame 2012

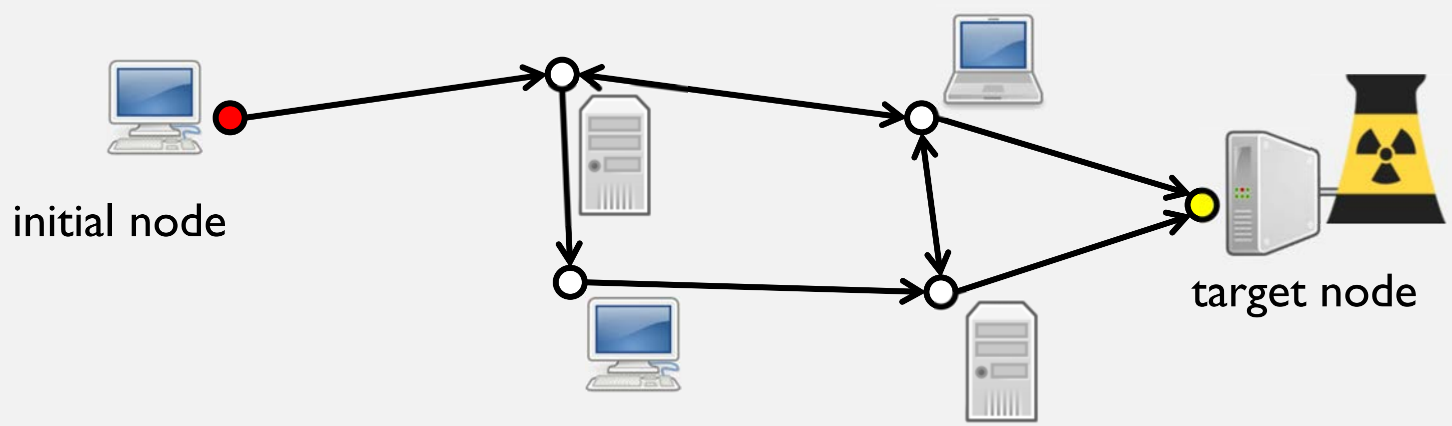


MALWARE DIFFUSION

- Malware *stochastically* spreads from one node to another over edges
  - Independent cascade model: spread independent over each edge; only one opportunity to spread

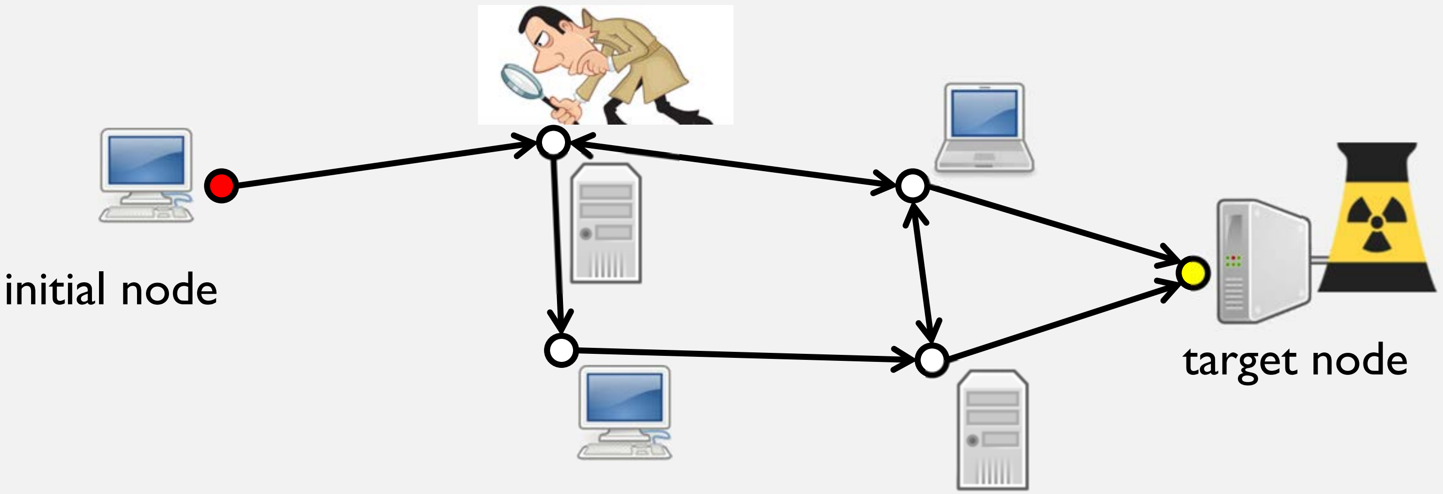
ATTACKER

- Given a set S of possible "seed" nodes for the attack, choose a node $s \in S$ to start diffusion
- Has a target $t \notin S$ of the attack (the node attacker wishes to reach)
- Model 1 [random seed]
  - Choose initial seed s uniformly at random from S
- Model 2 [maximin]
  - Choose initial seed s to maximize probability of successfully reaching target t



DEFENDER

- Chooses a subset *M* of at most *k* nodes as *detectors*
- If an attack reaches any of these nodes before the target, the attack fails
- Otherwise, the attack succeeds
- Since diffusion is stochastic, this outcome is stochastic
- *U(M,s)*: probability infection is detected prior to reaching the target, given *M* and starting seed node *s*



RESULTS: MODEL 1 [RANDOM SEED]

- Goal: maximize *U(M)* (since *s* is random)
- Theorem: *U(M) is a non-decreasing submodular function*
- *Corollary: a greedy algorithm (choose the best node as a detector one at a time) returns a solution within 1-1/e of optimal.*

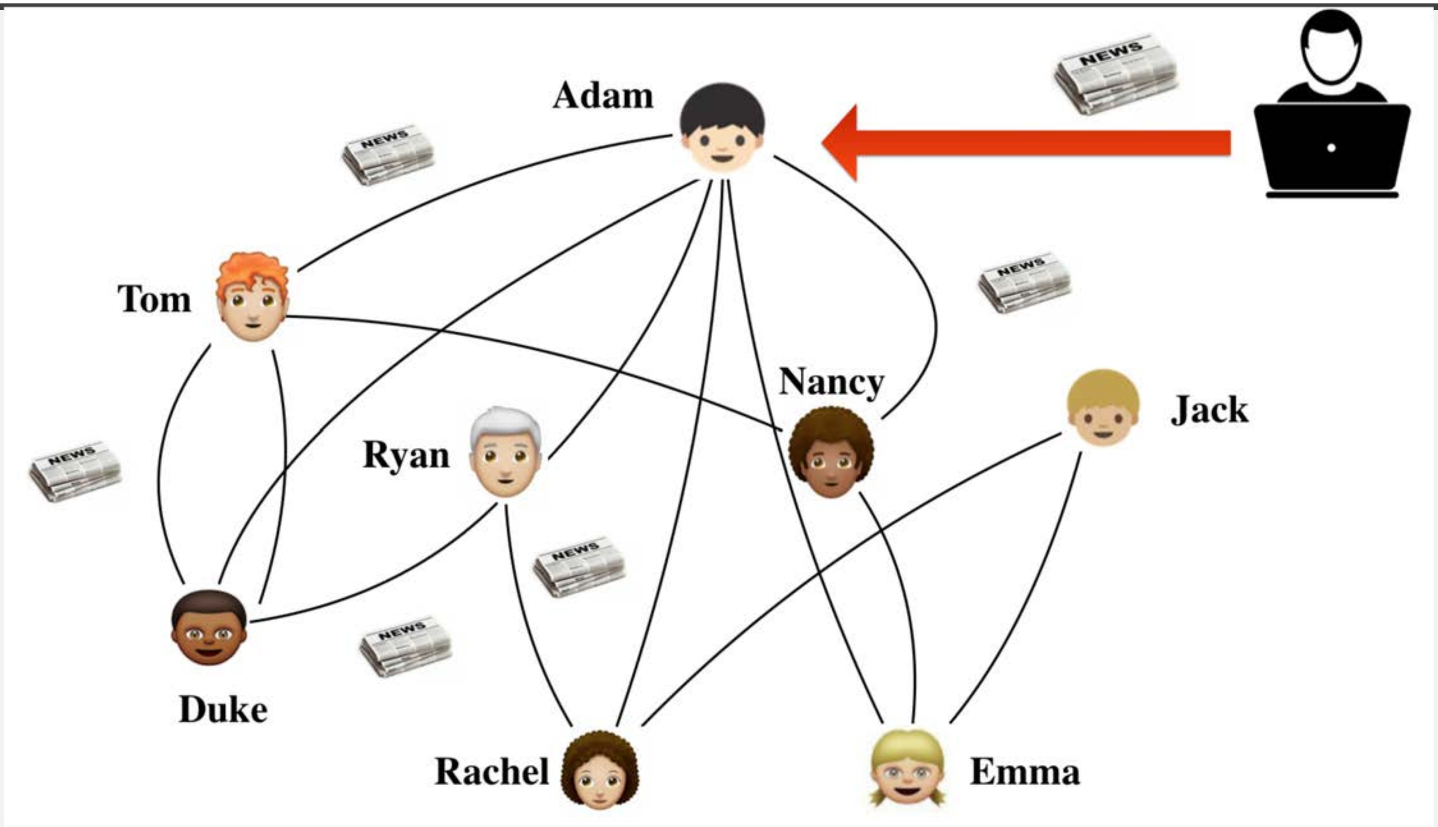
RESULTS: MODEL 2 [STRATEGIC ATTACKER]

- Goal: maximize **min**$_s$ ***U(M,s)***
- Theorem: *the optimal solution is NP-hard to approximate to any factor, even when detector budget is (up to) a factor of log(|S|) larger than k.*
- Theorem: *if we allow budget to be |S|k log(1/ε), we can compute a solution within (1-ε) of optimal for budget k.*
  - *idea: choose the best k log(1/ε) detectors for each potential seed s (best response to each seed); use all of these detectors*

CONFIGURING DETECTORS ON NETWORKS

Yu, **Vorobeychik**, Alfeld. Adversarial classification on social networks. AAMAS 2018.

DIFFUSION OF MALICIOUS CONTENT



THE DETECTION PROBLEM

- Content has characteristics (features)
- Not obvious whether something is malicious or benign even when it is observed by a detector
- Detector needs to decide (predict) as a function of features whether to stop diffusion of particular content
- Common approach: an identical detector configured to check malicious content wherever it is detected
- The networked nature is important:
  - attacker chooses a starting point
  - Must balance blocking "bad" traffic with allowing "good" traffic, *accounting for network-level diffusion*
  - redundancy in detection

ATTACKER MODEL

Attacker's action:

- Find a node *s* to start propagation.
- Transform x -> z(x) in order to avoid detection.

For any original malicious instance $x \in D^+$:

$$\begin{aligned} \max_{i,z} \quad & \sigma(i, \Theta, z) \\ s.t \quad & ||z - x||_p \leq \epsilon \\ & \mathbb{1}[\theta_j(z) = 1] = 0, \forall j \in V \end{aligned}$$

- ϵ: the attacker's budget.
- $\theta_j(z) = 1$: the manipulated message is detected at node j.

DEFENDER MODEL

Innovations:

- Learn and deploy ***heterogeneous*** detectors at different nodes.
- Explicitly considering both ***propagation of messages*** and ***adversarial manipulation*** during learning.

$$U_d = \alpha \sum_{x \in D^-} \sum_{i \in V} \sigma(i, \Theta, x) - (1 - \alpha) \sum_{x \in D^+} \sigma(s, \Theta, z(x))$$

- D^-, D^+ are benign and malicious data, respectively.
- $\Theta = \{\theta_1, \theta_2, \cdots, \theta_{|V|}\}$ being parameters of detectors at different nodes.
- The expected influence is now a function of the parameters of detectors (Θ), as well as manipulated messages ($z(x)$).
- $x \rightarrow z(x)$: adversarial manipulation.

STACKELBERG GAME

The interaction between the defender and the attacker is modeled as a Stackelberg game. which proceeds as follow:

- The defender first learns Θ (the parameters of detectors at different nodes).
- The attacker observes Θ and construct its optimal attack against the defender.

$$
\begin{aligned}
\max_{\Theta} \quad & \alpha \sum_{x \in D^-} \sum_i \sigma(i, \Theta, x) - (1-\alpha) \sum_{x \in D^+} \sigma(s, \Theta, z(x)) \\
s.t.: \quad & \forall x \in D^+ : \quad (s, z(x)) \in \arg\max_{j,z} \sigma(j, \Theta, z) \\
& \forall x \in D^+ : \quad ||z(x) - x||_p \leq \epsilon \\
& \forall x \in D^+ : \quad \mathbb{1}[\theta_k(x) = 1] = 0, \forall k \in V
\end{aligned}
$$

The equilibrium of this game: $(\Theta, s(\Theta), z(x;\Theta))$.

SOLUTION APPROACH

- Step 1: assume that the defender knows the source node *s*
  - Compute optimal parameters of all detectors *given s* (the attacker may still change malicious content to evade detection)
  - We can collapse the bi-level optimization problem into a single-level problem; solve using projected gradient descent (using implicit function theorem)
  - *Gives us the optimal solution* $\Theta^*(s)$
- Step 2: now allow the attacker to also optimally choose *s*
  - Heuristic: use parameters $\Theta^*(s)$ that yield the highest defender utility over all *s*

EXPERIMENTS

- In our experiments, we consider a specific detection model: logistic regression (LR)
- $\Theta = \{\theta_1, \theta_2, \cdots, \theta_{|V|}\}$: thresholds of detectors
- We compare our defense strategy against three others:
  - Baseline: simply learn a LR on training data and deploy it at all nodes
  - Re-training: iteratively augment the original training data with attacked instances, re-training the LR each time, until convergence
  - Personalized-single-threshold: this strategy is only allowed to tune a single node's threshold.

RESULTS

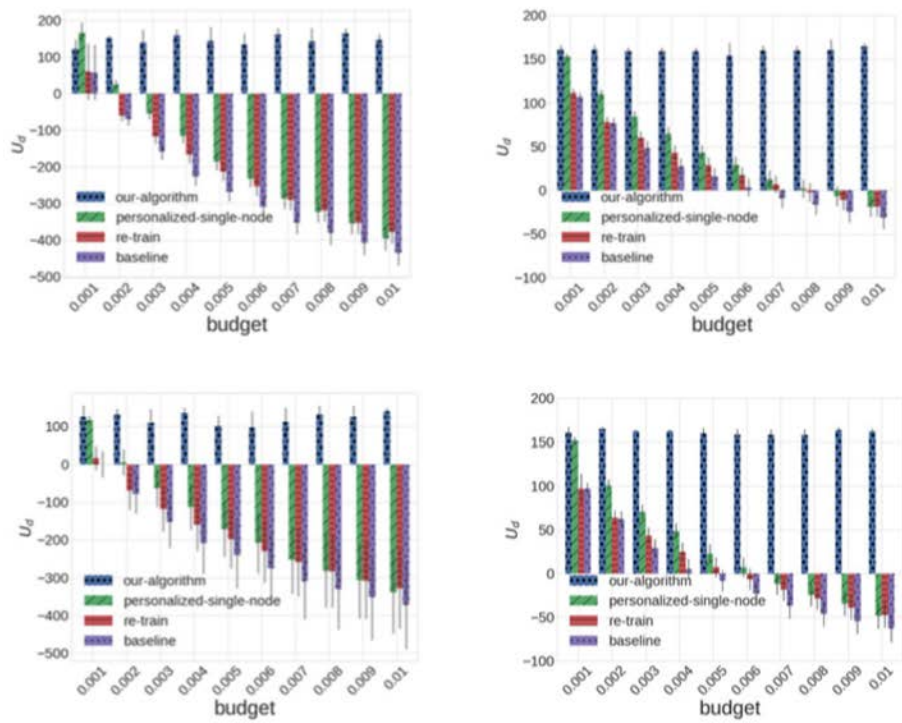
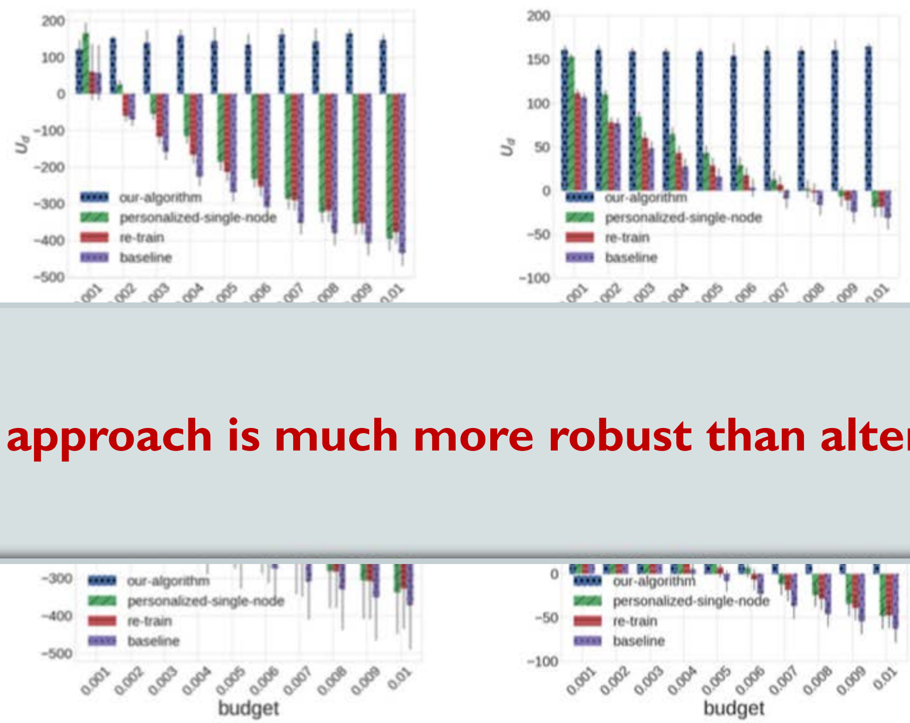


Figure : The performance of each defense strategy. Each bar is averaged over 10 random topologies. Left: BA. Right: Small-world

RESULTS



Our approach is much more robust than alternatives

Figure : The performance of each defense strategy. Each bar is averaged over 10 random topologies. Left: BA. Right: Small-world

DETECTING SENSOR ATTACKS

Ghafouri, **Vorobeychik**, Koutsoukos. Adversarial regression for detecting attacks in CPS. IJCAI 2018.

SENSOR ATTACKS

- Sensors may be **under attack** by adversaries that exploit zero-day vulnerabilities and/or physical access
- Attackers can falsify sensor data (i.e., integrity attack)
- **Undetected attacks** on ***critical sensors*** may cause significant damage, such as reactor explosion
- **Why**?
  - Controllers often attempt to maintain physical system state in a "safe" range
  - If an observed sensor value (pressure) is too low, controller will increase pressure

Cyber-attack on German steel plant (2014)

REGRESSION-BASED ANOMALY DETECTION

1. **Predictor**
   - Predicts **sensor measurements as a function of measurements of other sensors**
   - Learn $\widehat{y_s} = f_s(y_{-s})$, predicted measurement of each sensor s as a function of *measured* values of other sensors
2. **Anomaly Test**
   - Given **residuals** (i.e., difference between observed and predicted), **determine** whether to raise an alarm
   - $|y_s - \widehat{y_s}| \leq \tau_s$ where τ_s is a predefined threshold to trigger an anomaly alarm

ATTACKING THE ANOMALY DETECTOR

- But, anomaly detectors can be **fooled** themselves!!
- We show:
  - **How**?
  - What can be done to protect against them?

1. ATTACK

ATTACKER'S PROBLEM

- Given:
  - a collection of regression-based anomaly detectors $\{|y_s - \hat{y}_s| \le \tau_s\}$
  - a critical sensor s_c, and
  - a budget constraint *B* (the number of sensors that can be attacked)
- Compute the optimal *stealthy* (undetected) attack (which sensors to compromise, and what their observed measurements should be) to maximize (minimize) measured value of the critical sensor
  - For example, minimizing *observed* sensor value of temperature can lead an actuator to increase actual temperature
  - I'll use minimization as an example

ATTACKER'S PROBLEM

$$\min y_{s_c}$$

$$s.t: |y_s - f(y_{-s})| \leq \tau_s \quad \textbf{Stealth}$$

$$||y - y_{true}||_0 \leq B \quad \textbf{Budget}$$

ATTACKER'S PROBLEM

- **Proposition**: Attacker's Problem is NP-Hard *even when linear regression is used for anomaly detection.*
- We devise:
  - Exact solution for linear regression models (integer linear program)
  - Iterative algorithm for the general case (heuristic)

SPECIAL CASE: LINEAR REGRESSION

$|y_s - f(y_{-s})| \leq \tau_s$: can be represented using linear constraints (since *f()* is linear)

$||y - y_{true}||_0 \leq B$: can be represented using linear constraints if we add binary variables indicating which sensors are attacked

Thus, the full problem can be captured using a Mixed-Integer Linear Program (MILP)

GENERALIZING

$|y_s - f(y_{-s})| \leq \tau_s$: **cannot be represented using linear constraints** for arbitrary non-linear *f()*

ALGORITHM FOR ATTACKING GENERAL NON-LINEAR MODELS

1. Obtain a linearized model by a **first-order Taylor expansion** around the solution estimate
2. Transform the problem to a **MILP**
3. Constrain solutions to be close to previous iterate (trust region)
4. If the solution of MILP is **infeasible w.r.t. stealth contraint**,

   reduce trust region
5. Repeat.

EXPERIMENTS: ATTACKS

CASE STUDY: TENNESSEE-EASTMAN PROCESS CONTROL SYSTEM (TE-PCS)

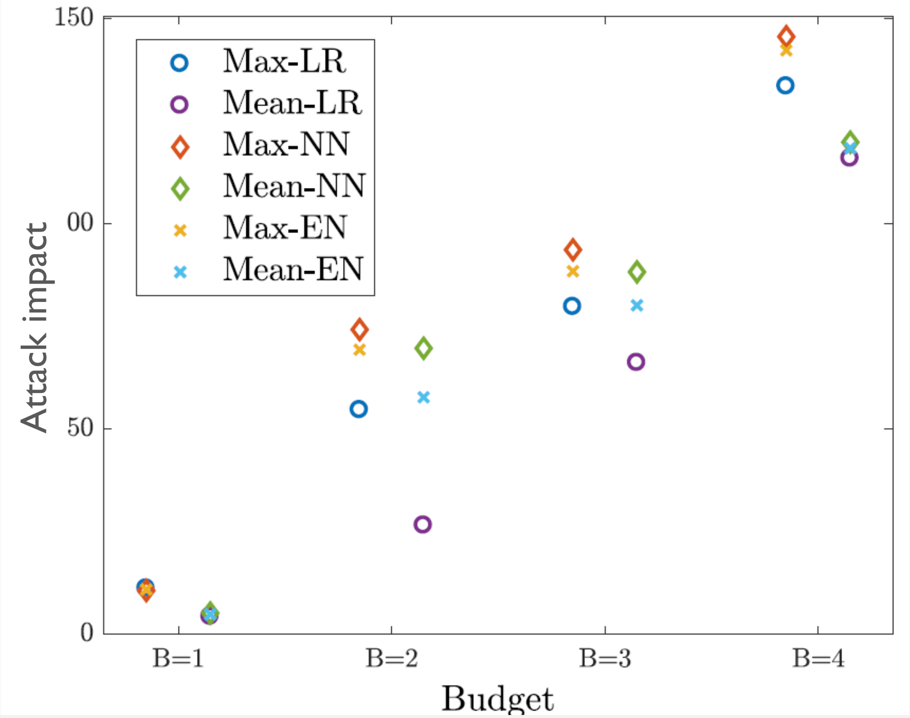
- Involving two simultaneous gas-liquid exothermic reactions for producing two liquid products

$$A(g) + C(g) + D(g) \rightarrow G(liq), \quad \text{Product } 1,$$
$$A(g) + C(g) + E(g) \rightarrow H(liq), \quad \text{Product } 2.$$

- **Five major units**: reactor, condenser, vapor-liquid separator, recycle compressor, and product stripper.
- Monitoring and control using **41 measurement outputs** and **12 control inputs**.
- Use a simulink model
- Consider **linear regression** and **neural network regression** for anomaly detection

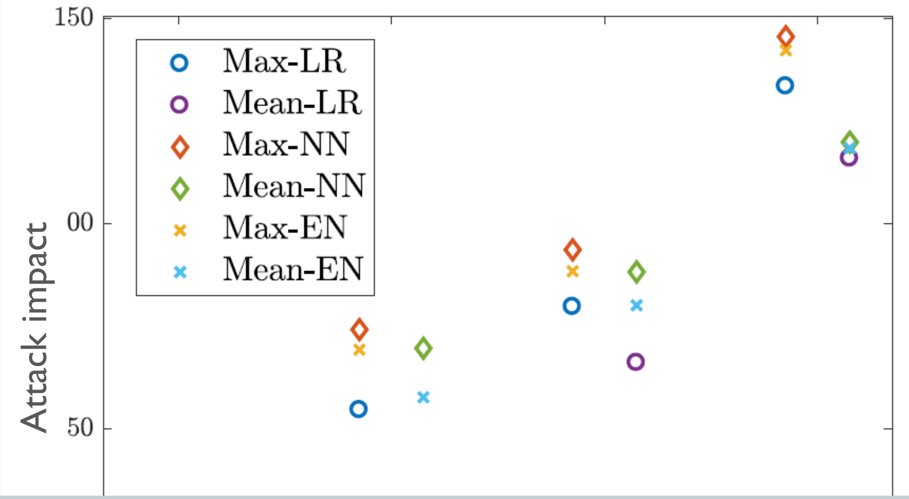
ATTACKING PRESSURE OF REACTOR

- Maximum and mean of the solution of adversarial regression:



ATTACKING PRESSURE OF REACTOR

- Maximum and mean of the solution of adversarial regression:



Neural network (diamonds) is more vulnerable than linear regression (circles)!

II. DEFENSE

DEFENDING AGAINST ATTACKS

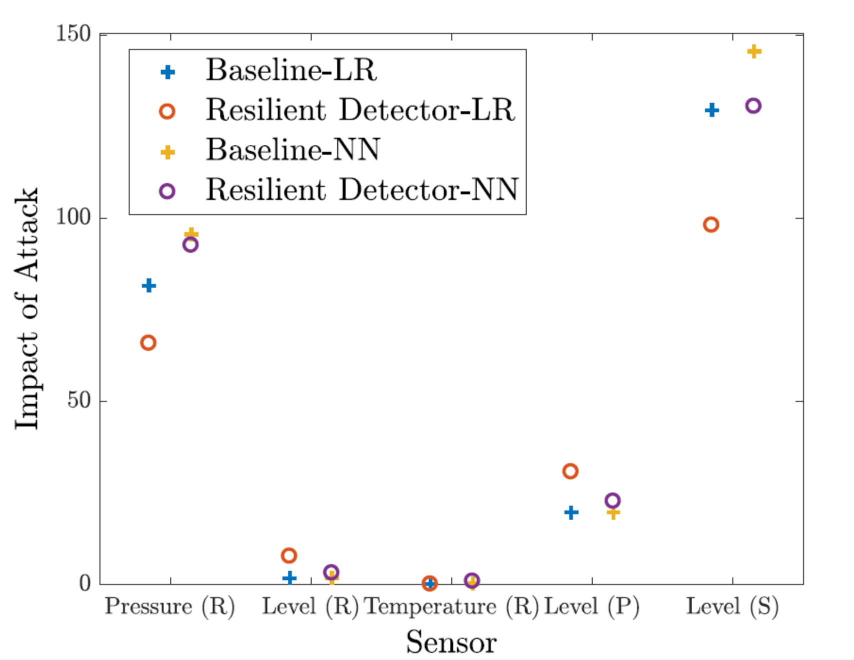
- In the anomaly detection system described, the defender can leverage the stealth constraint of the attacker's problem by appropriately choosing the detector thresholds
- Trade off:
  - Impact of attack: maximum distortion of critical sensor values induced by the attacker
  - False alarm rate: have a target false alarm rate
- Problem:
  - Minimize impact of attack (optimal solution to attacker's problem)
  - Subject to: false alarm rate is at most z

HEURISTIC ALGORITHM FOR OPTIMIZING THRESHOLDS

- Start with a baseline detector with false alarm rate z
- Iteratively:
  - Find optimal attack
    - A : sensors with largest attack impact
    - B : sensors with smallest impact
  - Reduce threshold on sensors in A
  - Increase threshold on sensors in B to keep false alarm rate at z
  - Stop when no longer reducing overall attack impact

EXPERIMENTS: RESILIENT DETECTOR

- Same setting as before
- Maintain the same # of false alarms as for an initial non-resilient detector



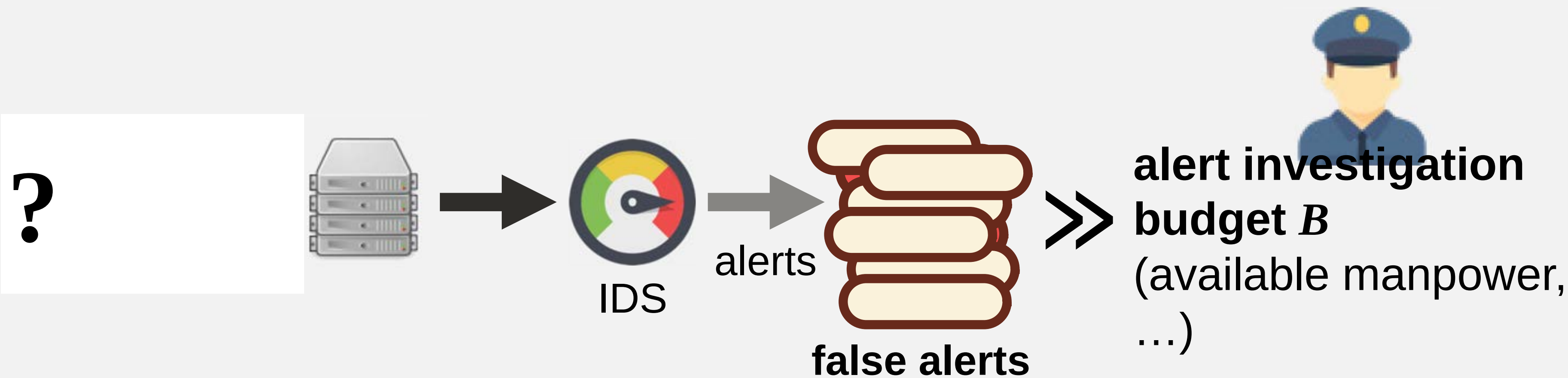
Significant reduction in attack impact relative to baseline for most vulnerable sensors

PRIORITIZING ALERTS

Yan, Li, Laszka, **Vorobeychik**, Fabbri, Malin. A game theoretic approach for alert prioritization. AICS 2017; ICDE 2018.

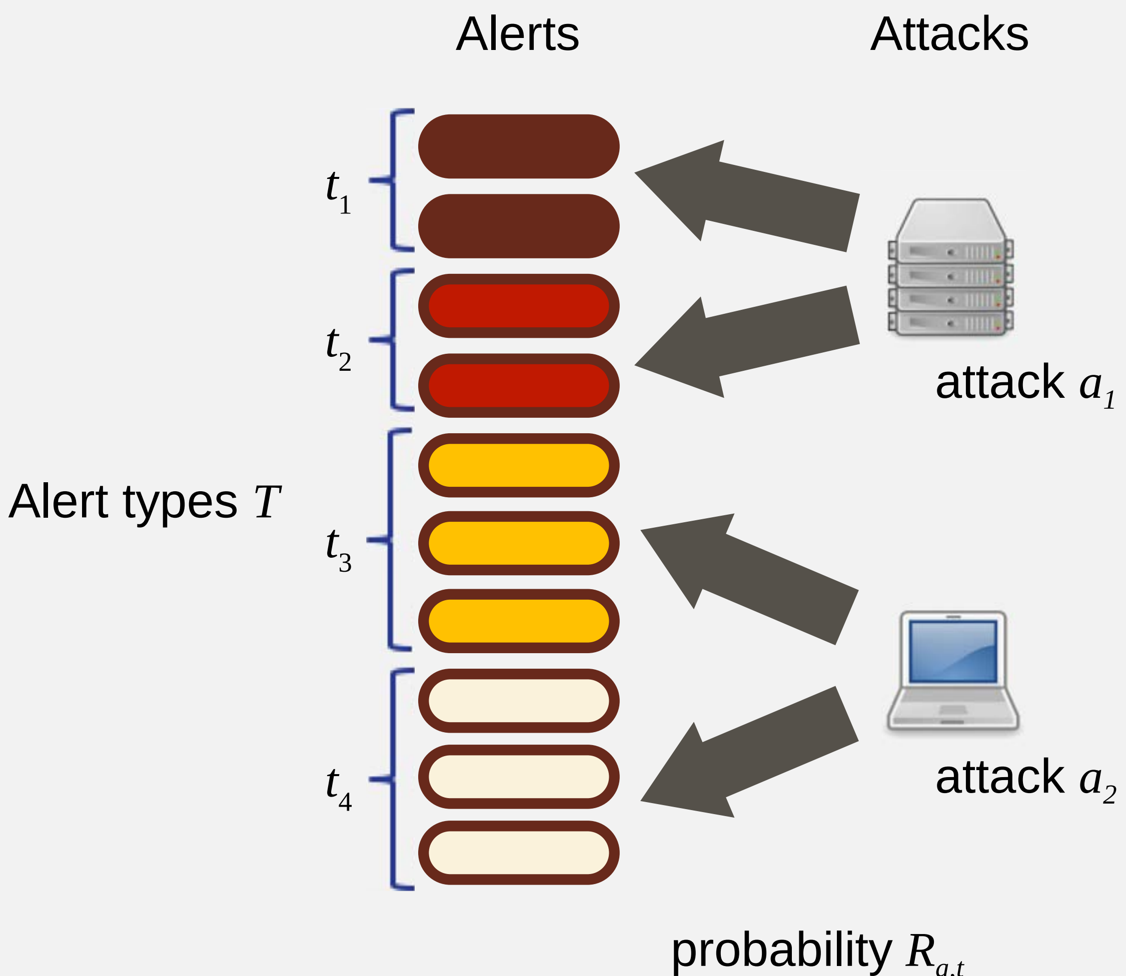
INTRUSION DETECTION

- Detectors generate alerts
- Typically, people would subsequently investigate alerts to find malicious activity

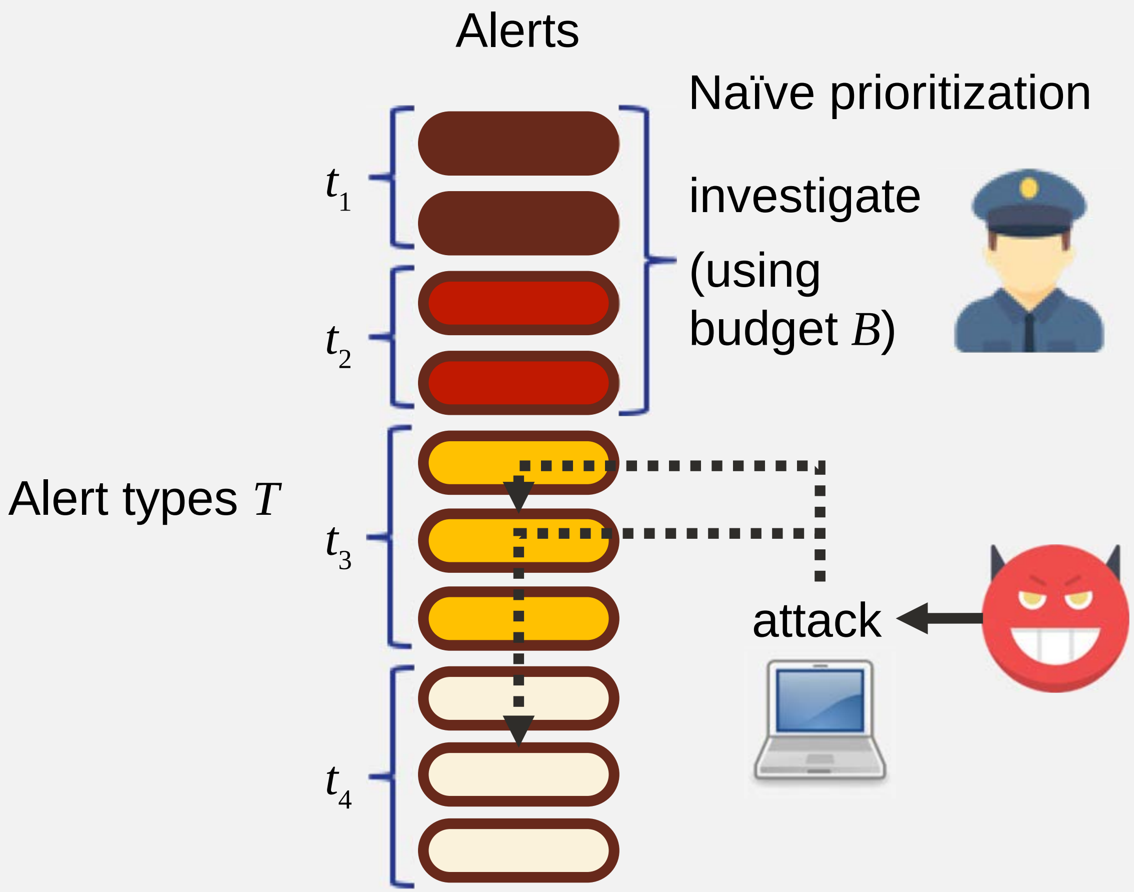


Problem:
Which alerts to investigate?

ALERT TYPES


Alerts
Attacks
t_1
t_2
attack a_1
Alert types T
t_3
t_4
attack a_2
probability $R_{a,t}$

ALERT PRIORITIZATION PROBLEM



ALERT PRIORITIZATION PROBLEM

GAME-THEORETIC MODEL

- Players

1. Defender: selects an alert prioritization strategy p, which is a probability distribution over possible orderings of T

2. Adversary:
selects an attack a from the set of possible attacks A

Goal: minimize probability of successful (undetected) attack

Solution approach: linear programming + column generation

CONCLUSION

- Detection is fundamentally a game
- This game must capture a number of features
  - Indirect as well as direct consequences of decisions
  - Adversarial actions to avoid being detected
  - Detectors are imperfect, and there are only so many alerts we can inspect
    - Need to account for intelligent attacks even as we select which alerts to investigate