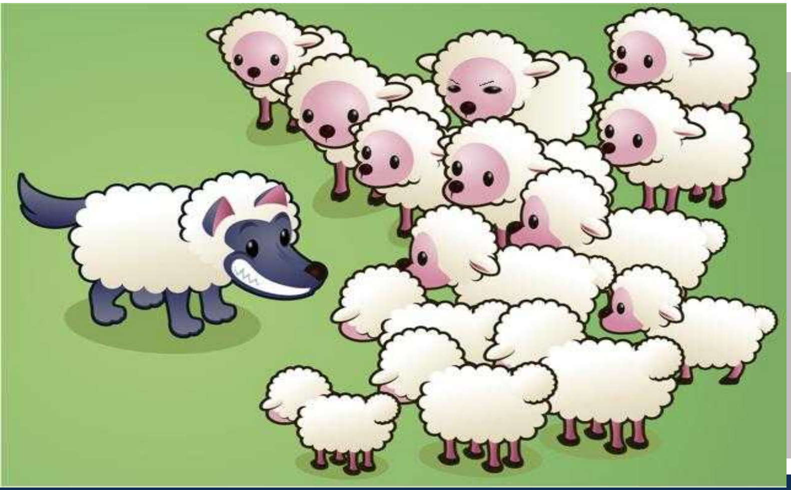
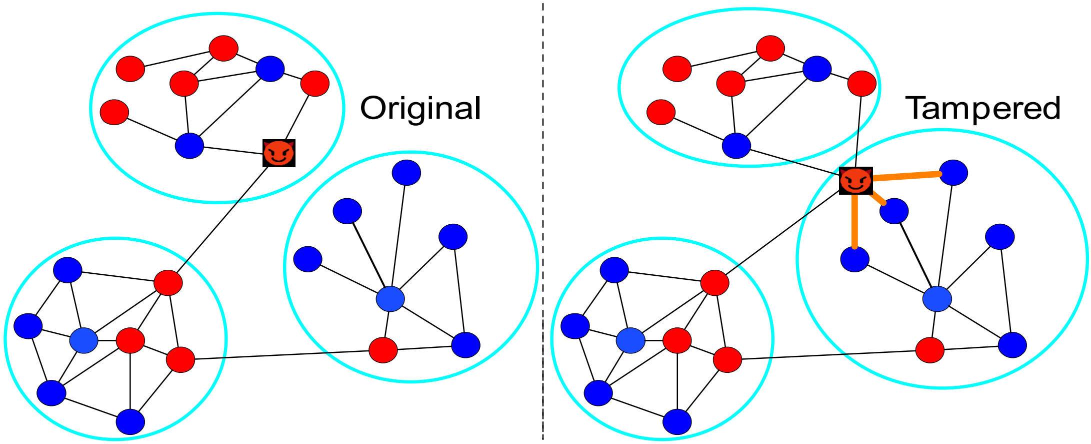
Exceptional service in the national interest

SAND2017-11548C





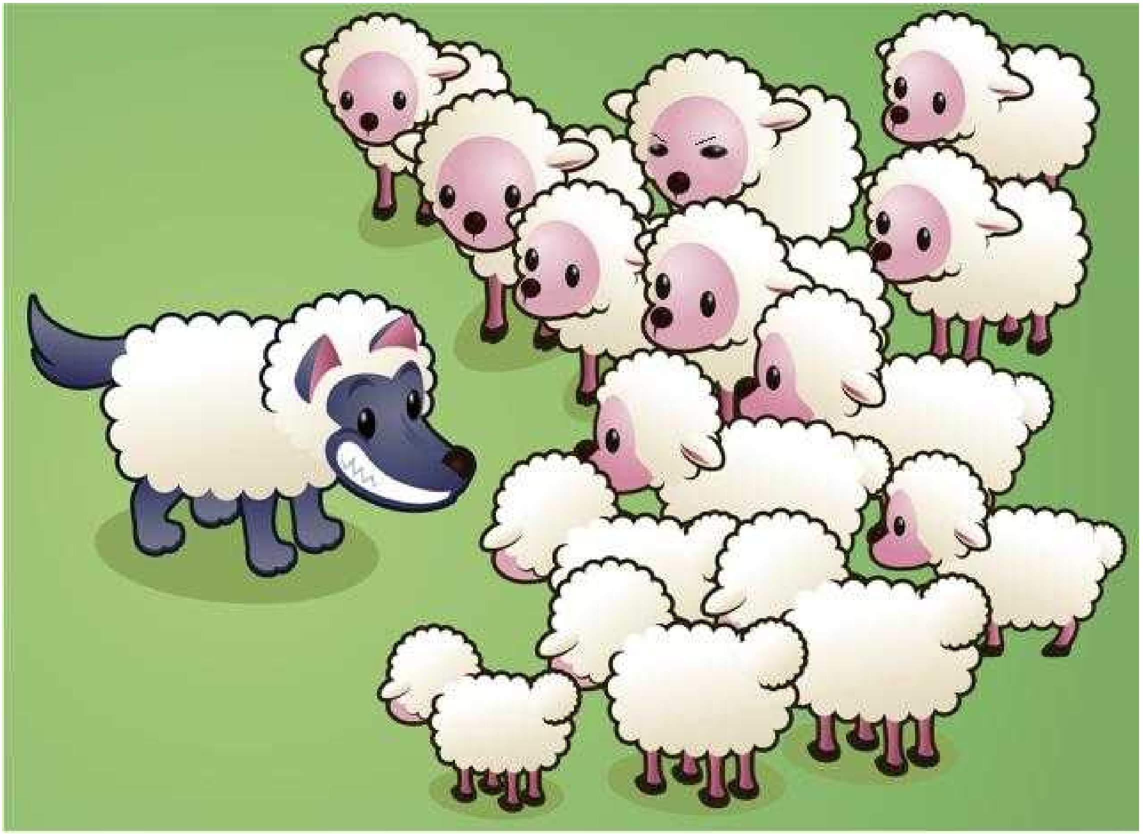
Counter Adversarial Graph Analytics

Philip Kegelmeyer, Jeremy D. Wendt, Ali Pinar, Kristen Altenburger

Sandia National Laboratories Stanford University

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.

Counter Adversarial Data Analytics (in general)



Common Wisdom: If white and fuzzy, then harmless

Goal: to be the suspicious sheep

Philosophy

We must learn to love life

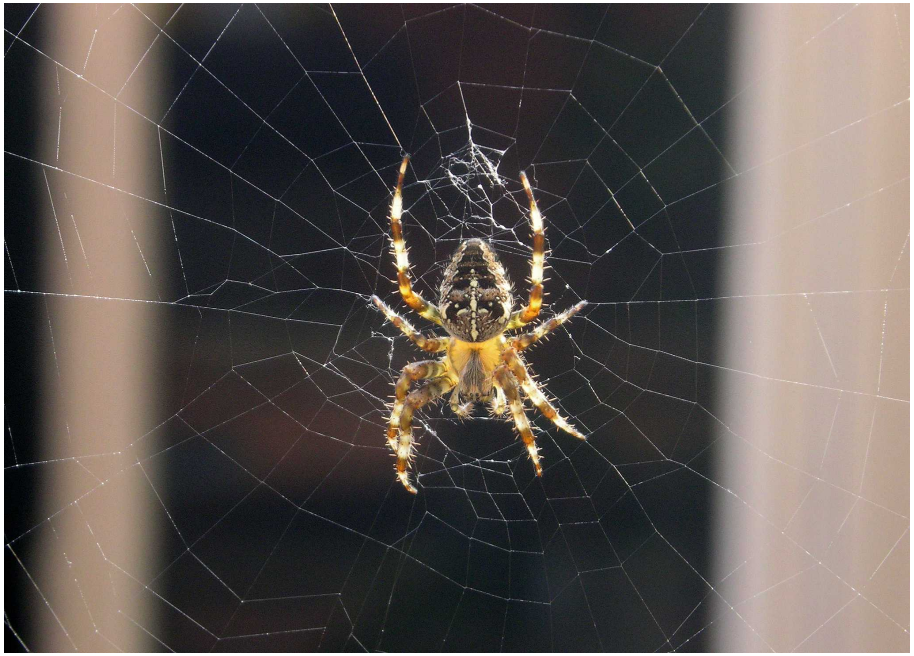
We must learn to love ~~life~~ data
… without ever trusting it.

without ever trusting it

Our broad question: how to turn this into quantifiable, practical advice?

An Algorithmically Informed, Empowered Adversary

- For the current talk, we assume the worst case: an adversary that knows every detail of our analytic and has some ability to alter the network.

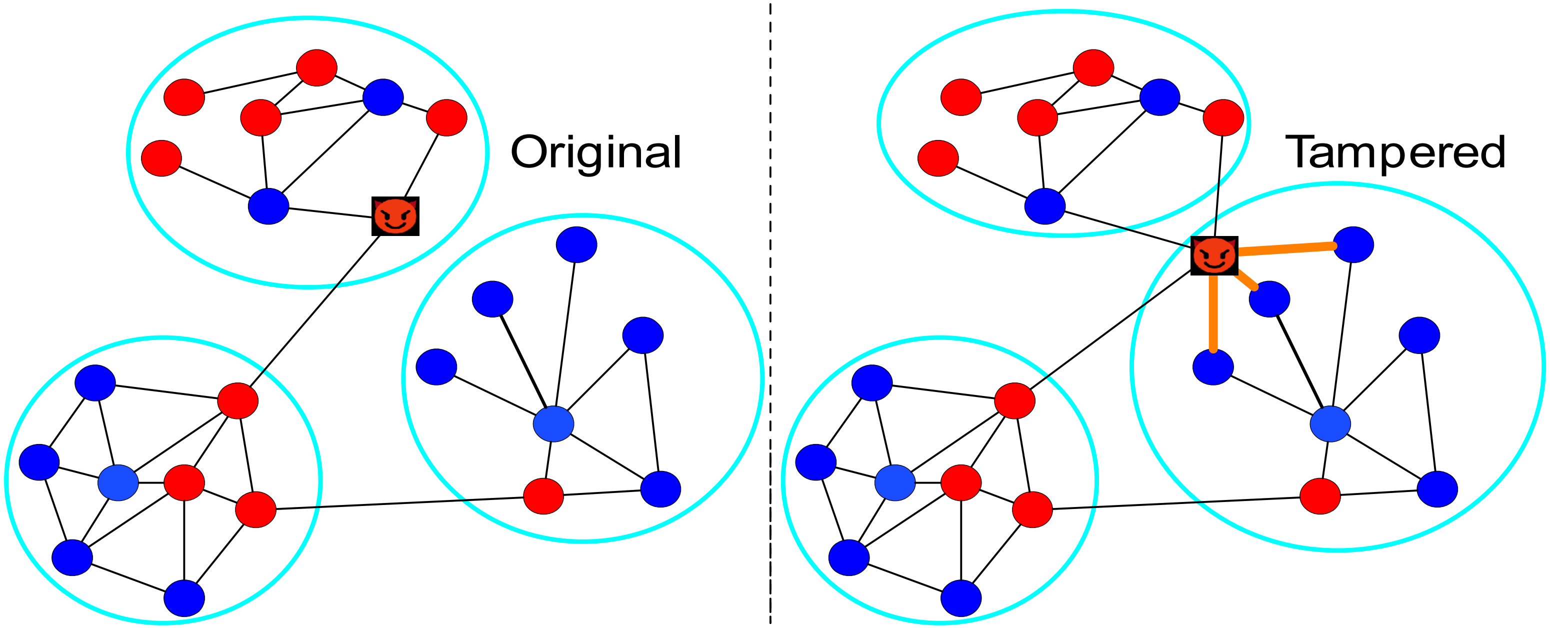


Fully Informed . . . . . . And Empowered

We aim to quantify just how badly we are hosed.

Community Detection For Prioritizing Investigation

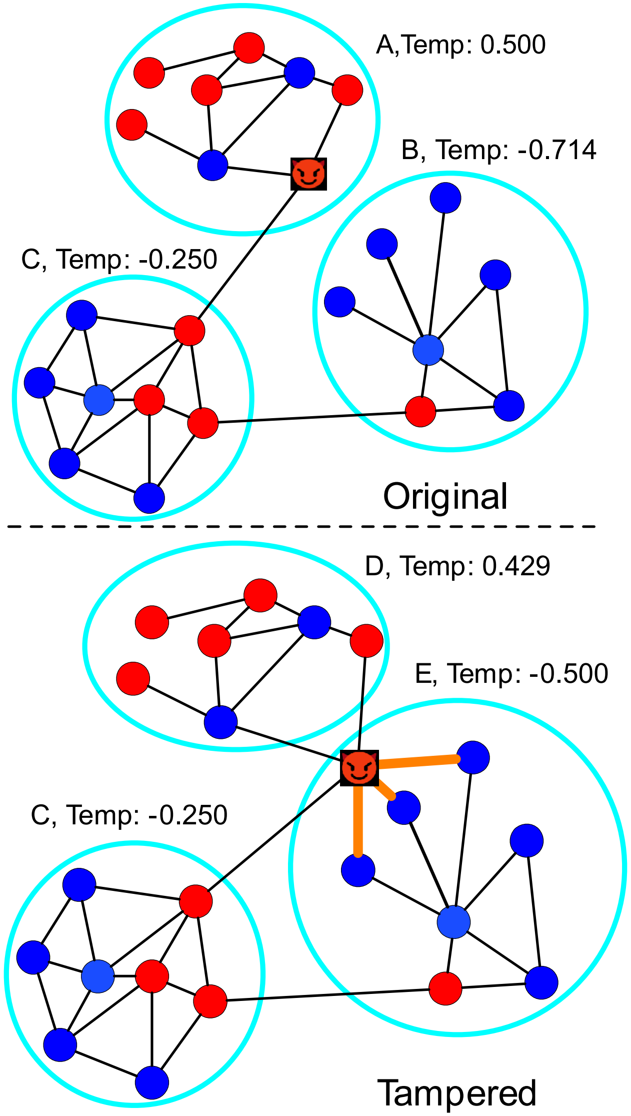
- Red and blue are the “temperatures” of nodes.
  A community’s temperature is the average of its nodes.
  The hotter the community, the more likely to be scrutinized.



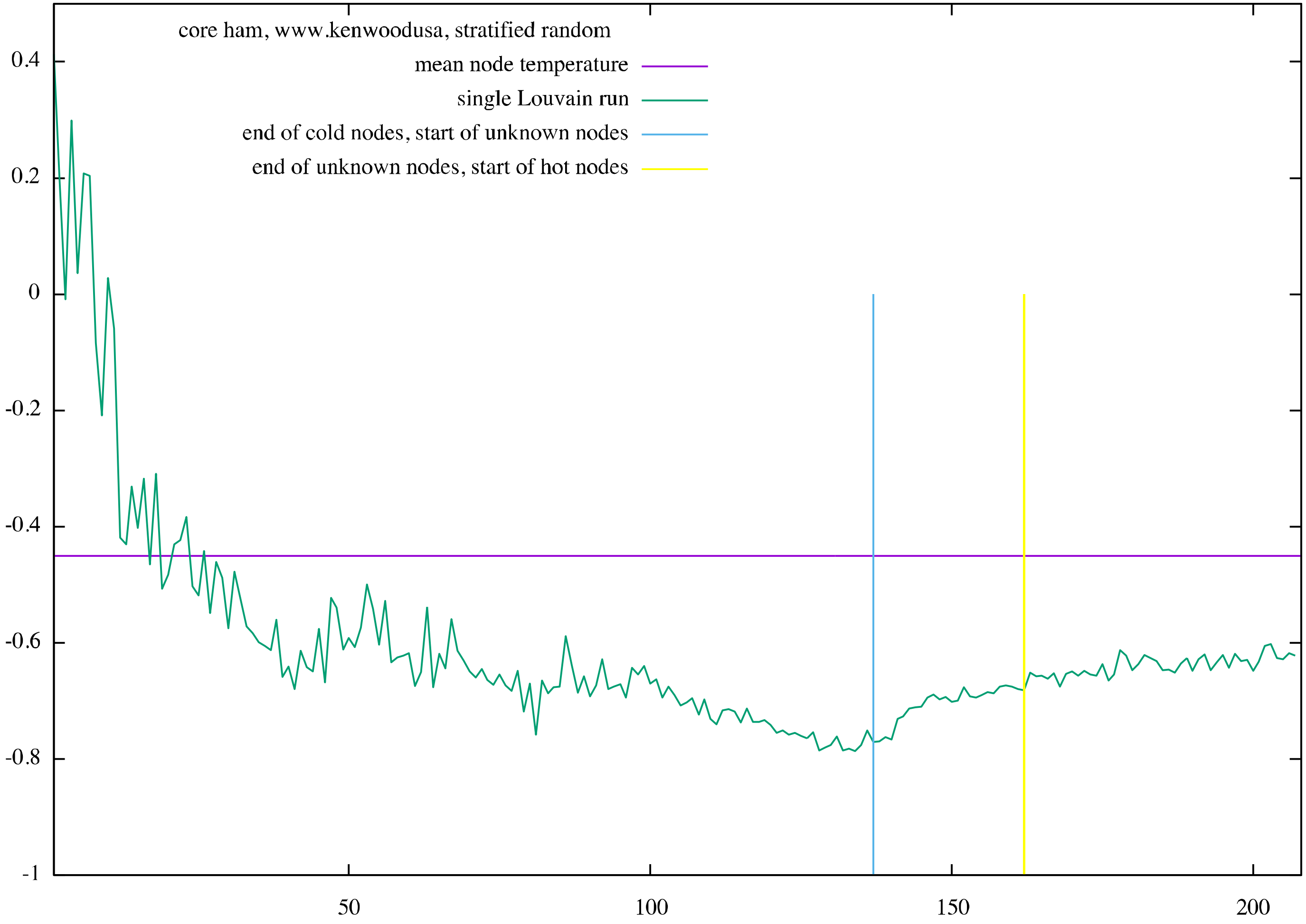
- The red fiend is the adversary node. Its goal: not be in a hot community.
  Attack: add links (orange) from self, to tamper with community structure.

Node Vs. Community Temperature

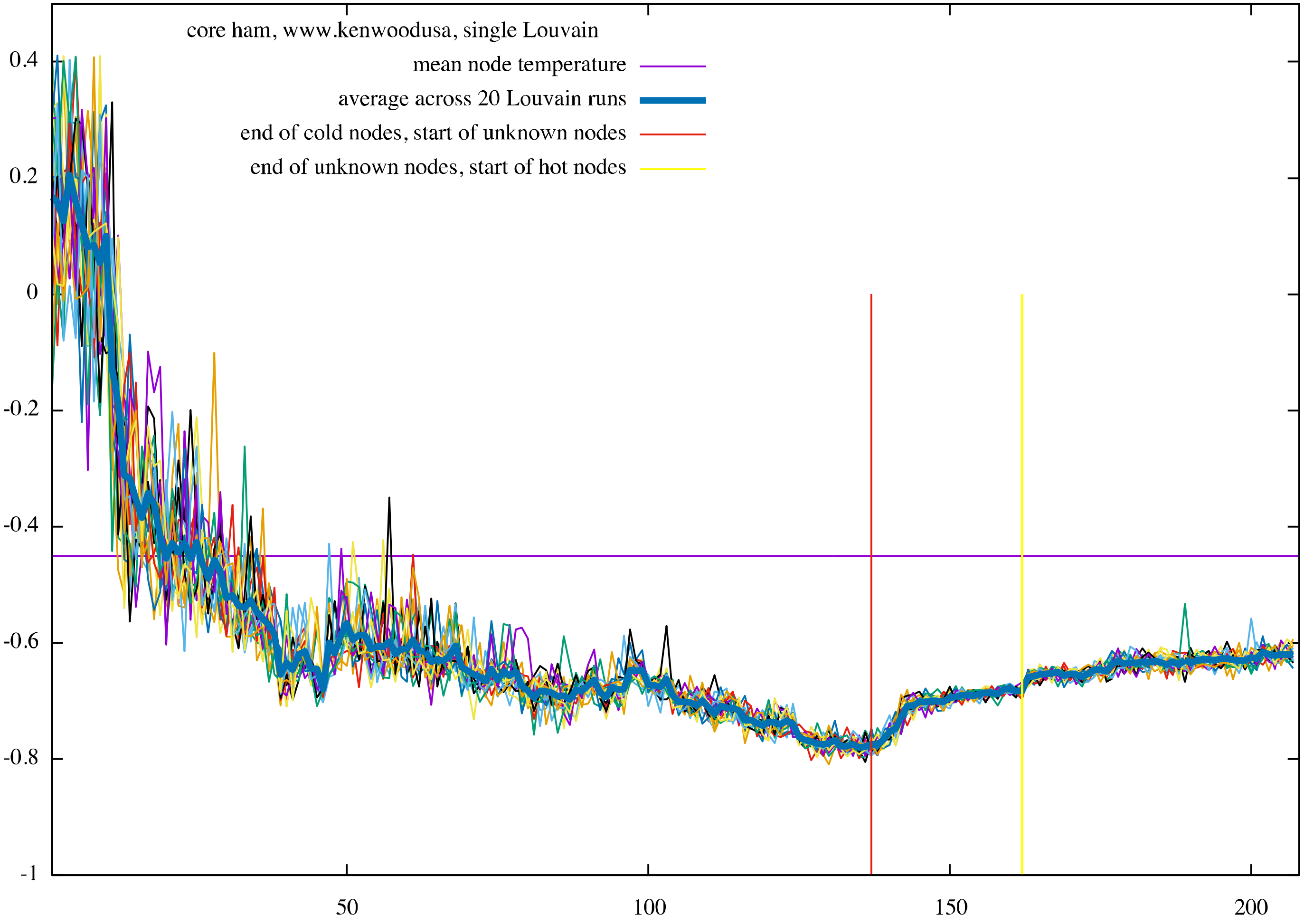
- Hot (red) = 1, Cold (blue) = -1, Unknown = 0.
- Community temperature is the average of nodes.
- So higher temperature means more suspicious.
- Compute the average temperature in each community
- • A: $T = (6 \times 1 + 2 \times -1)/8 = 0.500$
- • B: $T = (1 \times 1 + 6 \times -1)/7 = -0.714$
- • C: $T = (3 \times 1 + 5 \times -1)/8 = -0.250$
- • D: $T = (5 \times 1 + 2 \times -1)/7 = 0.429$
- • E: $T = (2 \times 1 + 6 \times -1)/8 = -0.500$



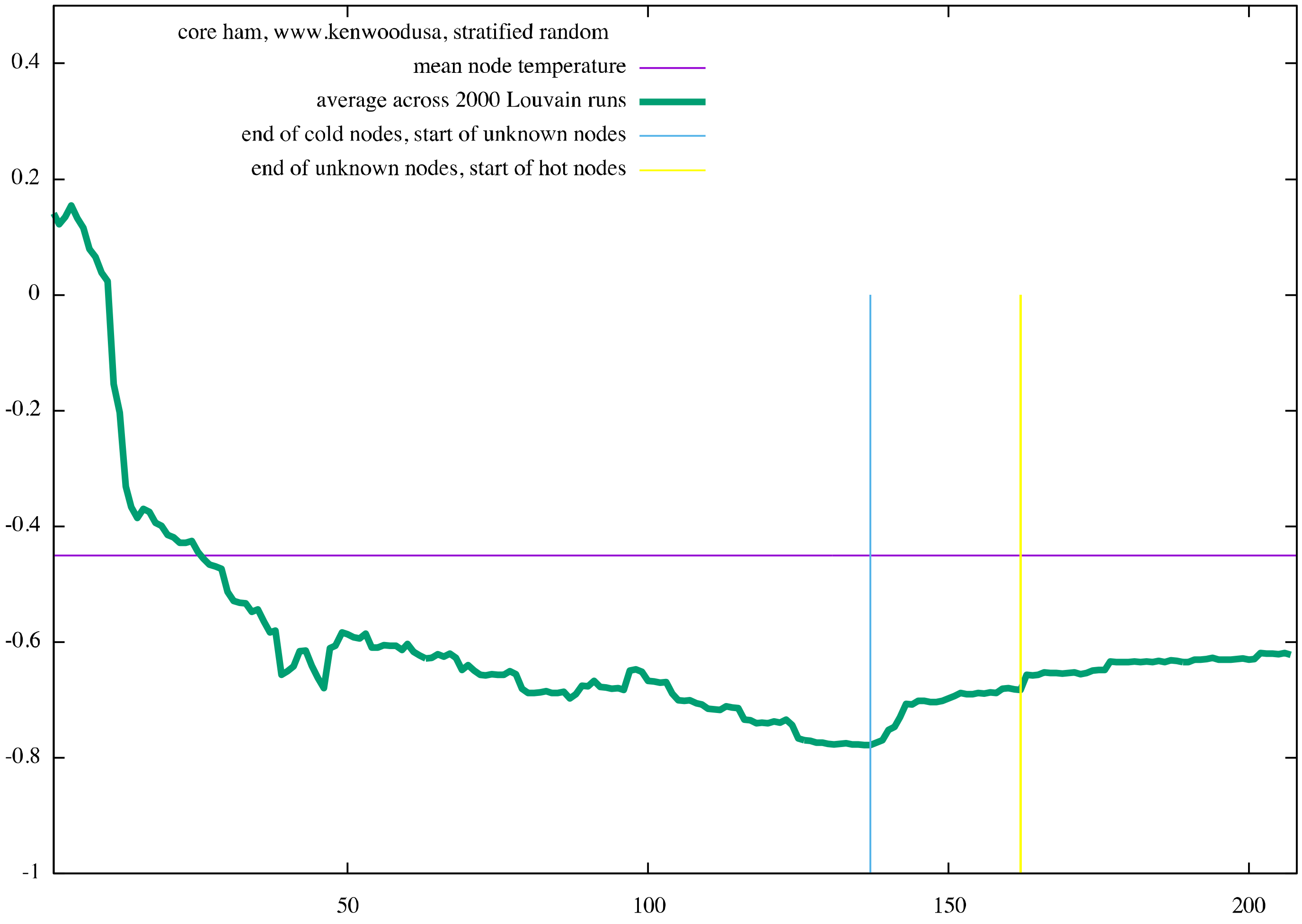
One Louvain of *One* Stratified Random Attack



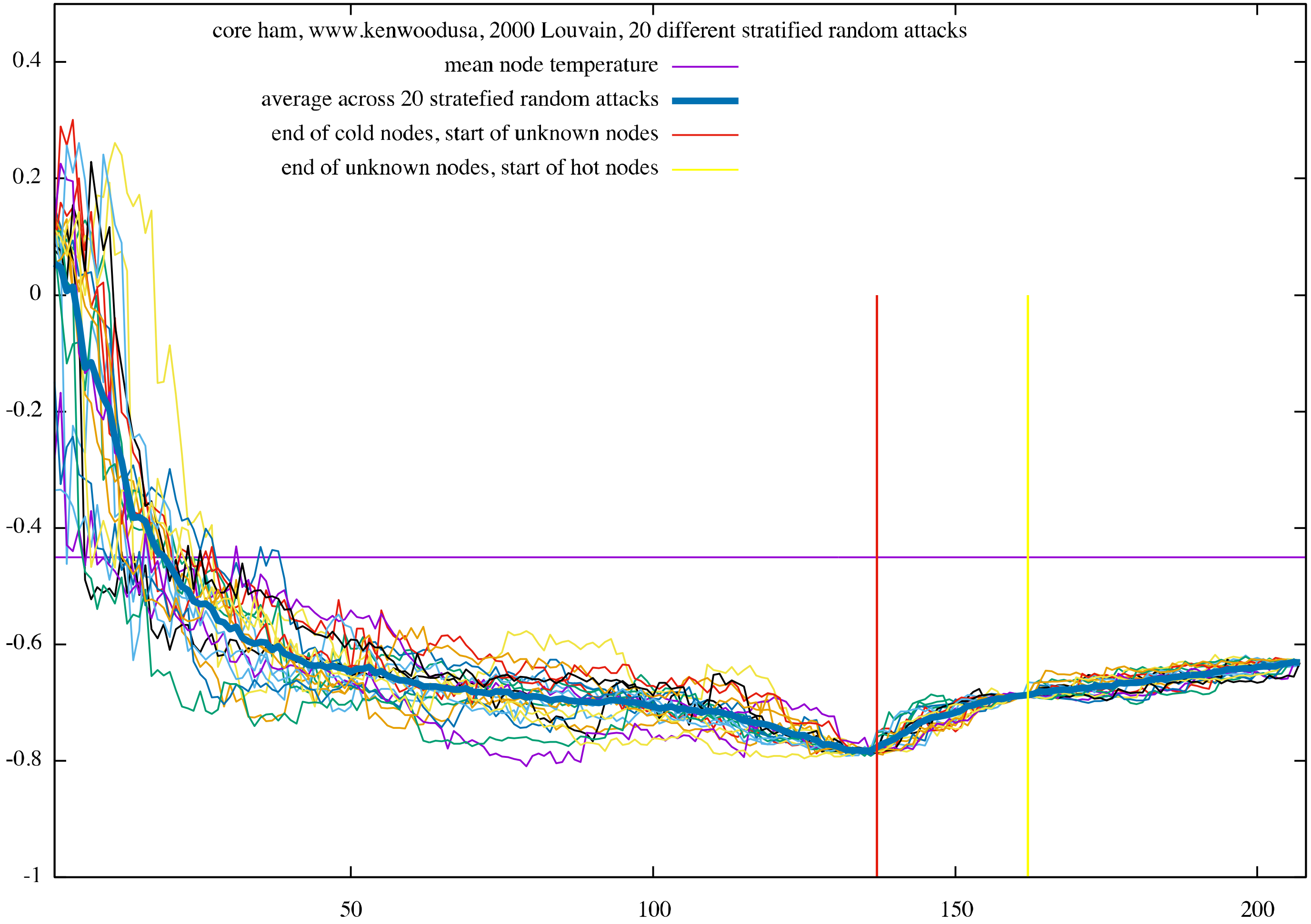
Twenty Louvains of *One* Stratified Random Attack



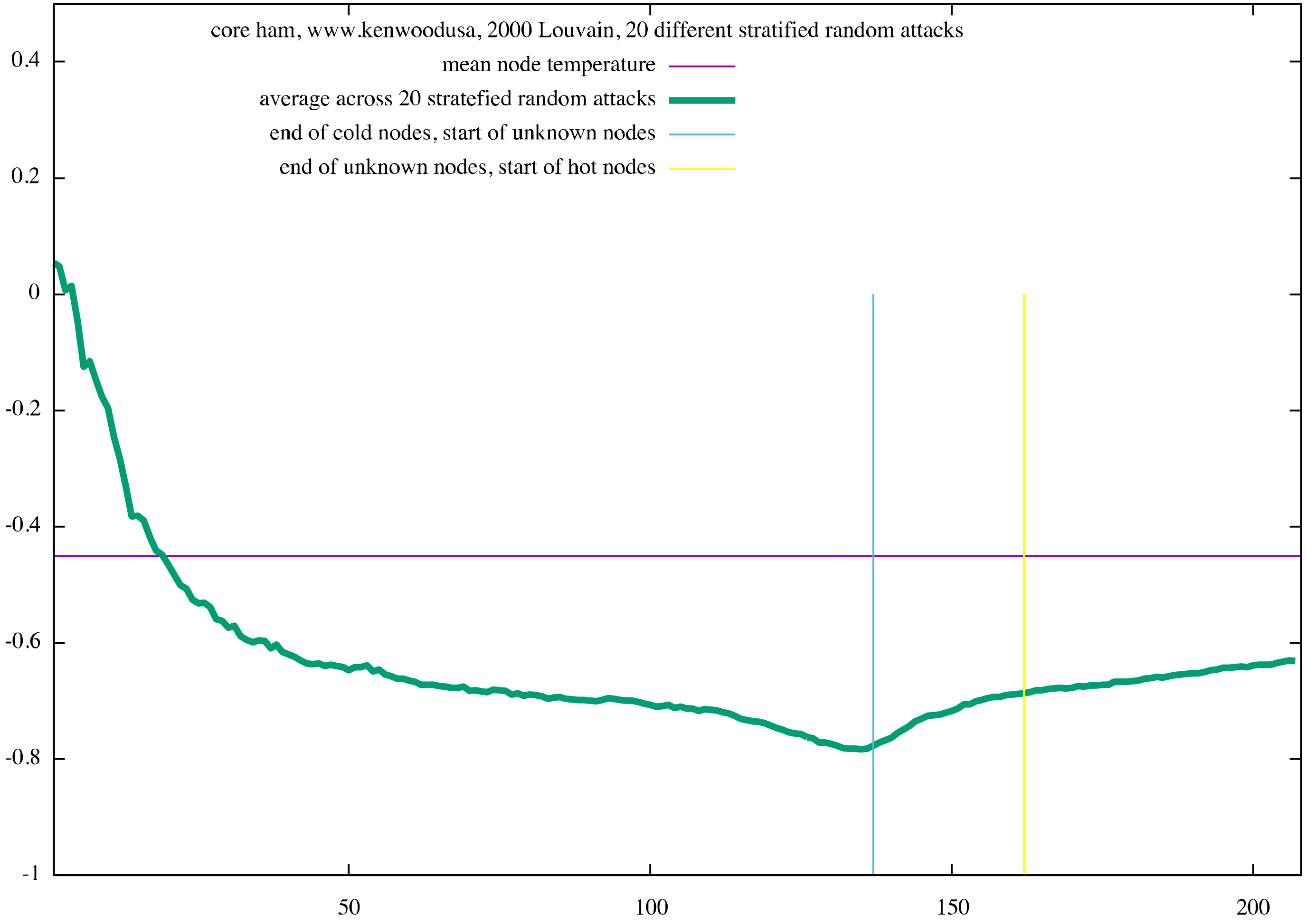
2,000 Louvains of *One* Stratified Random Attack



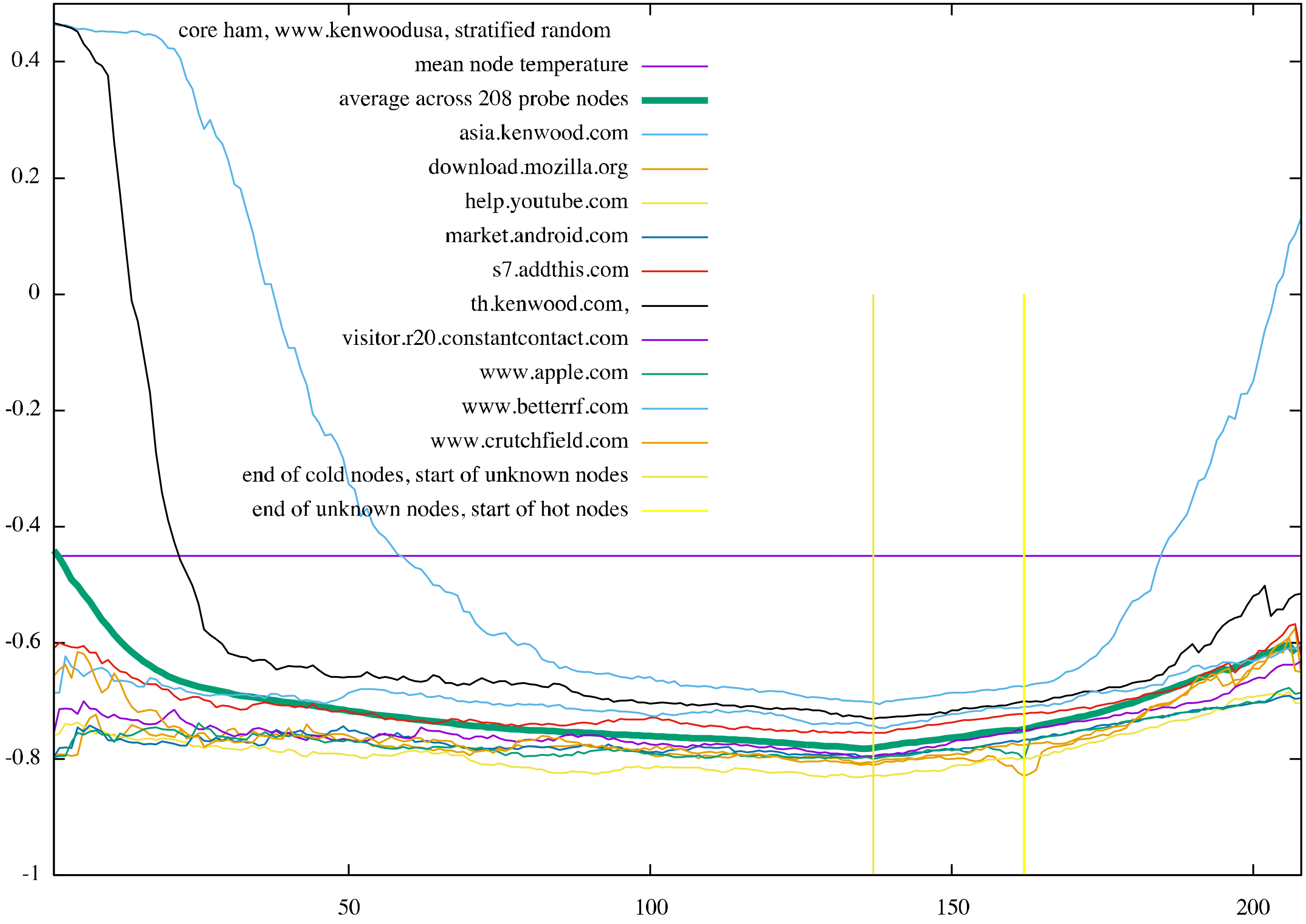
2,000 Louvains of *20* Stratified Random Attacks



Stratified Random Attack Against One Node Only

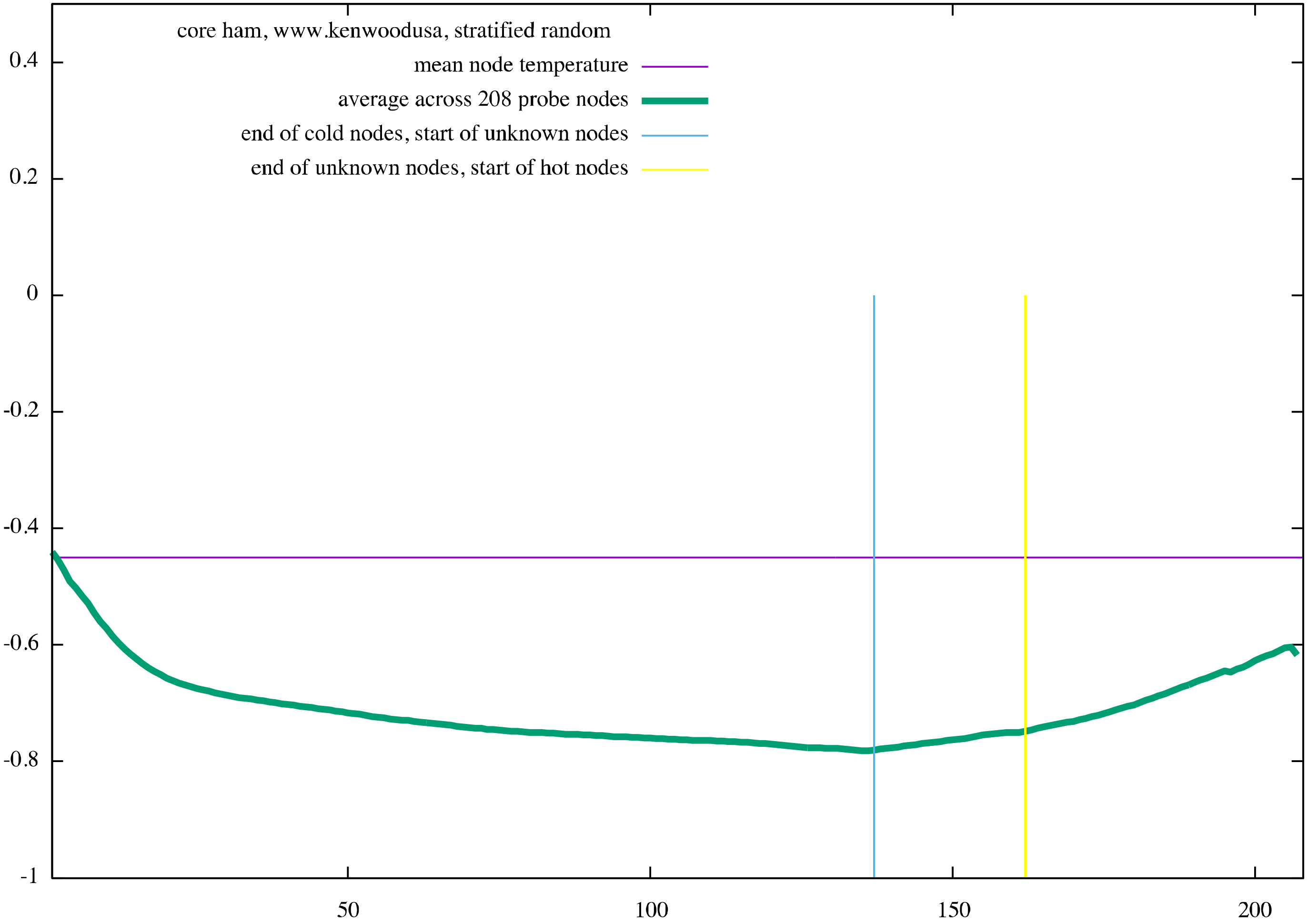


Different Probe Nodes Yield Different Curves



So Average Across All Probe Nodes

This, finally, is an overall efficacy curve for the ``stratified random" attack as applied to a single graph.



We’ve Devised Multiple Attacks

Recall: an “attack” is just a heuristic for choosing the “who to link to” ordering of all of the other nodes in the graph.

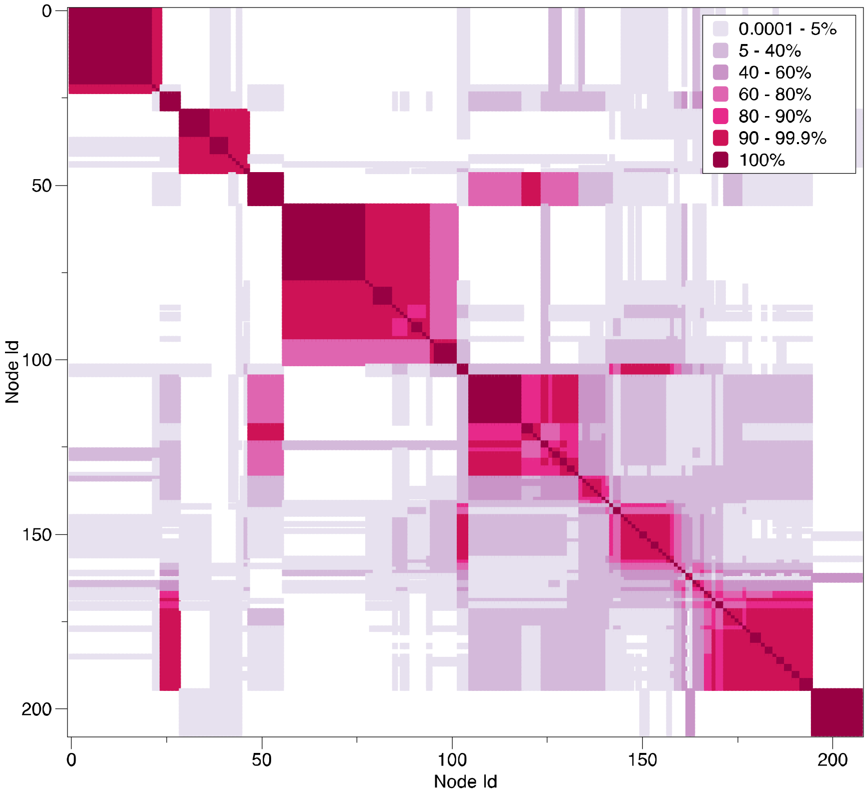
- Stratified Random:
  - Cold, then unknown, then hot nodes;
  - Each cohort randomly ordered.
- Cold and Lonely:
  - Cold nodes in order of increasing degree, unknown in random order, hot nodes in order of decreasing degree; each cohort randomly ordered.
- Greedy Pesimal:
  - Exhaustively search for best unattacked node to attack; repeat. (Infeasible for real graphs.)

```
ham_2_core.dot
www.kenwoodusa.com
# Probe vertex above here
mail.google.com
video.google.com
checkout.google.com
www.elkantennas.com
news.google.com
www.google.com
maps.google.com
chrome.google.com
www.blogger.com
google.com
code.google.com
services.google.com
www.orkut.com
promote.orkut.com
feeds.feedburner.com
googlewebmastercentral.blogspot.com
```

First 15 nodes in a particular attack

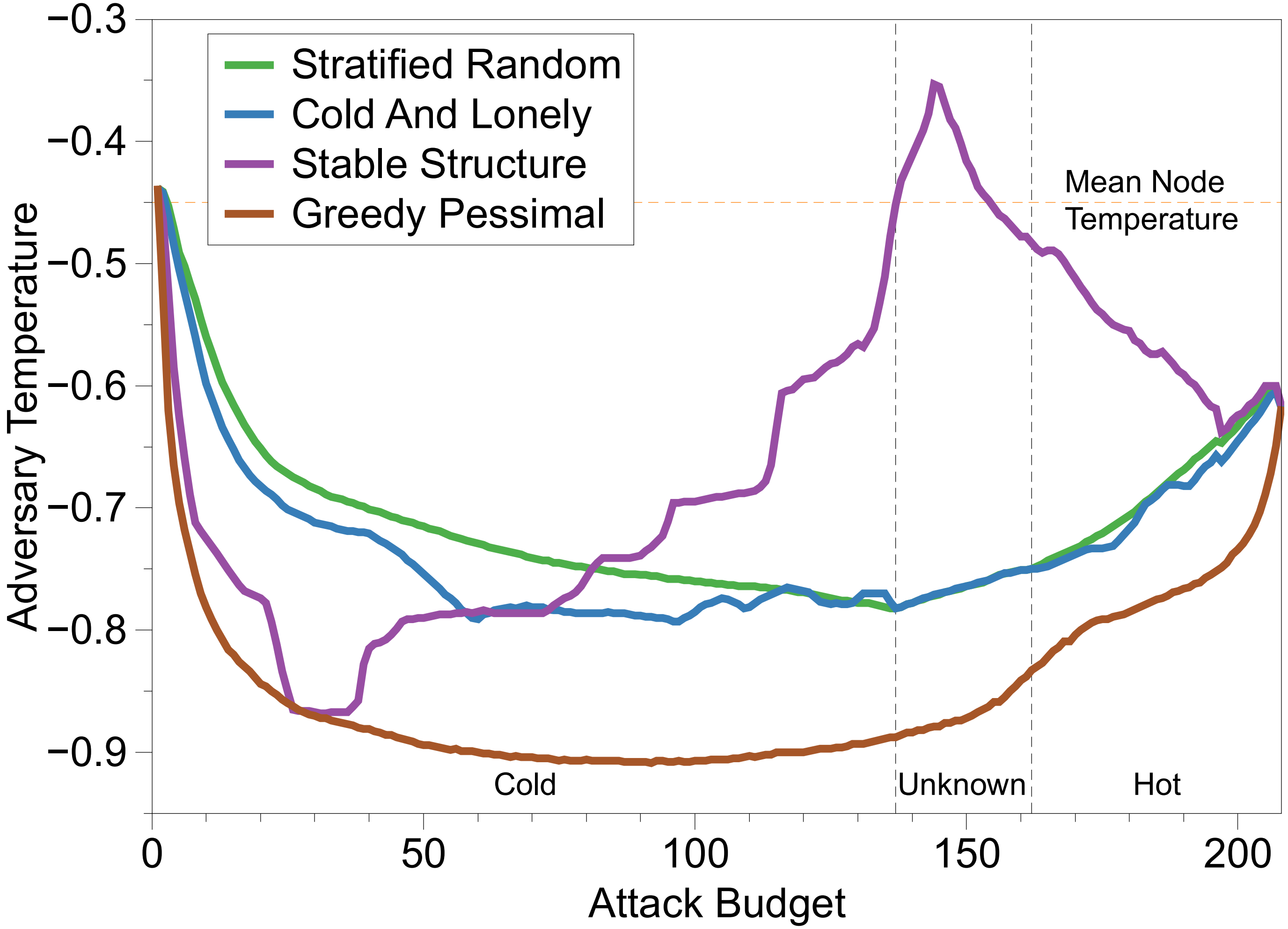
“Stable Structure” Attack

- Repeat Louvain N times. A “stable structure” is a set of nodes V such that all nodes in V always end up in the same community with each other. (Dark maroon in the figure.)
- The attack heuristic:
  - Pre-process to extract all stable structures.
  - Link to the coldest stable structure in random order, then next coldest, and so on.
  - Then revert to “stratified random” on the remaining free-agent nodes.
- Offers better scalability for larger graphs



Stable Structure From 2,000 Louvains

The Relative Efficacy of These Attacks



Can We Defend Against Attacks?

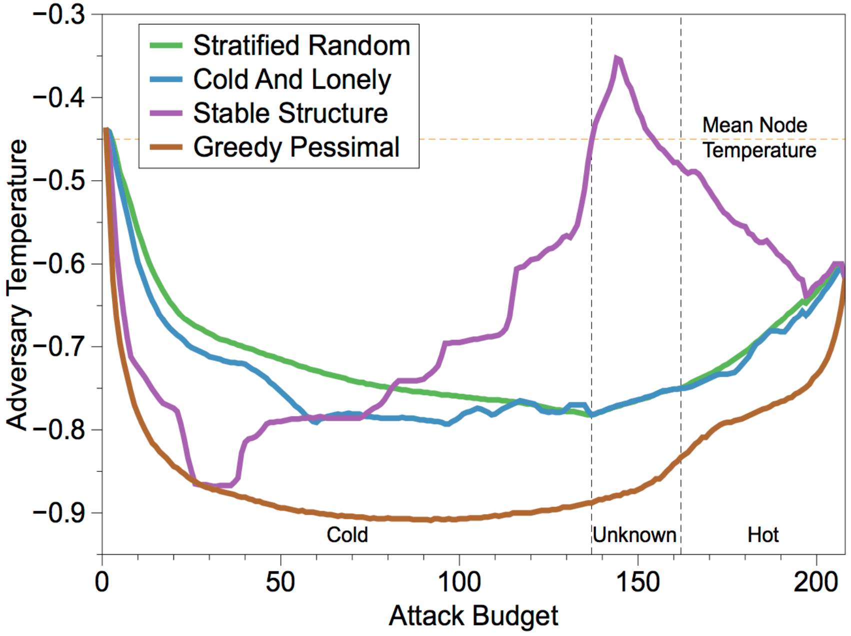
- Defend against these attacks: identify the inserted edges and remove them.
- In other counter-adversarial work, we found we could train ML to identify adversary-altered data.
- Can the same be done here?
- Note that we can’t simply train and test on random samples of edges from the same graph.

canacSBM

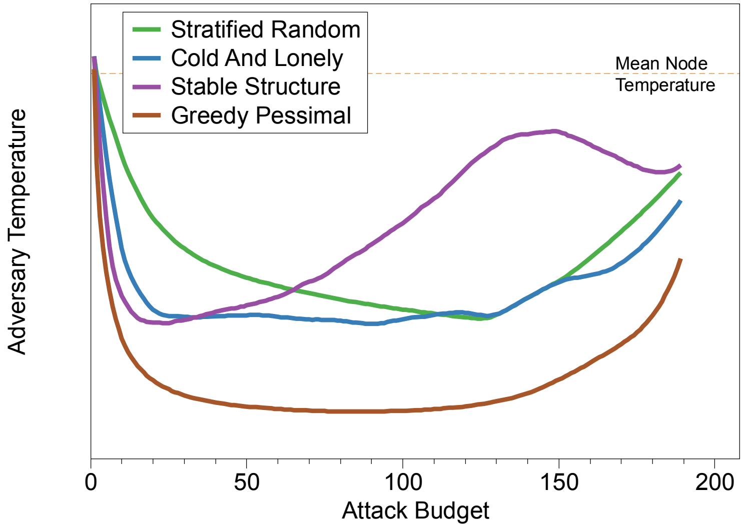
- In this case, we have only the one graph sample.
- So infer the real graph's characteristics, then generate more graphs that are statistically similar.
- We developed a community and node attribute corrected Stochastic Block Model, "canacSBM".
- Like a generalization of Chung-Lu:
  - Estimate communities
  - Preserve a node's expected degree within its estimated community.
  - Treat temperature attribute as blocks within a community, and do "attribute corrected" SBM.
- This is a drastic simplification; ask for our pre-print.

How Well Does canacSBM Match Real Data?

Does canacSBM agree with real data in terms of attack efficacy?



Original Graph



Averaged canacSBM

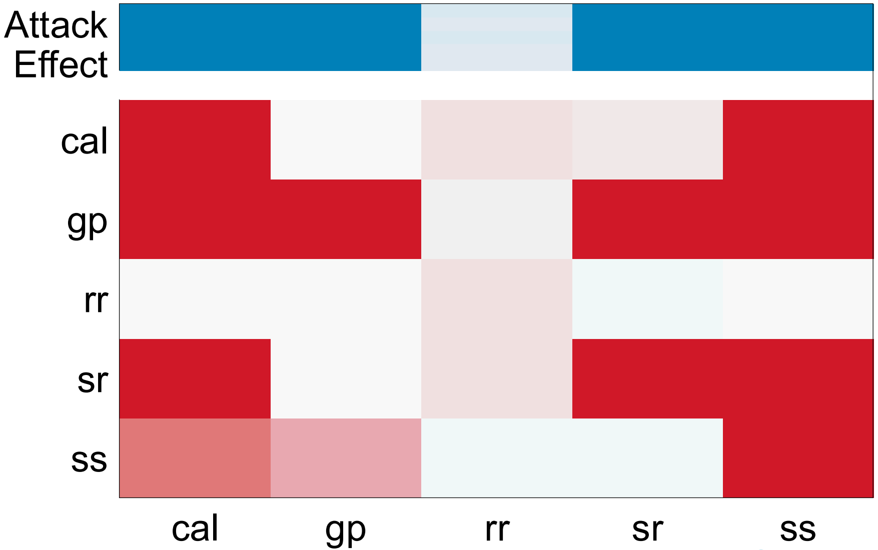
More pointedly, can canacSBM help in defense?

Defending Using ML and canacSBM

- Train defense model:
  - Select training node and training attack.
  - Attack canacSBM graph with budget of 20 nodes.
  - Extract features for each edge in attacked graph.
  - Train ensemble of decision trees to differentiate "inserted" vs. "original".
- Use model to defend:
  - Select testing node and training attack.
  - Attack original graph with budget of 20 nodes.
  - Extract features for each edge in attacked graph.
  - Apply ensemble created above on these features.

Measuring Remediation Effect

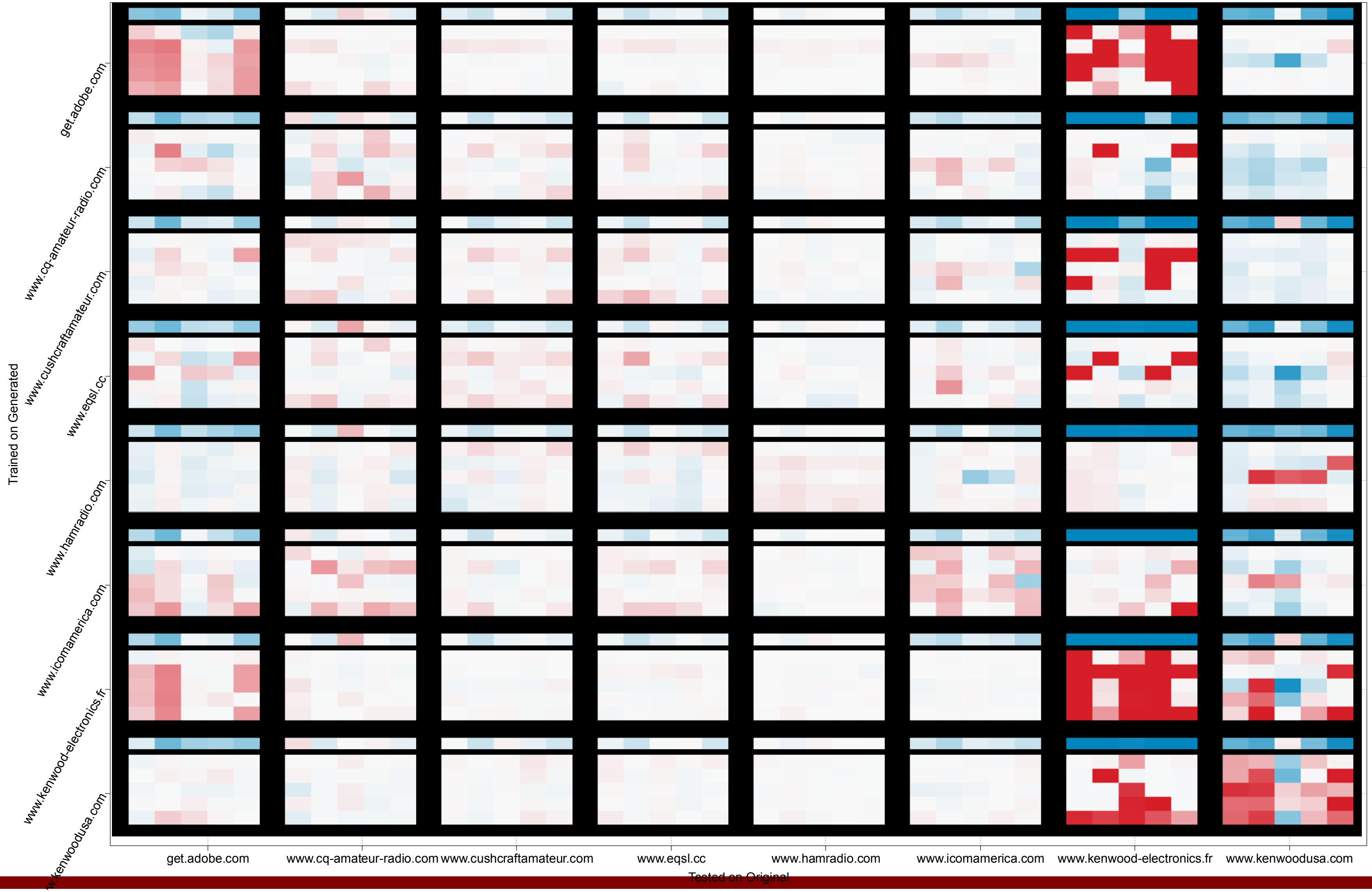
get.adobe.com



www.kenwood-electronics.fr

- Measure defense's effectiveness:
  - Remove all edges identified as "inserted" – call this remediation.
  - Compute probe's community temperature before attack, after attack, after remediation.
  - Attack effect is change in temperature due to attack (generally cooler); remediation effect is change in temperature from attacked-to-remediated.

Train on All canacSBMs; SMOTE; Test the Model



What’s Next? Similarly Undermining Node Labeling

- Still, possible community detection follow-ups:
- Design a robust-to-attack community detection algorithm that trades modularity against the presence of locally homogeneous hot sub-communities.
- Change the efficacy metric? Sort order likely matters more than temperature, given our scenario.
- Generalize our (unrealistically restricted) adversary attack model to one that permits a number of adversary nodes, in collusion.

What I Hope I Showed Today

- Why counter-adversarial analysis?
  - (And what does that mean, exactly?)
- Using community detection to prioritize investigation.
- Inventing attacks against that use.
- Quantifying the efficacy of those attacks.
- Some possible defenses.