www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 357-363

e-ISSN : 2583-1062
Impact Factor : 7.001

A HOLISTIC VIEW OF TOPOLOGICAL DATA ANALYSIS IN CYBER RISK MANAGEMENT

Ranjit Kumar[1], Dr. Vijay Kumar Pandey[2]

[1]Research Scholar Department of Mathematics Radha Govind University, Ramgarh Jharkhand, India

[2]Research Guide Department of Mathematics Radha Govind University, Ramgarh Jharkhand, India

ABSTRACT

It often happens in cyber security that identifying malicious or unusual behavior requires combining a number of weak symptoms of penetration, each of which would not be cause for concern on its own. Such indicators' trajectory might also be crucial. This makes Topological Data Analysis (TDA), a topic that uses methods from algebraic topology to study the high level structure of data for both exploratory analysis and as part of a machine learning process, a particularly good fit for the challenge of cybersecurity data analysis. We want to draw attention to a new and exciting field that has great promise to advance cybersecurity data science by presenting TDA and examining the work that has been done on its application to cybersecurity.

Keywords: Topological Data Analysis, Cyber security, Persistent Homology, Network Anomaly Detection, Data Breach Prevention.

1. INTRODUCTION

Using mathematical techniques with robust theoretical guarantees, Topological Data Analysis (TDA) enables practitioners to investigate a dataset's topology. Since the global structure of data is roughly its topology, TDA allows us to analyze the "shape" of datasets in ways that are not possible with conventional data analysis techniques. Many of the danger actors' behaviors in cyber security are not troublesome when taken separately; problems only emerge when taken into account collectively (Winding, Wright, and Chapple 2006). Because TDA can identify unusual trends on a worldwide scale, it is especially well-suited to analyze cyber security data. Mapper and persistent homology are the two main TDA approaches. Mapper is a technique that allows for the visualization and study of massive, high-dimensional datasets by embedding them into a much smaller network while maintaining topology. A persistence diagram, a succinct overview of a dataset's topology that may be vectorized and fed into other machine learning techniques, is created using persistent homology. A growing amount of research demonstrates that TDA can achieve state-of-the-art results on a wide range of machine learning problems. For instance, Zhao et al. (2020) attain the state of the art on several graph classification tasks by including persistent homology into graph neural networks. Applications for both TDA strands may be found in cybersecurity. big networks are typical in this field, and Mapper may ethically shrink big datasets so that human operators can more easily see and comprehend the data they are dealing with. In fact, the main use of Mapper that we see in the literature is to condense huge network traffic volumes into graphs that are easier for human analysts to understand. Mapper graph clusters are proven to correlate to anomaly categories (Coudriau, Lahmadi, and Franc¸ois 2016), and analysts may be alerted to traffic worth further study by their closeness to possible anomalies (Bihl et al. 2020). Additionally, persistence diagrams have been successfully used to cybersecurity data. "Little information can be gained from analysing individual records, [as] often the presence of intrusion behaviour or other anomalous activity can only be detected by looking at the aggregate behaviour of related records," according to Winding, Wright, and Chapple (2006). Persistence diagrams are a good fit for cybersecurity because they provide a way to summarize the global behavior of a large number of data. Persistence diagrams have been used to effectively detect abnormalities in network logs (Bruillard, Nowak, and Purvine 2016) and, when combined with convolutional neural networks (CNNs), identify Internet of Things devices from encrypted packet data (Collins et al. 2020). These examples are presented in the literature. This has to do with identifying change points more generally. In fact, research on topological change point detection has been generated under an ongoing TDA for Threat Detection1 research grant (Islambekov, Yuvaraj, and Gel 2019). Prior to implementing TDA, cybersecurity data must be included in a topological tool-compatible manner. This may be accomplished via embedding into R d or by using other methods to embed straight into simplicial complexes, which are the kind of structure needed for TDA. We examine various embedding options made in the literature since they have a significant impact on TDA's performance. These include both general and cybersecurity-specific graph embedding algorithms. This review is organized as follows. An overview of Mapper and persistent homology is provided in Section 2, along with resources for readers who may be interested. We provide a summary of methods for embedding cybersecurity data in Section 3.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 357-363
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

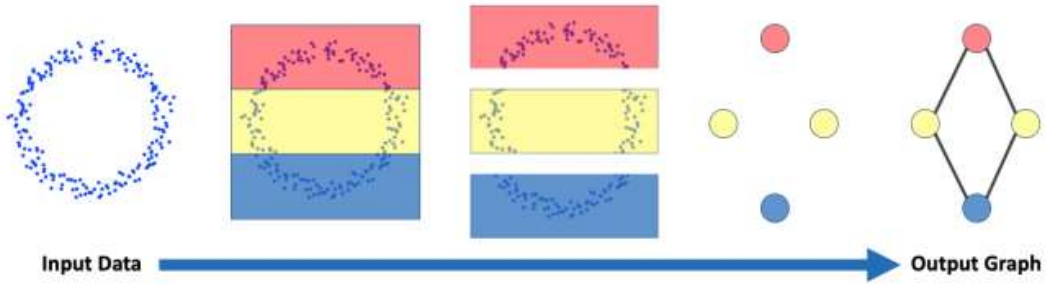


Figure 1: In Mapper data is split up, clustered, then the resulting cluster centres are connected together based on shared nodes, leading to a graph that summarises the dataset in a topology-preserving way.

either as graphs or vectors. We examine the research on Mapper and persistent homology for cybersecurity in Sections 4 and 5, respectively. We provide a summary of studies on cybersecurity graph metrics that are closely related to TDA in Section 6. We wrap up in Section 7, and we provide a list of datasets that readers may find useful in Appendix A. 2.

Overview of TDA

Topological Data Analysis can be split into the Mapper algorithm and the persistent homology pipeline. Although not at first glance related, they both rely on the Nerve Theorem: a result in algebraic topology that guarantees a certain representation of a space preserves its topology (Hatcher 2000, Corollary 4G.3). This theoretical guarantee underpins TDA, allowing representations of data that preserve global structure. In the following we give a brief overview of TDA, but for more details see the textbook by Edelsbrunner and Harer (2010) or the lecture notes from Nanda (2021).

Mapper

The Mapper method (Singh, Memoli, and Carlsson 2007) was the first to use a topological data perspective. It preserves topology while mapping large high-dimensional datasets into graphs. This low-dimensional representation of the data is easier to read than the data itself and may highlight key regions. Figure 1 shows the whole Mapper process, which we propose using to explain the technique. A dataset is partitioned into overlapping sections. This works with any function (the domain dictates what structure you map onto), although hypercubes are most frequent. A clustering technique for each hypercube's data produces nodes. When two clusters from distinct hypercubes share a point from the original data (which happens when they overlap), add an edge between the nodes in the algorithm output. You can translate enormous high-dimensional datasets into a graph that preserves the source dataset's structure using the Nerve Theorem. The method you partition your data and the clustering technique you choose affect Mapper's findings, however trials have proven that with adequate parameter adjustment and domain expertise, it may provide fresh insight into data. Lum et al. (2013) used Mapper to tumors, voting, and NBA player performance. Nicolau, Levine, and Carlsson (2011) employed Mapper to find a novel breast cancer subtype with good survival rates, demonstrating TDA's effectiveness.

Persistent homology

Here's how the persistent homology pipeline works. The point is that we get a (possibly vectorized) persistence diagram that simply summarizes the supplied data's global structure. Use this alone or feed it into a machine learning approach for downstream tasks. We require a simplicial complex filter to calculate persistent homology. A simplicial complex is a collection of nodes, edges, triangles, and other higher-dimensional equivalents. Every simplex contains each of its constituent simplices (i.e., any triangle in the complex contains its edges and vertices), and the intersection of any two simplices is also a simplex. A simplicial complex filtration consists of K0, K1, K2,..., such that K0 ⊆ K1 ⊆ K2 ⊆... (refer to Figure 2). Complex K's parameter is the filtering time or scale. Many approaches exist to generate simplicial complexes using data. The Vietoris-Rips complex is used for point datasets (Vietoris 1927). When k points are pairwise close, we add them as a simplex to the complex. Using K as the -Vietoris-Rips complex offers us a filter of simplicial complexes to calculate persistent homology. Complexes may be made in numerous ways. Cybersecurity may require building complexes from network data, linking nodes that have network connections. Time might be our scale parameter.

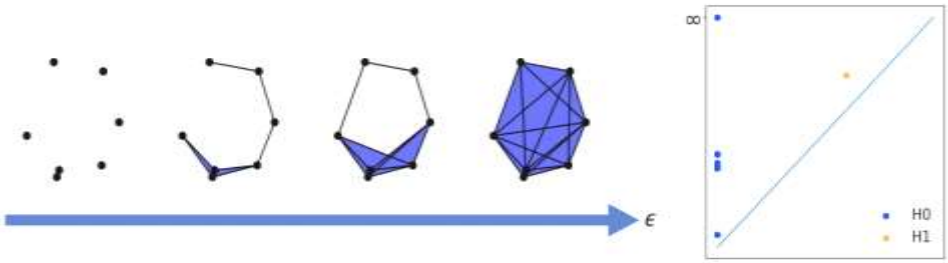


Figure 2: On the left is a Vietoris-Rips filtration, shown with increasing values for .

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 357-363
e-ISSN : 2583-1062
Impact Factor : 7.001

The filtering persistence diagram is on the right. Each blue point in the 0-persistence diagram indicates a linked component. The 1-persistence graphic shows the orange spot as the network connection hole. The persistent homology may be calculated using this equally valid filtering. For each simplicial complex K, we calculate phomology group. This shows K's topology: the 0-homology counts linked components, the 1-homology holes, the 2-homology voids (e.g., a football hole), and so on (Edelsbrunner, Letscher, and Zomorodian 2000). The topology of K may be described via homology groups. However, we must choose a value of when calculating the homology group. In a Vietoris-Rips complex, tiny values of will disconnect points, whereas big values will link everything. The persistence technique lets us consider everything at once, avoiding that option. Summing all p-homology groups yields the p-persistent homology group. This shows how topological properties survive during the filtration: simplices may enter at Ki, creating a hole. At Kj, more simplices may enter the filter to fill the hole. We refer to the hole's creation at time i, its death at time j, and its persistence during j − i. This lets us characterize the filtration's topological properties. Infinity is the death time for features that never expire, such as at least one linked component. Birth and death periods for topological characteristics help visualize filtering homology. In the persistence diagram, the x-axis represents birth time, the y-axis represents death time, and each point is a topological property of the filter. Figure 2 shows the hole in the filter as the orange spot in the 1-persistence plot. Higher above the diagonal in the diagram, a point survives longer and is more likely to reflect a topological feature than noise. The persistence diagram space is not Hilbert, but it has distances like the Wasserstein distance. To easily include persistence diagrams into machine learning operations, embed them. Many methods can vectorize persistence diagrams, although persistent pictures are most frequent (Chepushtanova et al. 2015). This vectorizes the persistence diagram by applying a Gaussian to each non-diagonal point and integrating the surface across a grid. Persistence landscapes (Bubenik et al. 2015) and Betti curves (Rieck, Sadlo, and Leitte 2017) also provide functional embeddings.

Embedding cybersecurity data

Cybersecurity data is often unsuitable for TDA or other machine learning algorithms. Choosing how to incorporate data is crucial in any cybersecurity data science approach. Since most objects we consider reside on networks, much data has a graph form. This graph structure has been respected by some academics and ignored by others, although both have been successful in the literature. In this part, we discuss cyber-specific embeddings that respect and ignore graph structure and universal graph embedding methods. Mapper is usually used to graph-agnostic vectors in cybersecurity TDA literature, even when the starting data has a graph structure. Mapper may be used directly to graphs (Hajij, Rosen, and Wang 2018), however the tools are less developed. Most publications include graph-agnostic data before calculating the Vietoris-Rips complex for persistent homology. However, certain techniques (Collins et al. 2020) employ the network structure and data properties to filter, proving their feasibility.

Graph-agnostic embeddings

Winding, Wright, and Chapple (2006) proposed vectorising network log data by summing numeric fields and counting enumerated fields. They suggested the following feature vectors:

• Source IP, number of destination IP addresses;

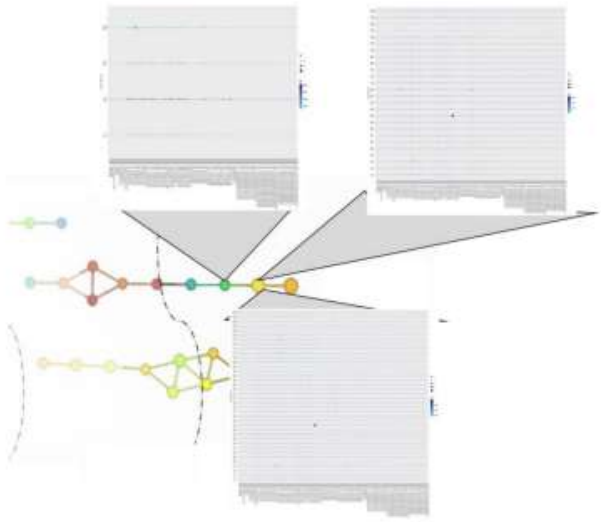


Figure 3: Mapper applied to firewall logs. Nodes in close proximity to those flagged as potentially anomalous could be worth further investigation. Reproduced with permission from Bihl et al. (2020).

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 357-363
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

• destination IP, number of failed access attempts;

• source IP, destination IP;

• destination perspective vector (consisting of destination IP, number of successful accesses, number of failed accesses, count of destination points, number of inbound bytes, number of outbound bytes). This technique has been used in several pieces of anomaly detection research (Jakub and Branisov ˇ a 2015; Bihl et al. ´ 2020). Bruillard, Nowak, and Purvine (2016) proposed embedding NetFlow data into feature vectors that summarise either the cumulative number of packets sent or received by an IP of interest, the total number of packets sent or received by each IP, or the total number of packets sent or received by each IP to an IP of interest. These vectors are created over a window of a given length that advances over the whole dataset. The length of window and increment time is parameters that the authors vary. They show that the choice of window length and increment time can affect the performance on downstream tasks.

Event data as graphs

Aksoy et al. (2019) considered a sliding window of 60 seconds that advanced 20 seconds at a time over event logs. For the events in each window they construct a graph – the specific graphs they consider are listed below.

• Authentication graphs are unweighted graphs built from authentication data that has source user/destination user as edges.

• Authentication failure graphs are as above, but restricted to failed authentication events.

• Process graphs are built from start/stop records of processes. The graph consists of edges between computers and process names.

• DNS graphs are built from DNS lookup events with edges from source computer to resolved computer.

• Flow graphs are built from the records of every network flow event. The edges are between the source computer and the destination computer. Collins et al. (2020) used encrypted packet data to build a filtration. They picked a device of interest and connected it to the last n devices that it exchanged packets with. They induced a filtration by the inter-packet arrival time, a metric that has previously been shown to contain discriminative information when the traffic is encrypted. Once all edges of a 2-simplex were present, they added the 2-simplex (Figure 6a). By doing so they avoided computing the expensive Vietoris-Rips complex, and instead used the natural structure of the data and a relevant feature to induce a filtration.

Graph embeddings

Given the prominence of graph-structured datasets in cybersecurity, we briefly cover R d graph embedding methods. Skip-gram models. Mikolov et al. (2013a,b) improved NLP word embeddings using skip-gram. Skip-gram maximizes the average log likelihood that your model can predict the next word from a phrase. Node2Vec (Grover and Leskovec 2016) and LINE (Tang et al. 2015) optimized this for graph-structured data by maximizing the likelihood of predicting a nearby node from v. Since neighborhoods across multiple hops might become too huge, they are sampled by random walk with various adaptations and sampling methods. Graph neural networks. GNNs combine neighboring features to aggregate feature vectors across nodes. Initially, graph neural networks were classed as spectral-aggregating neighbors in the spectral domain or spatial-aggregating otherwise. True spectrum filters need a costly Fourier basis calculation, hence networks were usually spatial. GNNs may be convolutional, attentional, or message-passing. How neighborhood data is aggregated differs. Fixed edge weights as coefficients for node feature vectors compute a linear sum in convolutional networks (Kipf and Welling 2017). Attentional networks develop a coefficient-based edge attention function (Monti et al. 2017). Most general, message-passing networks learn one function on nearby feature vectors and aggregate them (Battaglia et al. 2016). Large graph embedding. Cybersecurity data may be huge, and our methods cannot scale to massive graphs. GraphSage (Hamilton, Ying, and Leskovec 2017) samples neighbors instead of calculating the neighborhood, and minibatches update gradients for a restricted number of nodes. This lets GNNs handle graphs with over 100,000 nodes. clusterGCN (Chiang et al. 2019) and graphSAINT (Zeng 2020)

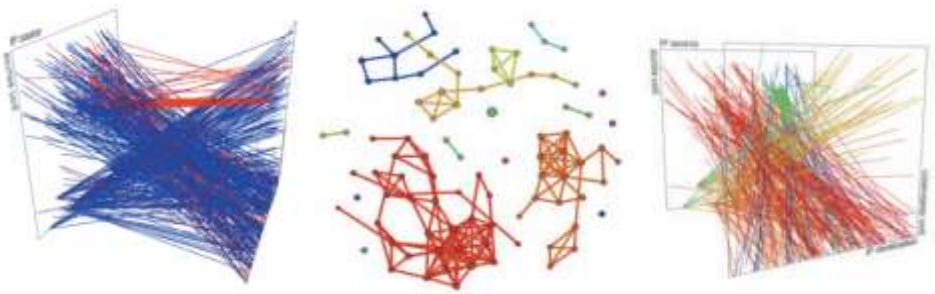


Figure 4: Data collected by a network telescope is shown on the left.

In the middle diagram is data Mapper graph. The large green dot is traffic trying to exploit a router vulnerability, the red component is Telnet or SSH traffic, the orange component is a sparse port scan from a single address, and the yellow component is a randomised scan and noise. The Mapper graph colors starting data on the right. Despite Mapper identifying assault categories, the data is chaotic. Reproduced with permission from Coudriau, Lahmadi, and Franc´ois (2016).

Remove unnecessary calculations to speed up GraphSage. Large graph processing techniques are available from Facebook and Twitter. Twitter limited big GNNs to one layer (Frasca et al. 2020), allowing massive computational speedups and parallelization. The lack of depth seems to affect network performance, since the authors of this study indicate in a blog post2 that it is best utilized as a rapid benchmark for other models. Facebook suggested Pytorch Big Graph (Lerer et al. 2019). PBG operates on multi-relational graphs with distinct edges. It provides techniques for multi-relational GNNs to function on graphs with billions of nodes and trillions of edges.

Mapper for cybersecurity

We examine Mapper cybersecurity literature by application domain in this section. It is mostly used for investigation and visualization, but it may also give quantitative data. Anomaly detection. Mapper was used on business firewall logs by Bihl et al. (2020). Section 3 discusses Winding, Wright, and Chapple (2006)'s graph-agnostic embedding method (counting categorical fields and summing numerical fields). The Gutierrez et al. (2018) histogram matrix (HMAT) was calculated after embedding the logs. This was meant to help analysts see unusual logs over time. Bihl et al. (2020) assigned an HMAT to each mapper graph node, thus anomalous nodes might alert researchers to neighboring nodes that may be worth investigating (Figure 3). This assertion is unverified, and the tool is portrayed as an experimental approach rather than a reliable anomaly detector. Mapper was used by Rizzo (2020) to visualize intrusion detection system data. Network telecopes. Analysis of non-host traffic may be done using IP ranges without active domains. This communication is presumably malicious since the devices listening to it are fake (Fachkha and Debbabi 2015). The literature calls such traffic internet background radiation (IBR) and listening equipment network telescopes. Coudriau, Lahmadi, and Franc¸ois (2016) found port scans and DDoS assaults using Mapper on network telescope data, surpassing typical clustering techniques when attacks don't cover all potential ports or IPs. Mapper uncovered several port scans in the data, but only a tiny percentage were found by Suricata's ruleset, showing that Mapper can outperform industry approaches. The connected components of the Mapper graph of source/destination IP/port data from a network telescope showed that most components represent a distinct attack, including the attempted exploitation of a known router vulnerability (Figure 4). Narita (2021) constructed Mapper graphs using network telescope data but just described them, without linking them to attack types. Traffic detection on Tor. Networks may promptly shut down and probe encrypted Tor traffic. van Veen (2018) studied Mapper, an unsupervised method for visualizing and categorizing network Tor traffic. Some qualitative assessments show that Tor traffic is on the Mapper graph's extremes. Payments for ransom. Akcora et al. (2020) created a Bitcoin transaction Mapper graph. If a specific percentage of addresses in a Mapper graph node get ransomware payments, the remaining addresses' risk ratings grow. Repeating the map and setting the risk threshold yielded suspicious Bitcoin addresses. This method detected suspicious addresses better than DBSCAN and XGBoost. An attack graph. The Mitre ATT&CK® architecture classifies computer network attacks as reconnaissance, delivery, exploitation, operation, data collecting, and exfiltration. This series of events naturally forms an attack graph. Navarro et al. (2018) used topological methods to analyze this network. They created an attack graph using context, attack patterns, and events. The reduced attack graph was mapped in hopes that people may detect recurrent patterns across assaults.

Persistence for cybersecurity

Persistence diagrams record topological data information that may be utilized for data analysis or vectorized and fed into machine learning systems for prediction and analysis. This section shows that it can discover and classify anomalies more quantitatively than Mapper. Anomaly detection. Bruillard, Nowak, and Purvine (2016) detected abnormalities using NetFlow data distance between persistence diagrams. Vectorising NetFlow data using a sliding window over packet counts (as discussed in Section 3) yielded a baseline persistence diagram from feature vectors B. The Wasserstein distance between B and B ∪ {x} persistence diagrams was calculated for each vector x. They observed that spikes in topological dissimilarity indicate anomalies (Figure 5). Distance increases topological dissimilarity. Activity forecast. Gabdrakhmanova (2018) summarized Euler persistence diagrams. This is the alternating sum of the Betti numbers, which are represented by the total persistence of points in persistence diagrams, rising by dimension. Based on recent behavior at a network switch (say, an ISP), they wanted to anticipate future activity. Seven days of bits/minute data. They created persistence diagrams for each two-hour piece of data. The Euler characteristic of that epoch was determined from it. They used Euler features to predict future activity in a neural network but did not analyze it. IoT fingerprinting. Postol et al. (2019) found that persistent homology classifies partial and noisy IoT data successfully.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 357-363
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

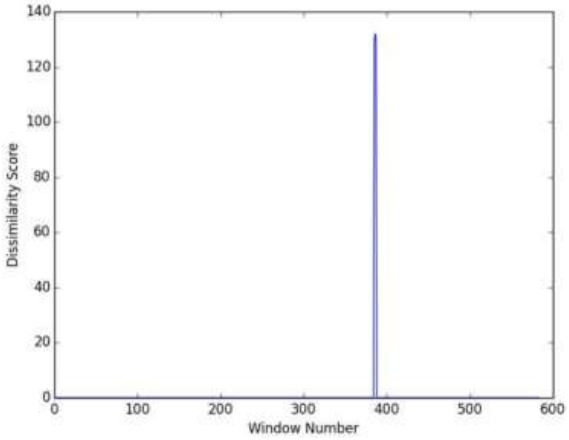
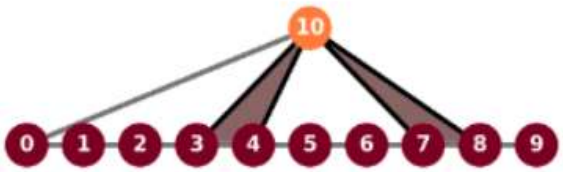


Figure 5: The spike in topological dissimilarity indicates a predicted anomaly. In fact, this was a port scan. Reproduced with permission from Bruillard, Nowak, and Purvine (2016).

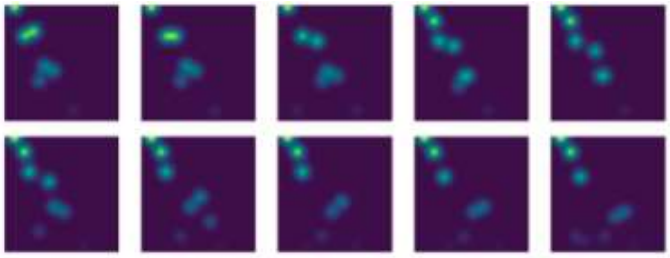
Networked IoT devices. The persistence diagrams of embedded IoT network traffic were calculated, and feature selection and logistic regression on functional embeddings of the diagrams classified cameras, sensors, and multifunctional devices like tablets. This works nicely with monthly data but not with daily data. Collins et al. (2020) also studied IoT data, but they used the natural network structure and filtered with the inter-packet arrival time (IAT), which has been shown to be useful for traffic classification and anomaly detection. They avoided computing the costly Rips complex by leveraging the data's inherent structure (Figure 6a). Next, they calculated the filtration's 1-persistent homology. Finally, they vectorized persistence diagrams to persistence pictures (Figure 6b) to train a CNN. Higher-dimensional topological properties helped them identify IoT devices with excellent accuracy, recall, and precision.

Graph metrics for cybersecurity

Aksoy, Purvine, and Young (2021) introduced a new centrality measure for graphs based on the derivative of the graph Laplacian. The kernel of the Laplacian is exactly the 0- homology group, whilst the non-zero image is known to contain rich geometric descriptors of the graph. Therefore by using the Laplacian we capture both topological and geometric information about the network. Their centrality measure can detect synthetic anomalies that have been injected into the network data. Another measure to compare graphs is the relative Hausdorff distance. This offers a fast but nuanced way to compare two graphs based on their complementary cumulative degree histograms (CCDH). The CCDH of a graph G is (N(k))∞ k=1, where N(k) denotes the number of vertices.



(a) A simplicial complex built from packet data



b) Examples of 1-persistence images of encrypted packet data.

Figure 6: The 1-persistent homology of packet data can be used to accurately fingerprint IoT devices on networks, even when the traffic is encrypted. Reproduced with permission from Collins et al. (2020).

in G with degree at least k. Comparing the CCDHs gives us the relative Hausdorff distance, which Aksoy et al. (2019) used to detect anomalies in graph sequences built from the Los Alamos dataset (see Appendix A). They built authentication graphs, authentication failure graphs, process graphs, DNS graphs, and flow graphs. Over longer time windows, they found that a spike in pairwise RH distance is indicative of a red-team event.

2. CONCLUSION

This study introduces Mapper and persistent homology and discusses its cybersecurity applications. Mapper is mostly used for visualization, helping analysts understand massive datasets and recommend data points for future inquiry. These approaches are challenging to assess since the articles don't include analysts' input on how effective the visualizations are. Coudriau, Lahmadi, and Franc¸ois (2016) divided network telescope data into Mapper graph clusters, indicating distinct assault types. Mapper can classify network abnormalities unsupervised. Persistence homology works on cybersecurity data too. Bruillard, Nowak, and Purvine (2016) used the 0-persistent homology to identify abnormal occurrences in network data: a spike in topological dissimilarity suggested port scans or DDoS assaults in network records. Collins et al. (2020) trained CNNs using 1-persistence photos to build simplicial complexes from the data's inherent network structure. Higher-order topological characteristics may properly fingerprint machines using topological representations of encrypted network data, proving their cybersecurity potential. Topological data analysis in cybersecurity has great potential. Current work on persistent homology shows that it can exploit TDA's ability to succinctly summarize global structure for cybersecurity challenges. We hope this evaluation draws cybersecurity AI researchers' attention to TDA.

3. REFERENCES

[1] Akcora, C. G.; Li, Y.; Gel, Y. R.; and Kantarcioglu, M. 2020. BitcoinHeist: Topological Data Analysis for Ransomware Prediction on the Bitcoin Blockchain. In Bessiere, C., ed., Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, 4439–4445.

[2] Aksoy, S. G.; Nowak, K. E.; Purvine, E.; and Young, S. J. 2019. Relative Hausdorff distance for network analysis. Applied Network Science, 4(1): 1–25.

[3] Aksoy, S. G.; Purvine, E.; and Young, S. J. 2021. Directional Laplacian Centrality for Cyber Situational Awareness. Digital Threats: Research and Practice (DTRAP), 2(4): 1–28.

[4] Battaglia, P. W.; Pascanu, R.; Lai, M.; Rezende, D.; and Kavukcuoglu, K. 2016. Interaction networks for learning about objects, relations and physics. NeurIPS. Bihl, T.; Gutierrez, R.; Bauer, K.; Boehmke, B.; and Saie, C. 2020. Topological Data Analysis for Enhancing Embedded Analytics for Enterprise Cyber Log Analysis and Forensics. 1937–1946.

[5] Proceedings of the 53rd Hawaii International Conference on System Sciences. Bruillard, P.; Nowak, K.; and Purvine, E. 2016. Anomaly Detection Using Persistent Homology. In 2016 Cybersecurity Symposium (CYBERSEC), 7–12.

[6] Bubenik, P.; et al. 2015. Statistical topological data analysis using persistence landscapes. J. Mach. Learn. Res., 16(1): 77–102.

[7] Chepushtanova, S.; Emerson, T.; Hanson, E.; Kirby, M.; Motta, F.; Neville, R.; Peterson, C.; Shipman, P.; and Ziegelmeier, L. 2015. Persistence Images: An Alternative Persistent Homology Representation.

[8] Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 257–266.

[9] Collins, J. R.; Iorga, M.; Cousin, D.; and Chapman, D. 2020. Passive Encrypted IoT Device Fingerprinting with Persistent Homology. UMBC Faculty Collection.

[10] Coudriau, M.; Lahmadi, A.; and Franc¸ois, J. 2016. Topological analysis and visualisation of network monitoring data: Darknet case study. In 2016 IEEE International Workshop on Information Forensics and Security (WIFS), 1–6.

[11] Dray, G.; Raissi, C.; Brissaud, J.; Poncelet, P.; Roche, M.; and Teisseire, M. 2007. Web Analysis Traffic Challenge: Description and Results. In Discovery Challenge ECML/PKDD.

[12] Edelsbrunner, H.; and Harer, J. 2010. Computational Topology - an Introduction. American Mathematical Society. ISBN 978-0-8218-4925-5.

[13] Edelsbrunner, H.; Letscher, D.; and Zomorodian, A. 2000. Topological persistence and simplification. volume 28, 454 – 463. ISBN 0-7695-0850-2.

[14] Fachkha, C.; and Debbabi, M. 2015. Darknet as a Source of Cyber Intelligence: Survey, Taxonomy and Characterization. IEEE Communications Surveys & Tutorials, 18: 1–1.

[15] Frasca, F.; Rossi, E.; Eynard, D.; Chamberlain, B.; Bronstein, M.; and Monti, F. 2020. Sign: Scalable inception graph neural networks. ICML workshop on Graph Representation Learning and Beyond.