A Literature Review of GNN within the field of Cybersecurity

Thesis

Andreas Philip Westh

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
A Literature Review of GNN within the field of Cybersecurity

Theme:
Cybersecurity

Project Period:
Fall Semester 2023

Project Group:
Individual

Participant(s):
Andreas Philip Westh

Supervisor(s):
Marios Anagnostopoulos
Pere Barlet-Ros

Copies: 1

Page Numbers: 23

Date of Completion:
November 29, 2023

Abstract:

A brief literature review on graph neural networks and their application to detection malicious domains.

Contents

1 Introduction **1**
1.1 Problem Formulation 1
1.2 Contribution(s) 2

2 Background **3**
2.1 DNS 3
2.1.1 Traffic Analysis 4
2.1.2 Security Aspects of DNS 5
2.1.3 Malicious DNS Activity 5
2.2 Machine Learning 6
2.3 Graph Neural Network 7
2.4 Data Enrichment 8

3 Literature Review **11**
3.1 Papers 12

4 Conclusion **19**

Bibliography **21**

A Appendix A name **23**

Chapter 1

Introduction

Across the Internet a technology remains a persistent problem due to its configuration: The Domain Name System (DNS) protocol, especially from a security standpoint. Using this protocol, malicious actors have through countless ways exploited its various vulnerabilities, especially those associated with creating domains pretending to be legitimate.

One way to mitigate this is via machine learning models that will identify and block malicious domains. Which is what this paper sets out to do, specifically using graph neural networks (GNN). These networks have in the past shown a lot of promise in how resistant they are to poisoned data. Creating a GNN model for filtering this malicious data requires careful consideration of several factors. Learning method, what attributes to apply, and more, along with the quality of the ground truth used for the model.

This paper lays out a literature review of the current state of the art of GNNs, especially how they pertain to malicious domain detection.

1.1 Problem Formulation

Much of this section surrounds the questions of *how* to go about creating a GNN that appropriately filters malicious DNS data. What areas surround the appropriate application of this.

Areas of note are the learning method within machine learning, both supervised and hybrid (a combination of supervised and unsupervised) have proved viable for GNNs. Another area is uncovering which attributes within the data is worth putting into the model, there are many DNS at-

tributes that can prove viable. Data enrichment is tied into the data as well, which will also be necessary to cover. Finally, ground truth will be important to touch upon, where and what kind of data is being collected creates challenges to be aware of (for instance blacklist data is very relevant, but some can be limited in scope by focusing solely on spam).

All these considerations need to be taken into account when creating a GNN model which accurately predicts and filters malicious domains. This is a type of anomaly detection, which is able to identify specific attack patterns. The key challenge in this has been how to achieve the necessary accuracy of its predictions for it to be viable.

And so, with the considerations of how to implement a GNN and the challenges of making the model do accurate predictions in mind, the main question becomes:

How are GNNs currently used to aid in malicious domain detection?

With the following sub-questions:

- What does the literature within the field tell?
- How have models improved?

1.2 Contribution(s)

This papers contribution is listing an overview of sixteen papers within the field.

Chapter 2

Background

For ncessary contextualization, what follows are related and underlying topics for this paper. These provide an overview within the topics DNS, graph neural networks, and data enrichment.

2.1 DNS

The Domain Name System is a fundamental component of the Internet's infrastructure, its main role being the translation of human-readable domain names into their actual IP addresses, which is the part that computers comprehend. This process of translation is what enables computers and servers to communicate over the Internet. As such people aptly refer to DNS as the phone book of the Internet.

A domain name consists of three parts: A Top-Level Domain (TLD), also known as domain extensions, these are for instance '.com', '.dk', '.org' and generally known names or connected to a country. Second-Level Domain (SLD), what comes before the TLD, in the website name 'blog.aau.dk' the 'aau' part would be the SLD; this part is what users register on behalf of an organization or person. The third part is the subdomain and comes before the SLD, it is optional and used for navigation of the website, in the previous example the 'blog' part would be the subdomain. There is a fourth part known as the Root Domain, it is usually hidden and only there implicitly as a dot at the end of a website name, meaning that in the previous example the domain is actually 'blog.aau.dk.' but generally the user will not see it.

Another noteworthy part of DNS is its decentralized nature. There is

an incredible amount of DNS servers across the entire world, all of which contain a relatively small amount of the domain name space. Whenever someone makes a DNS query, or as it is more commonly known 'visit a website', this query is not passed to a central server. What happens instead is that it starts at the top of the hierarchy with a root server, then moves on to a TLD server for that part of the query, and then lastly visits the authoritative DNS server which contains all the information of the domain. This decentralized attribute of DNS, along with its hierarchical structure, ensures reliability, redundancy, scalability and speed across the Internet.

However, it is not without its issues. Many forms of attacks exploit DNS' inherent simplicity via for example DNS spoofing or cache poisoning among others (elaborated on later). As such it becomes important to dissect and analyze DNS data related to attacks to find any patterns that might present themselves in order to combat these attacks.

2.1.1 Traffic Analysis

DNS traffic analysis is a powerful tool in the realm of cybersecurity. Every device connected to the Internet relies on DNS to resolve domain names to IP addresses, generating a substantial amount of DNS traffic. This traffic contains rich, insightful data about the behaviors and patterns within a network, making it a valuable resource for detecting and mitigating cyber threats.

Monitoring and analyzing DNS traffic allows security professionals to identify unusual patterns that may signal a cyber-attack. For instance, a sudden spike in DNS requests to a specific domain could indicate a distributed denial-of-service (DDoS) attack. Similarly, frequent DNS queries to a newly registered or otherwise suspicious domain may suggest a malware infection or command-and-control (C&C) activity.

However, manual analysis of DNS traffic is a daunting task due to the massive volume and dynamic nature of the data. Moreover, sophisticated cyber threats often employ evasion techniques, such as fast flux or domain generation algorithms (DGAs), to obfuscate their activities in DNS traffic.

Therefore, automated and intelligent analysis methods are required. Machine learning and, more specifically, graph neural networks (GNNs) represent promising solutions to this challenge. By modeling DNS data as a graph, GNNs can capture complex relationships and dependencies between different entities (e.g., domains, DNS servers). This allows for more accurate detection of malicious activities, even when traditional, pattern-based

detection methods fail. GNNs can learn from the underlying structure of DNS traffic, making them a powerful tool for cybersecurity.

2.1.2 Security Aspects of DNS

Given the hierarchical structure, there is a natural trust and delegation embedded in the system. This is due to each level being responsible for their domain and any subdomains. Which also consequentially means that if a malicious actor gains control over a DNS server they could redirect or intercept all traffic from this domain and its subdomains. From here a vantage point opens up for attackers to follow-up with, like phishing, DDoS, or even man-in-the-middle. The decentralized structure of DNS also comes with its security issues. Most commonly exploits involving DNS servers focuses on the trust it has in other servers' responses, resulting in DNS spoofing. While there exists countermeasures, such as DNSSEC, DNS over HTTPS, and DNS over TLS, none of these have seen widespread adoption. Furthermore, they have come with their own set of issues and in some cases attacks. There is no silver bullet for this issue.

Another approach an attacker takes when exploiting DNS is via covert channels, especially for data exfiltration. Commonly referred to as DNS tunneling, this technique allows the user to send any data over DNS requests and responses, which bypasses most of the standard network security measures like Firewalls, IDS, and IPS.

To combat these a new and potent countermeasure is that of a GNN. By taking DNS data and turning it into a graph, treating domains and DNS servers as nodes and any interactions as edges, a GNN is able to learn intricate patterns and connections in the graph. What this specifically counteracts is the aforementioned decentralization of DNS. DNS servers have a limited understanding of its domains and possible connections, making any malicious activity that move across servers difficult. GNNs, with their acquired patterns from the DNS data, are able to detect malicious activities, across DNS servers, even if these attackers are using complex evasion techniques .

2.1.3 Malicious DNS Activity

There have been a variety of unique DNS-oriented exploits over the years. What follows, is a non-exhaustive list of popular attacks and techniques that malicious actors have used for their own gains:

- **DNS Spoofing or Cache Poisoning:** In this type of attack, the attacker corrupts the DNS resolver's cache with false information, causing the resolver to direct traffic to an incorrect IP address, often controlled by the attacker. This can lead to users being directed to malicious websites masquerading as legitimate ones.[1]
- **DNS Hijacking:** Similar to DNS spoofing
- **DNS Amplification:** This is a Distributed Denial-of-Service (DDoS) attack that exploits DNS servers. An attacker spoofs their victim's IP and sends a request to a large amount of DNS resolvers, causing the resolver to respond to the victim. This response is much larger than the spoofed request, amplifying the attack.
- **DNS Tunneling:** This technique bypasses network security measures by encapsulating non-DNS traffic within DNS protocol. It sees legitimate use, but also use with data exfiltration and command-and-control communications for malware.
- **Fast-Flux:** More of a technique that can see use as an attack. The malicious actor's DNS records are ever-changing to different IP addresses of compromised hosts within their botnet. These act as proxies to host a service like selling malware, hosting illegal content, or pretending to be a normal site for phishing attacks. The use of a large scale botnet makes it difficult for law enforcement to track down the source.
- **NXDOMAIN Attack:** In this type of attack, the attacker floods the DNS server with requests for records that do not exist, with the intention of consuming server resources and causing a denial-of-service.

These attacks can be detected via a GNN, as one paper demonstrates with fast-flux [17].

2.2 Machine Learning

Machine Learning at its core is taking a large amount of data and training a program to learn a pattern contained within the data. This is often then make predictions of data of the same kind as the one fed. This labeling has

[1]While referred both as DNS spoofing and cache poisoning the rest of this paper will refer to it as DNS spoofing.

the potential to automate a great deal of tasks across a multitude of fields and also perform tasks that were not possible before. A current notable example of the latter would be large language models like ChatGPT.

A different approach in machine learning was introduced in 2008, in their seminal paper "The Graph Neural Network Model" by Franco Scarselli et al. [15]. In this paper they suggest a more focused effort on the actual connections between nodes of data and in tandem what potentially useful patterns arise.

2.3 Graph Neural Network

A GNN is a neural network that centers around the abstraction of graphs being a valuable way to interpret data and primarily make predictions. These models have two fundamentals parts: *Nodes* represent data and *edges* represent a relation between these data points. If it is valuable to the GNN, there is also the option to have the edges be directed, which is when they will point to a specific node (for instance when a parent node will point at a child node). This can be useful

GNNs are similar to neural networks when it comes to processing and learning from graph datasets. The root of GNNs is to create a local and global structure of a graph, and then continuously cluster and change the information of neighboring nodes. This is also called message passing or alternatively neighborhood aggregation, and its purpose is for the GNN to pick up on complicated patterns along with any connections within the graph's data.

GNNs have seen a lot of success in domains such as:

1. **Social Network Analysis:** GNNs have correctly predicted user behavior, calculate social influence, and uncover groups in social networks among other things.

2. **Bioinformatics:** GNNs have been able to predict protein function, molecular interactions modeling, along with analyzing biological pathways.

3. **Recommender Systems:** GNNs have modeled user to item interactions successfully, catch item similarities, and create personalized recommendations.

4. **Computer Vision:** GNNs have managed scene understanding, detected objects, and image segmentation tasks via the connections between objects and the regions within the image.

5. **Natural Language Processing:** GNNs have correct modeled syntactic and semantic connections between words, sentences and documents, all of which has meant advancements for sentiment analysis, machine translation, and document classification.

6. **Cybersecurity:** GNNs have seen use in detecting malicious domains, processing network traffic patterns, and uncovering any anomalies in system logs.

While there has been many advancements with regards to GNNs, there are still many obstacles. Of note is scaling to large graphs, how to handle dynamic and evolving graphs, implementing edge features and attributes. Solving any of these issues will lead to GNNs being significantly more useful in the real world and its problems where graph datasets are available.

2.4 Data Enrichment

Data enrichment is the act of enhancing the quality of raw data, done via adding additional information in some way, done both automatically and manually depending on the specific method and tool. The purpose of this is because datasets will commonly contain either unnecessary information or parts of the data will lack the necessary information, for instance with an attribute not containing a value in all rows or columns; additionally there may be biases in some way that will need to be adjusted. All of this serves to make the data more palatable to a machine learning model, so as to speed the process up, improve its accuracy or aid its understanding of potentially complicated problems. With network data this is especially true as a lot of it can have limitations via its collection method, meaning data enrichment is especially important for the purposes of this paper.

When it comes to data enrichment there are three areas that are worth mentioning. These are data processing, feature engineering and integration of external data sources. These all focus on improving the quality of the data in different ways.

Data processing is a set of methods that adjusts the raw data so that it is more appropriate for the model to be trained. Adjustments include how to

deal with certain values being missing from some of the data, the removal of data (disproportional values, outlier elements, noise, etc.), and similar tasks. In many ways this step can be seen as preprocessing, where these are steps dealing with the handling of the data before the machine learning model enters the picture.

Feature engineering is the method where you modify or create new features in the raw data to better align with solving the underlying issues you are trying to tackle within the model. While similar to data processing there is a more intricate approach to feature engineering, for instance involving reducing dimensionality or mathematical transformations of various sorts. In short, its application is mainly to further the efficacy of a given feature from the data.

Integration of external data sources is finding information from outside sources that enhances your dataset. This data can come from many places, from marketing data to geolocation and more. What is important is of course it having an appropriate relation to your data and the problem you are trying to solve with your model. Its main challenges consists of finding, combining and justifying the additional data. Different datasets can easily create issues for algorithms (and the programmers) to process correctly.

With these three in mind, there are many subtopics to cover within data enrichment, all of which have their place throughout this thesis to varying degrees.

Chapter 3

Literature Review

When it came to gathering relevant material we followed a systematic literature review [10]. This strategy resulted in the following approach:

- Preliminary search based on key terms from known papers [14] [18].
- A key criterion for the papers were to focus on fairly recent papers and therefore keeping the amount before 2010 to a minimum, mainly due to the rapid development within the field.
- Any interesting results would be checked for predatory journals list.

These criteria provided clear guidelines for finding and selection of appropriate papers for the following three areas: DNS, GNN, and data enrichment.

A Survey on Malicious Domains Detection through DNS Data Analysis

This survey by Yury Zhauniarovich et al. [18] gives a thorough and structured explanation of how DNS data is being used when trying to uncover malicious domains. Of special note is how it shines a light on the importance of malicious domains for an attacker to successfully conduct any attack over the internet, and underlines an extensive comprehension of how these domains change across the entire timeframe, throughout DNS queries and responses in the traffic. In this paper, its authors lay out a framework for detecting malicious domains regarding DNS data. They divide the current strategies into three parts:

1. Sources of DNS data and any enrichment done to this data.

2. The method of analyzing the data.

3. Evaluation strategies and metrics

These divisions help in explaining the strategies along with their potential challenges and positives. A significant hurdle in the paper is one of the acquiring data. Collecting large DNS datasets, a keystone to performing the task of detecting malicious domains, is rarely available to the public, due to several reasons that are mainly of the legal and bureaucratic variety. Additionally, researchers have not reached a consensus for creating ground truth via public intelligence, which means there are a lot of different approaches being practiced instead. The paper also discusses primary issues pertaining to the algorithm being used. That is to say the resilience of features, how well the algorithm can adapt to evasion techniques, as well as how best to interpret results. As far as the third attribute is concerned, the paper conveys the point that the field has not created the structure nor the foundation for empirically evaluating the robustness of these malicious domain detection schemes. Regardless of these concerns, it concludes on a positive note that the information laid out within this paper provides a methodology for a more homogenized approach for all further development in this field.

This paper is especially relevant for this thesis as it gives a great introduction to the challenges within the field as where it is at now. Several of the challenges they outlined proceeded to cause delays and other negative effects for this thesis.

3.1 Papers

Nearly all papers reviewed had a focus on DNS in some way. Utilizing DNS data is possible in an assortment of ways. This became one of three parts to explore in this review. The two others, like the background alludes to, is data enrichment and the GNN.

"Graph Theoretical Models of DNS Traffic" [3] modelled their graph on DNS requests and responses. They focused on the TLD of '.it' due to scope and to observe a particular region's traffic for any behaviors of note. Lastly, they made particular note of the TTL attribute of DNS records for their graph modelling. The authors normalized how often DNS resolvers would query a domain server based on its TTL, a higher TTL would result in fewer queries and vice versa. This ensured that their graph would be more realistic to real DNS traffic. This led to three undirected bipartite graphs, which

models activity between resolvers and domains. GALL, created edges between resolver and domain if there were at least a single DNS query; it focused on 'A', 'AAAA', and 'MX' records. GWEB, centered around web traffic, creating edges for 'A' and 'AAAA' queries where there were no host, or it preceded 'www' or 'web'. GMX represented email traffic and had edges for 'MX' queries. A last common-neighbors graph monitored the relationship of domains and resolvers to reveal subtle connections in the DNS traffic not observed by the other three graphs.

Another method is analyzing DNS queries where they failed. "Identifying Suspicious Activities Through DNS Failure Graph Analysis" [7] presumed that a failed DNS query is a suspicious activity of some form. Failed meaning that the DNS did not resolve and is often an indicator of malicious activity like trojans, bots, etc. These requests were logs from a large campus network and hence made for a good real world environment. This data was then utilized to create DNS failure graphs that categorized the kinds of failures. These graphs consisted of hundreds of isolated components, with a few being notably larger. From these larger components, the authors extracted the sub-graphs that had valuable patterns of any suspicious activity between hosts and domains.

The paper "Lean on Me" [2] uses DNS data in a collection with other services focusing instead on the interconnectedness of the Internet's services. It does so by analyzing 2.5 trillion queries from both passive and active DNS datasets. With active DNS data, the authors used the tool 'dig' to query DNS records, specifically for the four record types 'A', 'NS', 'MX', and 'CNAME'. This is a synthetic alternative where the authors made these queries compared to passive DNS data, which in this case was a log of real DNS queries made over a period of 2.5 years. The addition of the active dataset was to supplement the passive dataset that was lacking in its 'NS' and 'MX' records. They did this to enable their goal of mapping users'service dependency relationship offered by third party providers. This led to a graph that formed a connection with directed edges between domains if one of them utilized a service of the other, which led to discoveries of chain failures. Of particular note is the metric they introduced "Support Power" which tells how much a node is dependent on another node due to a service it provides.

"Detecting Malicious Domains via Graph Inference" [11] gathered HTTP proxy logs over seven months from a total of 98 servers all over the world. As such, any DNS data was a byproduct of the HTTP requests and responses. There was a minimal use of ground truth information, meaning

a select set of domains labelled malicious or benign, leaving the majority unknown. They did this for the necessity of follow-up analysis via potent inferences on the unknown majority. Chief among these inferences are belief propagation which estimated all domains' likelihood of being malicious. This is an iterative process that simultaneously enriches the data and detects unknown malicious domains. This produced a host-domain graph, analyzing hosts' IPs in an enterprise environment and what domains they connect to. This creates two types of nodes, host and domain nodes for corresponding parts. Edges are connections between these two kinds. Domain nodes are specifically representing second-level domains and was done due to these presumably being responsible for a domain's and its sub-domain's security. Resulting in more paths and in turn information propagation being more likely between nodes. As an aside, since the IPs of a enterprise are transient, hosts were very likely to represented by more than one node of the graph.

"Detection Of Malicious DNS and Web Servers using Graph-Based Approaches" [6] used historical network traffic from a Microsoft dataset related to the Novidade campaign. They did this as their goal of the paper was to gather information about DNS hijacking attacks specifically. They improved their data by removing irrelevant data. These were mainly IP attributes, more importantly they ended up removing anything not connected to the main component of their graph, making it more coherent. The product of this being a bipartite graph. This is two sets of nodes, one for the DNS servers' and the other for web servers' IP addresses, with edges that intersect between the two sets. These edges were drawn when a DNS server redirected a victim's IP to a web server. As a result, no edges connected nodes of the same set. This setup provided easy analysis of the relationship between these two sets, using methods like community detection, node2vec and belief propagation.

The authors of "Attributed Heterogeneous Graph Neural Network for Malicious Domain Detection" [17] passively extracted DNS traffic from logs over two weeks on a university network. Additionally, over a five month period, they collected twelve publicly available datasets of known malicious domains, from websites like URLhaus. These malicious lists totaled 0.9 malicious domains. Theyenriched their data in a number of ways. They had 12 lists of known malicious domains, which when cross-checking with their DNS data revealed 2.277 malicious domains already existing within their network. They also utilized VirusTotal's API for confirmation of any domains labeled malicious. Lastly, they used a known list of benign do-

mains, Alexa top 1 million domain list, to improve that corresponding part of the data. This resulted in the Heterogeneous graph, seeing use in this appropriately named paper, where they created their own particular semi-supervised version called GAMD (GNN Anti-Malicious Domain). The general idea translates into there being multiple different types of nodes and edges, with complex and diverse relationships making up the overall graph. The purpose of this is to provide a more in-depth analysis method, which the creators of GAMD argue it does within the field of malicious domain detection. According to their tests, on the same hardware, their model outperforms its state-of-the-art "rivals" GAT and FANCI in three of the four measurements.

"Automating Botnet Detection with Graph Neural Networks" [19] did not collect DNS directly either but instead focused on network traffic data in general. Another paper [19] did a clever thing to enrich their data with botnet sources. While utilizing the 2018 IP real world network traffic from CAIDA, they supplemented it with real botnet traffic (external to the dataset), and from here embedded synthetic botnet traffic onto the regular network traffic A standard GNN also saw use in a project [19].

In "Deepdom: Malicious Domain Detection with Scalable and Heterogeneous Graph Convolutional Network" [16] its authors gathered comprehensively their dataset from a collection of DNS logs, DNS traffic, passive DNS (pDNS) datasets, and WHOIS dataset(a famous database for DNS data). Passive DNS datasets in this context means a special form of passive DNS collection that only records select attributes from DNS messages. It was these four methods that ensured cross-verification of data, WHOIS and pDNS additionally provided more context of the domains. This culmination in the creation of a meticulous view of the data and resulted in Deepdom, a heterogeneous GCN. This is actually two graphs. Domain Resolution Graph shows domains resolving to IP addresses. While Domain Graph handles the complex representation of relationships between domains and IPs, with its edges weights showing how strong an association between two domains are.

"Discovering Malicious Domains through Passive DNS Data Graph Analysis" [8] gathered its DNS data from sensors within interconnected servers. The authors targeted 'A' records which provides fundamental parts of IP and DNS data, like how often a domain resolved to a specific IP within a given timeframe. This data then saw enrichment via a domain graph, which has the basic intuition of the stronger association there is between two domains the more they are prone to belong together. Should domain then be

labelled as malicious, any strong connections to are worth investigating, if not directly label malicious as well. It is worth noting that malicious labelling will start from what are known as 'seed' domains that are known malicious domains, in theory you would feed the domain graph as many malicious domains in order for it to know all possible malicious associations, in practice there are significant overlaps between these malicious domains.

"Following Passive DNS Traces to Detect Stealthy Malicious Domains Via Graph Inference" [12] had a novel approach to passive DNS data collection where they utilized the data collected by Farsight sensors across the Internet. All of its data anonymized to preserve privacy. However, these sensors gather hundreds of millions of records daily and so the authors sought in data enrichment to filter the data, focusing on 48 IP attributes divided into three categories: domain-based, IP block-based, and query-based. This helped with domain associations. The culmination of this data treatment is the G-IP graph, with the following attributes. Within domains on the same IP links together due to probably belonging to the same organization. Any domain on a public IP is not linked right away, since anyone can use public IPs. The graph is aware of typical behaviors from malicious actors when it comes to hosting and how they will act to avoid getting caught, following these patterns create edges in the graph between malicious nodes. As a final addendum, the associations in the G-IP become stronger if nodes share dedicated or public IPs, and also if the way they change host providers carries a similarity.

In "GMAD: Graph-based Malware Activity Detection by DNS traffic analysis" [9] its authors captured DNS queries in front of DNS servers in large ISP networks. They gathered these at different times of the day in two hour intervals. To enrich the data they made a Domain Name Travel Graph (DNTG). Nodes represent domain names with a sequential correlation between nodes as directed edges, to show the way DNS query travelled and highlights potential patterns. To further enrich the data, they created graph clusters for related domain names to better detect malware domains working together. It used the attributes of client sharing ratio and query frequency to do the clustering. All of this resulted in the DNTG capable of detecting advanced malware activity like command and control servers, malware dropping, blacklist checking and more.

The paper "GNN-Based Malicious Network Entities Identification in Large-Scale Network Data" [4] does not mention any specific use of DNS data except what is there implicitly in the network traffic data that the

authors use. The enrichment involved a heterogeneous graph built on various network entities. These are given attributes, some of which are their relationships to known Indicators of Compromise (IoCs). A strong relationship is naturally an indicator of malicious activity. On top of this they use VirusTotal's database to check these domains further. This graph ends up representing the entities: Domains, users and IP addresses. The intent is capturing the most important information about malicious entities. To reduce complexity the authors extracted sub-graphs from each entity and who have some level of connection to a malicious seed.

"Mining Agile DNS Traffic Using Graph Analysis for Cybercrime Detection" [1] used a parser on network traffic and dump files. This filtered specifically after all 'DNS NOERROR' request responses that were answered with at least one IPv4 address record. The authors also added a duplicate filter to avoid recording the same event more than once. These results were then given to a series of modules that culminate in the analysis of 'change events'. These events are turned into bipartite graphs, with nodes being FQDNs and IP addresses, and any edges between them showing a relation of some sort. From here a process of elimination starts where graphs that are not evaluated as agile are removed, and then the remaining graphs are evaluated again. Agile in this case means rapid changes in DNS mapping, complex network of connections or other significant DNS changes. In other words the high agile graphs reveal malicious activity.

"NestedGNN: Detecting Malicious Network Activity with Nested Graph Neural Networks" [5] used publicly available cybersecurity datasets. As it did not contain a lot of compromised hosts comparatively, they used a sampling method where they only used 10% of both healthy and compromised hosts. As to enrichment they sought to represent the authentication relationship between hosts as a graph with hosts as nodes and authentications as edges. This created a attributed authentication graph. This consisted of an outer graph which showed the authentication relationships between hosts. Inside were inner nodes with 6607-dimensional one-hot attributes, compared to the outer nodes with no attributes. This nested structure provided a representation ripe for analysis of the network data.

In "Unveiling the potential of Graph Neural Networks for robust Intrusion Detection" [13] its authors took a partially alternative approach to data collection. They used the CIC-IDS2017 dataset which contains both benign and categorized malicious traffic. On top of this dataset's network flow they built their host-connection graph. Its nodes being hosts that are involved in the network traffic, as well as each flow of these hosts' traffic being a

node. This enabled them to focus on their relationship. This resulted in two undirected edges, one from source host to the flow and the other from flow node to destination host. Their model uses a non-standard message-passing algorithm to adhere to the needs of a network intrusion detection system. A phase of message-passing then processes the information in the graph, with the final product being GNN that is notably high in its accuracy with a weighted F1 of 0.99.

Chapter 4

Conclusion

The original intent of this paper was to collect DNS data and from it build a GNN graph. Unfortunately there were issues that came to a head with the data collection. While several attempts were made, each dataset contained fundamental issues that made it inappropriate to move forward with. Conditions unfortunately worsened further due to external personal factors. As such this paper did not so much fall short of its intentions, but instead off a cliff. All the same, the author of this paper have tried to include all decent parts of a literature review, however all too brief. There has been a genuinely very curious attempt at wrestling with a complex topic mostly new to its author. None of this excuses the quality of the final product, instead it serves as an attempt to at least explain the situation.

Bibliography

[1] Andreas Berger et al. "Mining agile DNS traffic using graph analysis for cybercrime detection". In: *Computer Networks* 100 (2016), pp. 28–44.

[2] Matteo Dell'Amico et al. "Lean on me: Mining internet service dependencies from large-scale dns data". In: *Proceedings of the 33rd Annual Computer Security Applications Conference*. 2017, pp. 449–460.

[3] Luca Deri et al. "Graph theoretical models of DNS traffic". In: *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. 2013, pp. 1162–1167.

[4] Stepan Dvorak, Pavel Prochazka, and Lukas Bajer. "GNN-Based Malicious Network Entities Identification In Large-Scale Network Data". In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2022, pp. 1–4.

[5] Yuede Ji and H Howie Huang. "NestedGNN: Detecting Malicious Network Activity with Nested Graph Neural Networks". In: *ICC 2022-IEEE International Conference on Communications*. IEEE. 2022, pp. 2694–2699.

[6] Jinyuan Jia et al. "Detection of malicious dns and web servers using graph-based approaches". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 2625–2629.

[7] Nan Jiang et al. "Identifying suspicious activities through dns failure graph analysis". In: *The 18th IEEE International Conference on Network Protocols*. IEEE. 2010, pp. 144–153.

[8] Issa Khalil, Ting Yu, and Bei Guan. "Discovering malicious domains through passive DNS data graph analysis". In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. 2016, pp. 663–674.

[9] Jehyun Lee and Heejo Lee. "GMAD: Graph-based Malware Activity Detection by DNS traffic analysis". In: *Computer Communications* 49 (2014), pp. 33–47.

[10] *Library Guides: Literature Review: Systematic literature reviews*. URL: https://libguides.csu.edu.au/review/Systematic.

[11] Pratyusa Manadhata et al. "Detecting malicious domains via graph inference". In: *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*. 2014, pp. 59–60.

[12] Mohamed Nabeel et al. "Following passive dns traces to detect stealthy malicious domains via graph inference". In: *ACM Transactions on Privacy and Security (TOPS)* 23.4 (2020), pp. 1–36.

[13] David Pujol-Perich et al. "Unveiling the potential of graph neural networks for robust intrusion detection". In: *ACM SIGMETRICS Performance Evaluation Review* 49.4 (2022), pp. 111–117.

[14] Krzysztof Rusek et al. "Unveiling the potential of graph neural networks for network modeling and optimization in SDN". In: *Proceedings of the 2019 ACM Symposium on SDN Research*. 2019, pp. 140–151.

[15] Franco Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[16] Xiaoqing Sun et al. "Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks". In: *Computers & Security* 99 (2020), p. 102057.

[17] Shuai Zhang et al. "Attributed heterogeneous graph neural network for malicious domain detection". In: *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE. 2021, pp. 397–403.

[18] Yury Zhauniarovich et al. "A survey on malicious domains detection through DNS data analysis". In: *ACM Computing Surveys (CSUR)* 51.4 (2018), pp. 1–36.

[19] Jiawei Zhou et al. "Automating botnet detection with graph neural networks". In: *arXiv preprint arXiv:2003.06344* (2020).

Appendix A

Appendix A name

Here is the first appendix