# GEMADA: Graph Embeddings for Augmenting Domain Attribution

Ron Graf
*Department of Cyber Security Engineering*
*George Mason University*
Fairfax, VA
rgraf2@gmu.edu

Joseph Mercado
*Department of Cyber Security Engineering*
*George Mason University*
Fairfax, VA
jmercad7@gmu.edu

***Abstract*—This research applies graph embedding techniques to passive DNS data in order to calculate a similarity score between two pairs of domains. This similarity score captures historical DNS record patterns between two domains in a similar way to how a human security analyst analyzes the passive DNS data of two domains to see if they are both controlled by the same advanced persistent threat group. We intend for this system to provide an augmentation to analyst attribution workflows that can be implemented at scale. The results of our research indicate that graph embeddings retain important information from passive DNS graphs such that the relationship of two domains being controlled by the same threat actor is preserved.**

***Index Terms*—graph embeddings, advanced persistent threats, passive DNS, threat attribution**

## I. Introduction

Attributing malicious infrastructure to a specific threat actor is one of the most difficult, but simultaneously highest value, tasks for security intelligence analysts. The analytic process behind these assessments is complex and currently relies heavily on scarce, expert-level security research labor. Due to the complex nature of attribution and its often equal in magnitude importance geopolitical and otherwise, it is the opinion of the authors that this process should always include a human-in-the-loop and that any fully automated attribution determination should be treated with great caution. However, we also believe there to be significant opportunity for advancement in tools available to analysts to machine-augment their analytic processes, making their investigations both more efficient and analytically robust. This work sets out to establish a graph embedding-based method to aid security analysts and researchers in the analytic process of attributing malicious infrastructure to advanced persistent threat (APT) actors.

Domains are one of several indicators of compromise (IoCs) that surface as the artifacts of a malicious actor's activities. For example, security researchers can come across domain IoCs as the result of an incident response investigation, malware reverse engineering, the detection of phishing campaign, etc. The design of the Internet's Domain Naming System (DNS) has resulted in the fact that any domain existing on the internet is owned and controlled by an entity. The identity of this entity is of significant intelligence value to security analysts. Over the last decade, structural trends in internet infrastructure, including but not limited to, the rise of cloud computing technologies and the anonymization of domain registration records have further complicated the task of attributing a domain to an entity. These trends can, in the case of actors with the highest degrees of operational security, result in scenarios where domain attribution is impossible without access to back-end data only available to the hosting services or domain registration companies themselves. However, for actors that have intentionally or unintentionally employed a lower degree of operational security in their domain hosting activities, patterns can be identified and used by security researchers as evidence of a link between two or more domains. Establishing links between multiple domains improves an analysts odds at discovering evidence that can be used to attribute one or more of those linked domains to an actor, which in turn, provides a degree of evidence that the entire cluster of linked domains is tied to the actor.

Security researchers responsible for executing on the aforementioned tasks in order to determine which APT group operates a particular domain as a part of the infrastructure the group controls can be presented with an overwhelming number of data points to analyze and weigh as evidence for or against attributing a domain to a threat actor. Passive DNS is an important source of data points interrogated by security analysts during this phase of an investigation. This source of data provides analysts with the resolution history of the domain in question. Researcher are then able to compare the resolution history of the domain in question with previously attributed domains to make an assessment as to whether there are similar patterns in their resolution histories. Example patterns, with varying degrees of strength to link domains to one another include but are not limited to temporarily overlapping resolution to the same IP address, non-temporarily overlapping resolutions to the same IP, resolution to IPs within the same Autonomous System Name (ASN). This assessment is then used as one factor in the overall decision by the analyst to attribute a particular domain.

The system we propose is an attempt at capturing the logic behind this analyst process using a graph embedding approach, allowing for increased speed and scale of this phase of an analyst's investigative workflow. This research provides the following contributions:

- A first of its kind publicly available data set of APT-attributed domains and their passive DNS data for benchmark use. [1] [2]
- Evidence that graph embeddings fit using passive DNS data can embed domains as numeric vectors that retain information useful for measuring similarity to other domains.
- A system to recommend domains controlled by the same APT group, which can be fit using any set of APT-attributed domains and corresponding passive DNS data set.
- A baseline set of performance metrics to measure and improve upon.

Additionally, the python code used to generate our results is publicly available in a reproducible fashion in the form of a git repository containing a Jupyter notebook. [3]

## II. Related Works

Systematically analyzing passive DNS data to aid in attribution of domain names to APT groups has been publicly discussed as early as 2014. [1] However, the large majority of published research regarding applying statistical techniques to passive DNS data is done so through the lens of malicious domain prediction. This task is related to but distinctly different from the task of attributing a domain to an APT group, which we set out to provide a solution to via this work. Applying machine learning solutions against the task of attributing threat activity to a specific APT group has been primarily limited to malware attribution using features derived from malicious binaries [2] [3] and attributing intrusion activity through applying natural language processing (NLP) to threat intelligence reporting [4]. Nonetheless, much of the prior body of research in malicious domain detection is relevant to this work and informs the direction of our research.

The goal of malicious domain detection is to predict whether a domain is involved with any malicious activities. As a result, it is generally formulated as a binary classification problem. Traditional methods previously used mainly adopt handcrafted features such as structural features, linguistic features, and temporal features to train supervised classifiers. The major drawback of these prior approaches is the time-consuming and labor-intensive requirement of using large amounts of labeled data. Likewise, the existing methods often focus on the content information of domains and treat each domain independently while ignoring the complex relations between domains hidden in the DNS topology.

However, recently, researchers have begun to utilize the structural information contained with DNS topology for reasoning, so that they can detect malicious domains in DNS-related graphs. These result in graph-based models that can effectively detect the malicious domains that apply evasion methods like Domain-flux or Fast-flux. [5]. Further research has explored the idea that belief-propagation (BP) or graph neural network (GNN) methods can be used to infer a domain's maliciousness label based on maliciousness of its neighbors. [6]

To address some of the limitations of previous existing methods, researchers have proposed multiple approaches. For example, an attributed heterogeneous graph neural network model named GNN-based Anti-Malicious Domain (GAMD), which uses an attributed heterogeneous graph to model DNS scenarios in fine granularity and then design a corresponding heterogeneous graph neural network method to detect malicious domains. [7]

Another method proposed is to detect malicious domains by jointly handling domain features and domain associations in a heterogeneous information network (HIN), known as SHetGCN. SHetGCN utilizes metapath guided short random walks in a HIN to find neighbors of a domain node, then it leverages attention-based aggregations to perform graph convolutions and gets the final embedding vector of a domain for detection after several iterations. [8] This is method is similar to HinDom, which constructs a HIN of clients, domains and IP addresses to model the DNS scene and generates a combined meta-path to analyze the associations among domains. With a meta-path based transductive classification method, HinDom performs well enough to reduce the cost of acquiring labeled samples by 10%. [9]

To the best of our knowledge our work is the first publicly available research that applies a graph embedding approach to passive DNS data in order to augment the task of attributing domains to APT actors.

## III. Methodology

To build a system that takes an input domain and measures similarity to other domains that have been attributed to a threat actor, we needed to compile a large data set of domains that have been attributed with some degree of confidence to a known APT group. This task is not as straight forward as it sounds for two primary reasons 1) domain indicators become publicly attributed to a threat actor via a heterogeneous set of publicly released documents, ranging from something as informal as a researcher's tweet, to something as formal as an official indictment against individual APT operators issued by a government entity and 2) the notion and identification of an APT group varies significantly across researchers within the security community.

First, we will address the implications of the first complicating factor. In between the two ends of the formality spectrum, between tweets and indictments, exists other publicly released documents that provide attribution such as blog posts or technical white papers from security vendors. These documents are one of if not the largest source of public attribution of IoCs from institutions with credible histories of accurate attributions. However, this wealth of vendor issued data comes in a wide variety of formats in varying degrees of machine readability and ingesting all these documents into a structured

[1] https://gemada.s3.us-east-2.amazonaws.com/pulses_to_actor.csv
[2] https://gemada.s3.us-east-2.amazonaws.com/pdns_combined.csv
[3] https://github.com/rg7822/gemada

format requires a large amount of human labor. Indicator sharing platforms exist, in part, to mitigate the burden of individual triage of these publications to extract IoCs in a standardized format. One such platform is Open Threat Exchange (OTX), operated by AT&T Cybersecurity (formerly AlienVault). This platform bills itself as a crowdsourced security platform, where anyone can sign up for an account and publish details of a threat. Threat details are published in discrete units called *pulses*, which include metadata about the threat such as target, suspected threat actor, malware family, etc., as well as structured IoCs such as file hashes, IP addresses, domain names, host names, vulnerabilities exploited, etc. If a pulse comes from a threat where the researcher has assessed an attribution of the threat activity, they will usually indicate that fact either in narrative description by naming the threat actor or through use of an "adversary" field in metadata. This fact allows us to use OTX as the source for the data set of attributed domains used to build our system.

Next, we will address the implications of the second complicating factor, the ambiguity of APT identification within the security community. There does not exist a widely accepted and adhered to naming convention for APT groups. The United States' National Cybersecurity FFRDC's (federally funded research and development center), operated by the MITRE Corporation, Common Vulnerability and Exposures (CVE®) system was launched in 1999 and has since gained widespread adoption. This system applies a standard naming convention for what would otherwise be an abstract, non-standard naming convention of software vulnerabilities. For example, CVE-2021-44228, refers to a remote code execution vulnerability in the popular logging library log4j from late 2021, all intelligence or security communications referencing this specific vulnerability can use the identifier CVE-2021-44228 and have a common understanding of what software vulnerability is being referred to. This type of widespread use of a universal standard does not exist for threat actors, resulting in actors often having multiple identifying names referring to the same or similar actors. For example, the Russian threat group that operates out of Russian military intelligence GRU Unit 26165, which we know with high confidence due to US Department of Justice indictments, has been referred to as at least the following aliases according to the open-source Malware Information Sharing Platform (MISP) project: APT28, Fancy Bear, Grizzly Steppe, Group 74, IRON TWILIGHT, Pawn Storm, SIG40, SNAKEMACKEREL, STRONTIUM, Sednit, Sofacy Group, Swallowtail, TAG_0700, TG-4127, and TsarTeam. This wide variety of aliases referring to what is operationally the same threat group stems from the fact that threat research organizations each have unique vantage points into threat activity, whether they are a victim, incident responder, email platform, government, etc. This heterogeneity in threat perspective results mixes with an organization's internal need to have a commonly used naming convention within their own organization for researchers to work with one another investigating the activity and results in distinct identifying names for threat groups being individually researched by separate organizations, sometimes simultaneously. Further, when organizations performing security research publish their work publicly, they most commonly use their own specific name for the group they are releasing information on. As a result, you could be reading a blog from one research group that contains valuable IoCs related to what that research group refers to as *APT28* and then read another blog from a different research organization also containing IoCs which they claim are related to *Fancy Bear* not realizing that both blog posts contain IoCs tied to the same APT operating out of GRU's Unit 26165. This complicates our work, as our goal is to recommend domains given an input domain that are likely to be linked to the same actor. This means that to build a robust system that does not miss linked domains due to differing naming conventions we need a well-defined lookup table of actor synonyms and a common actor ID. There exist a few attempts at a common threat identification system, for example, MITRE's ATT&CK framework assigns universal ids to list of synonymous groups, e.g., G0007 for GRU's Unit 26165. Another publicly available provider of unique threat group ids for lists of synonymous groups comes from the Malware Information Sharing Project (MISP), which is what we use to mitigate threat group identity ambiguity in our system. MISP's list of universal id's provides a universally unique identifier, e.g., 5b4ee3ea-eee3-4c8e-8323-85ae32658754, for lists of groups that are synonymous with one another when referred to in publicly released research from organizations across the security community. The last is well-maintained, by respected analysts in the security community, frequently updated, and publicly available on GitHub. [4]

Now that we have established OTX as our main source of attributed domains and the fact that we use MISP's universal threat actor IDs to disambiguate threat actors, we will move on to explaining our process for collection of the passive DNS (pDNS) data that will then be depicted as a graph. Domain indicators sourced from OTX are typically stored alongside a variety of domain related information such as WHOIS, associated URLs, results of scan data, etc. Importantly for us, many domains stored as indicators by OTX are also stored alongside the domains pDNS data as well, allowing us to use OTX as our source for a pDNS as well. At this point in our work, we needed to decide on a strategy for what pulses to mine domains from and thus which domains to pull pDNS data on. Due to the nature of the problem, we are attempting to solve, we felt it was important to design our collection strategy around pulses with attributions to threat actors that had be well-covered in open sources by security researchers. The rationale for focusing on the well-establish threat groups, as opposed to emerging threat groups, is that often-emerging threat groups are named by threat researchers only later to be merged with a previously established, more well-known threat group, once more evidence emerges. For this reason, we focus on 10 specific threat groups, which we chose primarily due to the large amount of data available in OTX related to

[4]https://github.com/MISP/misp-galaxy/blob/main/clusters/threat-actor.json

these groups. We ended up with 3 Russian groups, 2 Chinese groups, 4 Iranian groups, and 1 North Korean groups. (Fig. 1) The number of domains tied to these groups via the related OTX pulses we identified ranged from 55 domains to 559 domains. To have even sample sizes for each threat group, which is needed to avoid prediction bias towards groups with more domains in the data set, we down-sample each group to 55 randomly selected domains, except in the case of the group that only has 55 domains, in which case all domains are used.

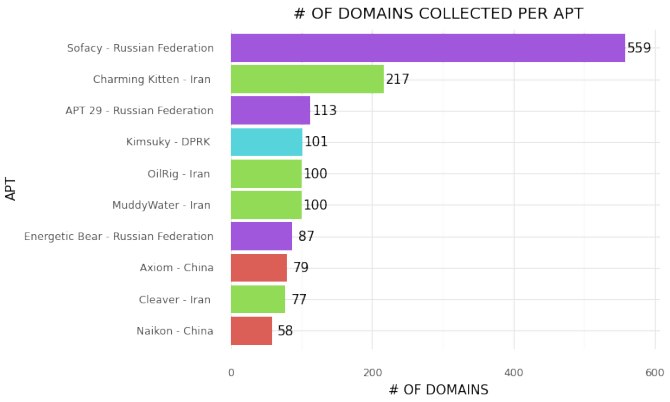


Fig. 1. Count of Domains Per Threat Actor

Once we sampled down to 55 domains for each of the 10 threat groups, we were left with a pDNS data set that contained pDNS records for 550 unique domains which amounted to 4,573 pDNS records which we use to construct a graph. This graph contains the following 5 node types: domain, nameserver, IPv4, ASN, and country. We add a node for each unique occurrence of each node type within our pDNS dataset. Undirected edges exist between the following 4 node type pairs: domain-nameserver, domain-IPv4, IPv4-ASN, and ASN-country. We chose to include this specific graph structure based on what entity and relationship types a security analyst focuses on while examining pDNS records to identify overlapping patterns between domains. Once all nodes and edges were added, our graph contained 2,668 unique nodes and 5,252 edges.

Once the graph was constructed, we used multiple graph embedding algorithms to produce vectorized representations of each node in the graph, otherwise known as node embeddings. The 3 graph embedding approaches we chose to use were DeepWalk [10], Walklets [11], and Role2Vec [12]. Fig. 2 provides an example of what the domain nodes from our graph look like when the vectors they have been embedded to using the DeepWalk technique are projected into a 2-dimensional space using the t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction technique.

After the node embeddings were generated, we extracted all domain type nodes from the graph and calculated pairwise cosine similarity between all 150,975 unique combinations of domains. We applied a min-max scale transform to the cosine similarity calculation so that the similarity calculation would be on a more intuitive 0 to 1 scale. Once we had a pairwise

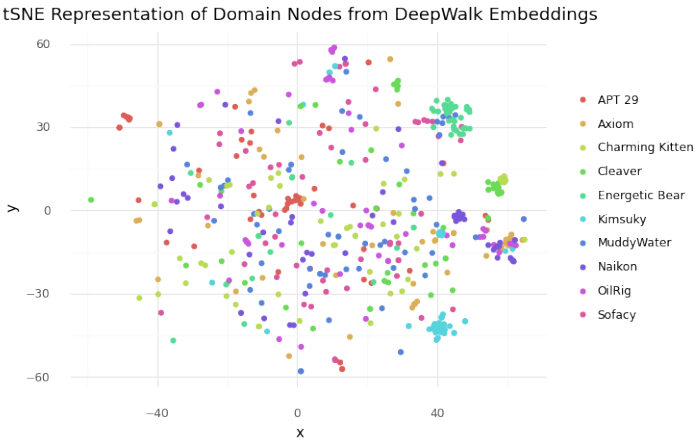


Fig. 2. Count of Domains Per Threat Actor

similarity calculation between each domain in our data set, we could evaluate the quality, from the perspective of measuring similarity of domains likely to be tied to the same APT, of our node embeddings via experimental evaluation.

To evaluate how well our similarity metric calculated on the respective node embeddings captures whether two domains are controlled by the same APT, we measure accuracy and coverage across different similarity threshold values and node embedding techniques. Accuracy in this context is defined as the number of domain pairs with a similarity score above the threshold that are from the same APT divided by the total number of domain pairs above the same threshold. Coverage in this context is defined as the number of domains pairs with a similarity score above the threshold value. Our goal is to identify which node embedding techniques provide better or worse relative accuracy and coverage across our data set of domain pairs. Our evaluation data set contains 550 total domains, 55 per APT with 10 unique APTs, which generates a total of 150,975 domain pairs. For each APT group, 1,485 of the 150,975 domain pairs exist such that both domains in the pair are labeled as belonging to that particular APT group. In a high-performance system, we would observe each of those 1,485 domain pairs per APT group having higher similarity scores than the remaining domain pairs for which the two domains in the pair come from different APT groups. A perfect system for generating similarity scores would result in all 1,485 domain pairs per APT group, a total of 14,850 domains, with a similarity score of 1.0, and the remaining 136,125 domain pairs having a similarity score of 0.0. Our experimental design measures how far away the results for each specific approach are from this theoretical system producing perfect results by calculating two area under the curve (AUC) measurements. The first AUC measurement is computed using the previously defined accuracy measurement as the y-axis and the cutoff threshold value as the x-axis, ordered from 0 to 1. The second AUC measurement is computed using the previously defined coverage measurement as the y-axis and the cutoff threshold value as the x-axis, ordered from 1 to 0.

## IV. Results

The three graph embedding techniques we used were: DeepWalk, Walklets, and Role2Vec. See Table I for parameter settings for each of these methods. Additionally, we used a fourth technique, which we refer to as our ensemble approach, that takes the similarity scores from each of the three embedding techniques and uses the median of the three. The results, displayed in Table II, show that each technique have strength and weaknesses from both an accuracy and coverage perspective at different thresholds. At the very high-end of similarity spectrum, looking at scores greater than 0.98, Role2Vec performs the most accurately at 100% covering 14 domain pairs. This trend, however, quickly changes as the threshold value decreases. Between the 0.98 and 0.92 threshold values, the ensemble approach provides the highest level of accuracy, however, with relatively weaker coverage, only covering more domain pairs than the DeepWalk method. Below the 0.92 threshold values, DeepWalk becomes the clear leader in accuracy, until about the 0.5 threshold value, where DeepWalk, Ensemble, and Walklets roughly converge in terms of accuracy. At the higher end of the threshold spectrum, roughly 0.93 and above, Walklets and Role2Vec are approximately equally strong in terms of domain pairs coverage. Below the threshold value of 0.93, Role2Vec is the dominant performer in terms of coverage. There is a clear trade-off between accuracy and coverage, with Role2Vec having the best coverage AUC at 50.2% but the worst accuracy AUC at 21.6%, likewise, DeepWalk has the best accuracy AUC at 41.8% but the lowest coverage AUC with 32.1%. The ensemble method provides the most evenly distributed performance between accuracy and coverage, with an second-best accuracy AUC of 35.9%, although only 0.1% better than the third highest performer, Walklets, and a second-best coverage AUC at 37.0%. Overall, we believe that these initial results are promising and serve as a baseline for a system that could introduce efficiencies into a threat analyst's workflow when making attribution decisions involving domain indicators.

TABLE I
Parameter settings for each of the graph embedding techniques

| Technique | Walks | Walk Length | Window Size | Dimensions |
|---|---|---|---|---|
| **DeepWalk** | 20 | 5 | 80 | 128 |
| **Walklets** | 20 | 5 | 80 | 32 |
| **Role2Vec** | 20 | 5 | 80 | 128 |

TABLE II
Performance metrics for each technique

| Technique | Accuracy AUC | Coverage AUC |
|---|---|---|
| **DeepWalk** | 41.8% | 32.1% |
| **Walklets** | 35.8% | 33.6% |
| **Role2Vec** | 21.6% | 50.2% |
| **Ensemble** | 35.9% | 37.0% |

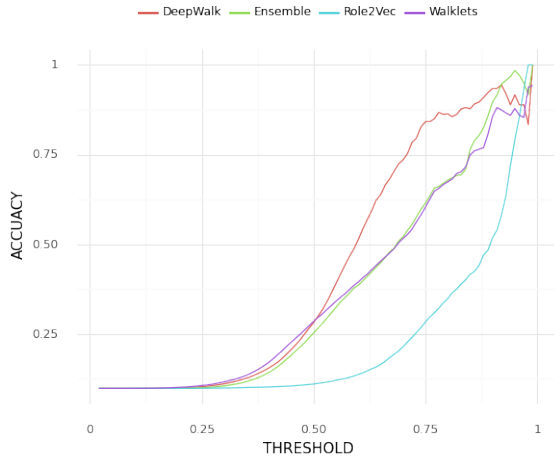


Fig. 3. Accuracy Over Similarity Score Threshold for each Approach

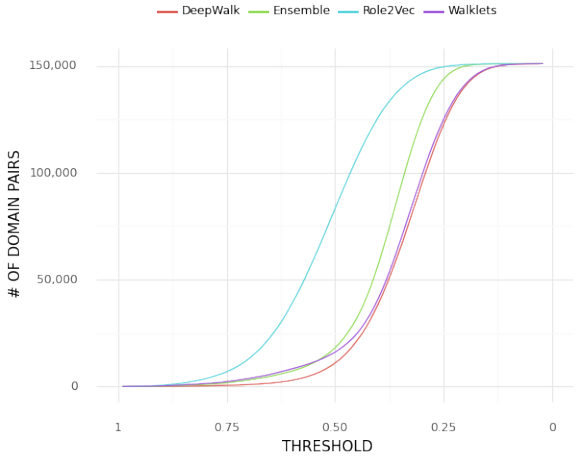


Fig. 4. Domain Pair Coverage Over Similarity Score Threshold for each Approach

## V. Future Work

There are a number of directions this research can be extended. We would like to see a larger data set constructed with both more unique threat actors and more domains and compare results to the current data set. Additionally, we would be interested in a method that can handle domains that are benign. In the systems current state, a determination that the domain is malicious must be made prior to system use in order for the metrics to be meaningful. Related and further, being able to extend the system to be able to identify emergent APT groups outside of the distinct unique set tracked in the code used to fit the graph embedding would significantly increase the benefit of this system to the threat research community. Lastly, we believe that exploring the temporal components within the passive DNS data and encoding that information in a format that a graph embedding can learn from would yield a significant performance benefit across all approaches.

## VI. Conclusion

This work represents a novel application of graph embeddings on passive DNS data to provide a solution to the problem of attributing domain infrastructure to an APT actor. We have made the data set of attributed domains and their corresponding passive DNS data publicly available as an option for a benchmark data set to evaluate further attempts at solving this problem. The results of this paper offers evidence that passive DNS data when expressed as a graph can provide the underlying representation from which graph embedding techniques can extract meaningful vector-space representations of domains. The node embeddings produced during our research provided what we believe to be a useful tool to aid security researchers in their investigative process, while investigating the passive DNS records of a particular domain as evidence while making an attribution determination. We provided a spectrum of options that allow for balancing or favoring accuracy and/or coverage and importantly outputs in the form of a similarity metric that allow for end-user tuning based on use-case specific acceptable levels of noise. Importantly, this system's architecture can be readily applied to any proprietary data sets to produce domain embeddings that are specific to individual threat research organizations. We believe this research represents a solid baseline for which to compare further techniques against, as well as, provides a tool that can aid security researchers in making attribution decisions.

## VII. References

## References


[1] Frankie Li. Apt attribution and dns profiling. Blackhat USA, 2014.

[2] Ishai Rosenberg, Guillaume Sicard, and Eli David. Deepapt: Nation-state apt attribution using end-to-end deep neural networks. *Lecture Notes in Computer Science*, page 91–99, 2017.

[3] Ishai Rosenberg, Guillaume Sicard, and Eli David. End-to-end deep neural networks and transfer learning for automatic analysis of nation-state malware. *Entropy*, 20(5):390, May 2018.

[4] Lior Perry, Bracha Shapira, and Rami Puzis. No-doubt: Attack attribution based on threat intelligence reports. In *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 80–85, 2019.

[5] Kai Lei, Qiuai Fu, Jiake Ni, Feiyang Wang, Min Yang, and Kuai Xu. Detecting malicious domains with behavioral modeling and graph embedding. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 601–611, 2019.

[6] Mohamed Nabeel, Issa Khalil, Bei Guan, and Ting Yu. Following passive dns traces to detect stealthy malicious domains via graph inference. *ACM Transactions on Privacy and Security*, 23:1–36, 07 2020.

[7] Shuai Zhang, Zhou Zhou, Da Li, Youbing Zhong, Qingyun Liu, Wei Yang, and Shu Li. Attributed heterogeneous graph neural network for malicious domain detection. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 397–403, 2021.

[8] Xiaoqing Sun, Zhiliang Wang, Jiahai Yang, and Xinran Liu. Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks. *Computers Security*, 99:102057, 2020.

[9] Xiaoqing Sun, Mingkai Tong, and Jiahai Yang. Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification, 2019.

[10] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. *CoRR*, abs/1403.6652, 2014.

[11] Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. Walklets: Multiscale graph embeddings for interpretable network classification. *CoRR*, abs/1605.02115, 2016.

[12] Nesreen K. Ahmed, Ryan Rossi, John Boaz Lee, Theodore L. Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. Learning role-based graph embeddings, 2018.