Why topology helps to detect cyber-intrusions

Martine Ceberio[1], Olga Kosheleva[2], and Vladik Kreinovich[1]

Department of [1]Computer Science and [2]Teacher Education
University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA
mceberio@utep.edu, olgak@utep.edu, vladik@utep.edu

1. Cyber-intrusions are a big problem

- Cyber-intrusions are a big problem for communications.
- Economy-wise, they cost companies billions of dollars every year.
- They threaten national security.
- It is important to decrease their negative effects.

2. Why detecting cyber-intrusions is difficult

- At first glance:
  - since modern machine learning (AI) methods are so powerful,
  - why not use them to detect cyber-intrusions?
- These methods have indeed been used, but their success is limited.
- There are two main reasons for this limitation.
- First, machine learning means, in effect, recognizing the known patterns.
- Machine learning tools learn to recognize intrusions which are similar to the previous once.
- However, adversaries are constantly coming up with new intrusion schemes.
- Machine learning tools cannot do much against such innovative attacks.

3. Why detecting cyber-intrusions is difficult (cont-d)

- The second reason is the too-much-information problem.
- All the transactions are usually recorded.
- So after each intrusion, we have a very large amount of data.
- Some of this data may be relevant, some not – but we do not know this a priori.
- As a result, we have to submit all this information to the machine learning tool.
- For such huge amounts of data, AI tools work very slowly and not very effectively.

4. Need to select relevant features

- A natural way to overcome this problem is to select relevant features.
- Then, instead of using the whole record of an intrusion, we can use only these features.

5. How to select relevant features: a natural idea

- Intrusion affects computer networks.
- We want to detect which computers have been affected – and are now under full or partial control of an adversary.
- All the computers in a network perform a lot of activities.
- So, a natural thing to do is to compare these activities:
  - between different computers and
  - between past and present activities of the same computer.
- For new threats, past activities are probably not that helpful; so:
  - if we have to select the smallest possible number of features,
  - it makes sense to only use comparisons of different computers' activities.
- Depending on the level of similarity, we can have clusters of computers that have some similarity level.

6. How to select relevant features: a natural idea (cont-d)

- For smaller levels of similarity, we have larger clusters.
- For larger levels of similarity, we have smaller clusters.
- A smaller cluster corresponding to a higher level of similarity is contained in a cluster corresponding to a lower level of similarity.
- Clusters with such an inclusion relation naturally form a graph, in which:
  - clusters are vertices and
  - edges correspond to inclusion.
- So, the question becomes: how can we detect cyber-intrusions based on this graph?

7. Empirical result: topology helps

- It turns out that a very good indication of an intrusion is:
  - when we have a closed cycle in this graph,
  - or, more generally, when we suddenly have an additional cycle in addition to whatever structure we had before.
- The precise meaning of this comes from topology.
- Namely, we can represent each graph as a topological structure in space in which:
  - vertices are points, and
  - edges are lines connecting the corresponding points.
- For this structure, cycles are described by the corresponding *homology groups*.

8. A natural question and what we do in this talk

- A natural question is why additional cycles are a good indiction of a cyber-intrusion.
- In this paper, we provide a possible explanation for the relation between cycles and cyber-intrusion.

9. Towards an explanation: what happens in a normal situation

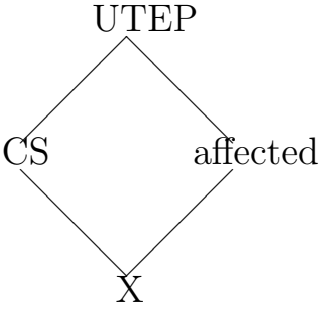
- Usually, computers form a hierarchical structure.
- Here is an example.
- A computer of a faculty from our university's Computer Science (CS) department is a part of CS domain.
- A computer of a faculty from the Department of Teacher Education (TED) is a part of a TED domain.
- Both CS and TED domains are, in their turn, parts of general domain of our University of Texas at El Paso (UTEP), etc.
- Every two clusters in this description are either included in each other or disjoint.
- If we make an inclusion graph, we get a tree, i.e., a graph without cycles.

10. What if we have an intrusion

- Suppose now that we get an intrusion.
- As a result of this intrusion, some computers in different domains become affected.
- What happens if we group computers into clusters – by similarity of their actions?
- We still have the original hierarchical clusters.
- However, we also get a new cluster: of affected computers.
- In this case, when we form an inclusion graph, we get a cycle:
  - in addition to a usual hierarchy-related path from the room to the affected computer,
  - we also have a different path – via the cluster of affected computers.

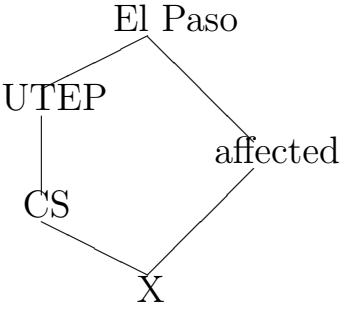
11. What if we have an intrusion (cont-d)

- For example, suppose:
  - that one of CS computers (call it X) is affected (we hope not), and
  - that the intrusion has affected only UTEP computers so far.
- Suppose that affected computers include:
  - some (but not all) computers from CS department and
  - some (but not all) computers from the TED department.
- Then we have the following cycle:



12. What if we have an intrusion (cont-d)

- Similarly, suppose that the intrusion has gone from UTEP to some other computers in our city of El Paso.
- Then, we have a new cycle:



13. Main reference

- H. Jenne, S. G. Aksoy, D. Best, A. Bittner, G. Henselman-Petrusek, C. Joslyn, B. Kay, A. Myers, G. Seppala, J. Warley, S. J. Young, and E. Purvine, "Stepping out of Flatland: Discovering Behavior Patterns as Topological Structures in Cyber Hypergraphs", *The Next Wave*, 2024, Vol. 25, No. 1; also CoRR abs/2311.16154, ArXiv preprint, 2023, https://arxiv.org/abs/2311.16154

14. Acknowledgments

This work was supported in part:

- by the US National Science Foundation grants:
  - 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science),
  - HRD-1834620 and HRD-2034030 (CAHSI Includes),
  - EAR-2225395 (Center for Collective Impact in Earthquake Science C-CIES),
- by the AT&T Fellowship in Information Technology, and
- by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329,