arXiv:2203.07969v1 [cs.LG] 15 Mar 2022

PDNS-Net: A Large Heterogeneous Graph Benchmark Dataset of Network Resolutions for Graph Learning

Udesh Kumarasinghe
udeshk@scorelab.org
University of Colombo
Sri Lanka

Fatih Deniz, Mohamed Nabeel
(fdeniz,mnabeel)@hbku.edu.qa
Qatar Computing Research Institute
Qatar

ABSTRACT

In order to advance the state of the art in graph learning algorithms, it is necessary to construct large real-world datasets. While there are many benchmark datasets for homogeneous graphs, only a few of them are available for heterogeneous graphs. Furthermore, the latter graphs are small in size rendering them insufficient to understand how graph learning algorithms perform in terms of classification metrics and computational resource utilization. We introduce, PDNS-Net, the largest public heterogeneous graph dataset containing 447K nodes and 897K edges for the malicious domain classification task. Compared to the popular heterogeneous datasets IMDB and DBLP, PDNS-Net is 38 and 17 times bigger respectively. We provide a detailed analysis of PDNS-Net including the data collection methodology, heterogeneous graph construction, descriptive statistics and preliminary graph classification performance. The dataset is publicly available at https://github.com/qcri/PDNS-Net. Our preliminary evaluation of both popular homogeneous and heterogeneous graph neural networks on PDNS-Net reveals that further research is required to improve the performance of these models on large heterogeneous graphs.

KEYWORDS

dataset, benchmark, heterogeneous graph, GNNs, DNS, malicious domains

ACM Reference Format:
Udesh Kumarasinghe and Fatih Deniz, Mohamed Nabeel. 2022. PDNS-Net: A Large Heterogeneous Graph Benchmark Dataset of Network Resolutions for Graph Learning. In *Proceedings of Graph Learning Benchmark (GLB '22).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The availability of various graph datasets across multiple domains have fueled the accelerated development of graph learning techniques that take into consideration both the graph structure and node/edge attributes to either learn low dimensional representations of graph nodes or classify nodes. However, the research community has identified several limitations with the current benchmark datasets making it difficult to characterize and differentiate modern graph learning techniques: (1) The size of the graphs in terms of the number of nodes and edges is quite limited and (2) Most of these graphs are homogeneous containing only one type of nodes and one type of edges.

To address these issues, we introduce a new graph dataset called PDNS-Net, a large-scale heterogeneous graph containing the IP resolutions of Internet domains observed in October 2021 through passive DNS data collection [23]. The graph is constructed from a seed set of malicious domains collected from VirusTotal [22] and the hosting infrastructure behind these seed domains are extracted from a popular passive DNS repository that passively records most of the domain resolutions occur all around the world [5]. Due to various practical reasons, attackers utilize similar hosting infrastructures to host their domains. The network security research community has utilized the structural properties along with the domain/IP attributes to distinguish malicious domains from benign ones [15, 20]. However, the datasets utilized and the experiments carried out in such research works suffer from several limitations: (1) the collection and labeling methodology are not clear, (2) primarily homogeneous GNN models are utilized, (3) the datasets are small and (4) the datasets are not publicly available.

In order to assist researchers and practitioners design and explore various graph learning methods quickly as well as execute legacy GNN models that do not scale with the dataset size, we have created a smaller version of PDNS-Net called mPDNS-Net (miniPDNS-Net) by sampling PDNS-Net [13].

Our preliminary results of executing various graph neural network models (both homogeneous and heterogeneous) reveal that (1) larger graphs perform better in the classification task compared to the smaller ones in the heterogeneous setting but surprisingly not in the homogeneous setting, and (2) The classification metrics among homogeneous and heterogeneous graph models are marginal, especially for smaller graphs. Thus, PDNS-Net provides the opportunity for the research community to advance the state of the art on graph learning on large heterogeneous graphs by addressing these observations. Furthermore, PDNS-Net provides new research opportunities to advance the graph learning on heterogeneous graphs in several directions, including imbalanced classification, adversarial robustness and explainability.

2 BACKGROUND

2.1 Homogeneous and Heterogeneous Graphs

Definition 2.1. A **Graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes data structure, where $\mathcal{V}$ is the vertexes (or node set), and $\mathcal{E} \in (\mathcal{V} \times \mathcal{V})$ denotes the edge set. Let $A \in [0,1]^{|\mathcal{V}| \times |\mathcal{V}|}$ represent a binary adjacency matrix where $A_{ij} = 1 \; if \; (i,j) \in \mathcal{E}$.

GLB '22, April 25–29, 2022, Lyon, France

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

A Heterogeneous Graph is a graph structure that represents different entities and their different relationships. A heterogeneous graph can be formally defined as follows.

Definition 2.2. A **Heterogeneous Graph** [18] is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ and $\mathcal{E} \in (\mathcal{V} \times \mathcal{V})$ denote the set of nodes and edges, respectively. Furthermore, $\mathcal{G}$ is associated with a node type mapping $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge type mapping $\psi : \mathcal{E} \rightarrow \mathcal{R}$ where the number of node types $|\mathcal{A}| > 1$ or the number of edge types $|\mathcal{R}| > 1$. Edge set $\mathcal{E}$ is also represented as an adjacency matrix $A \in [0,1]^{|\mathcal{V}|\times|\mathcal{V}|\times|\mathcal{R}|}$ such that $A_r \in [0,1]^{|\mathcal{V}|\times|\mathcal{V}|}$ is a sub matrix representing the edge of type $r \in \mathcal{R}$.

In the context of DNS, entities such as domains, or IPs and the relationships among them are represented as a heterogeneous graph. These relationships between entities can be different connections such as a domain resolving to an IP, or a domain is a subdomain of another. This structure is heterogeneous since it contains either different types of nodes or different types of edges. An advantage of heterogeneity is the ability to represent rich and diverse relationships among the entities. Figure 1 shows the high level schema of the PDNS-Net DNS graph.

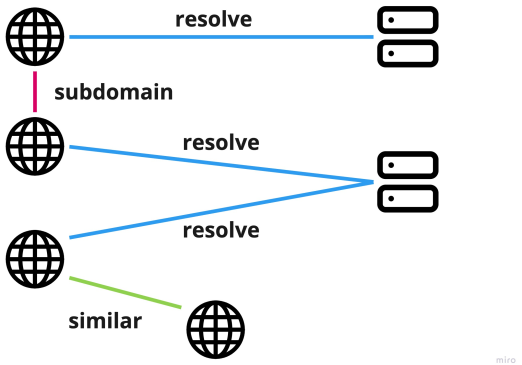


Figure 1: High level schema of the Heterogeneous DNS graph

2.2 DNS and Malicious Domains

DNS is a hierarchical naming system that helps map domain names to IP addresses in the Internet. It is one of the core protocol suites of the Internet. It provides a distributed database that maps domain name to record sets, such as IP addresses. We make a distinction between private and public domains. A domain is public if its subdomains or path suffixes are not created and not under the control of the domain owner, whereas a domain is private if its subdomains are created and managed by the domain owner.

Internet domains are the launchpad for many cyber attacks we observe nowadays. We term those domains that are utilized in such attacks as malicious domains [19].

2.3 Passive DNS

Passive DNS (PDNS) [23] captures traffic by sensors cooperatively deployed in various DNS hierarchy locations. For example, Farsight PDNS data [5] utilizes sensors deployed behind DNS resolvers and provides aggregate information about domain resolutions. We use Farsight PDNS DB to extract domain resolutions retrospectively.

3 DATA COLLECTION AND CHARACTERISTICS

3.1 Data Collection

We obtain ground truth data and passive DNS resolutions to build the DNS graph from the data collected from different sources during the period of 11/10/2020 to 18/10/2020.

Malicious Ground Truth. As shown in Figure 2, we collect malicious domain ground truth from VirusTotal Feed [22]. We execute this pipeline to identify malicious domains created by attackers as they exhibit homophily relationships.

Benign Ground Truth. Our benign ground truth is collected from the Alexa top 1m list [1]. Although some malicious domains make to the top domain list if they attract high popularity, such domains do not persist on the top domain list over a period of time. Thus, following prior research outcomes [15], we select those domains in Alexa top 1m that consistently appear for 90 days as our benign ground truth.

3.2 Passive DNS Expansion

We expand the malicious seed identified from the VirusTotal feed to identify domain-IP resolutions using the Farsight PDNS service [5]. The following steps are performed to collect the passive DNS resolutions:

1. Collect the IPs hosting the malicious domains.
2. Collect the domains hosted on IPs collected in step 1.
3. Collect the IPs hosting the domains collected in step 2.

3.3 Graph Construction

The collected domain-IP resolutions are then used to build a heterogeneous knowledge graph consisting of nodes of type domain, IP, subdomain and 3 types of relationships among them.

domain-resolve-ip. The relationship represents the domains hosted on a given IP.

domain-similar-domain. The character level similarity between two domains is used to generate relationships between domain nodes. In order to find domain similarity, n-gram is used to process domain names and embed each domain in a high level representation using TF-IDF vectorization. Then, the cosine similarity is used to identify domains with character level similarity. Tri-grams are used for the n-gram process and 0.8 similarity threshold is used for cosine similarity.

domain-subdomain-domain. The relationship represents the domains sharing the same apex domain.

3.4 Node Feature Extraction and Metapaths

The lexical features of domain nodes in Table 1 capture the lexical formation of domain names. Attackers create many domains to remain agile but such domains leave certain revealing patterns allowing one to associate them by their lexical features. These features are extracted by pre-processing the domain names before

Table 1: Lexical Features

Feature Name	Description	Source
#Subdomains	The number of levels in the subdomain part of the FQDN	[14]
Minus	The number of dashes appear in the FQDN	[14]
Brand	Does it impersonate a popular Alexa top 1000 brand?	Derived from [11]
Similar	Does the domain contain words within Levenshtein distance 2 of a popular Alexa top brand?	Derived from [11]
Fake_TLD	Does the domain name include a fake gTLD (com, edu, net, org, gov)?	Derived from [16]
Pop_Keywords	Does the domain name include popular keywords?	Derived from [11]
Entropy	The entropy of the FQDN	[2, 3]

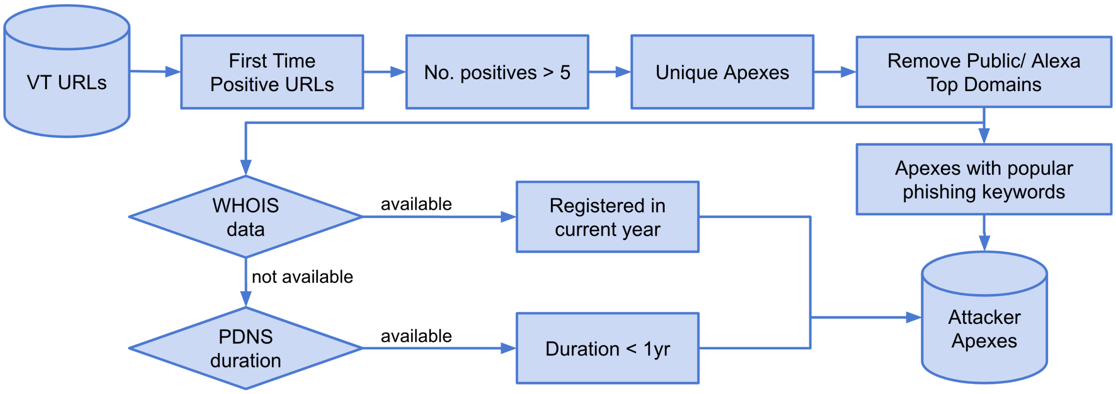


Figure 2: Groud truth malicious domain filtering pipeline

building the graph. Subnet and ASN of each IP are used as categorical features of the IP nodes. An added advantage of these features is that they are quite efficient to collect.

We define the following metapaths [4] for the PDNS-Net graph:

- domain - similar - domain
- domain - subdomain - domain
- domain - resolve - IP - resolve - domain

3.5 Graph Pruning

We perform the following pruning based on empirically identified thresholds in order to reduce the noise in the graph.

IP Pruning. IP nodes having higher degrees are most likely from firewalls or some public IPs. Thus, domains hosted on such IPs are less likely to be related to one another. We prune IPs hosting more than 1500 domains.

Public domain removal. Public domains (i.e. wix.com or 000webhostapp.com) host many unrelated subdomains. Two subdomains sharing the same public domain are less likely to be related. Thus, we prune all public domains from the resolution graph.

Isolated node removal. We prune those connected components having only one domain node as they do not contribute to any graph learning algorithm.

3.6 mPDNS-Net Dataset

The majority of current graph adversarial attacks [7] and many of the advanced GNN algorithms [9, 21] are inefficient on large graphs such as ours. Therefore, a representative subgraph of PDNS-Net is generated to be compatible with current graph adversarial attacks and GNN models. Multiple methods have been researched on sampling representative subgraphs from large-scale graphs. According to [13], exploratory sampling strategies based on random-walks outperforms both uniform node sampling and edge selection based strategies. Thus, an exploratory random-walk graph sampling method, Metropolis-Hastings graph sampling algorithm [10], is used to generate a representative subgraph of the PDNS-Net graph. We refer to the sample dataset as mPDNS-Net.

3.7 Descriptive Statistics

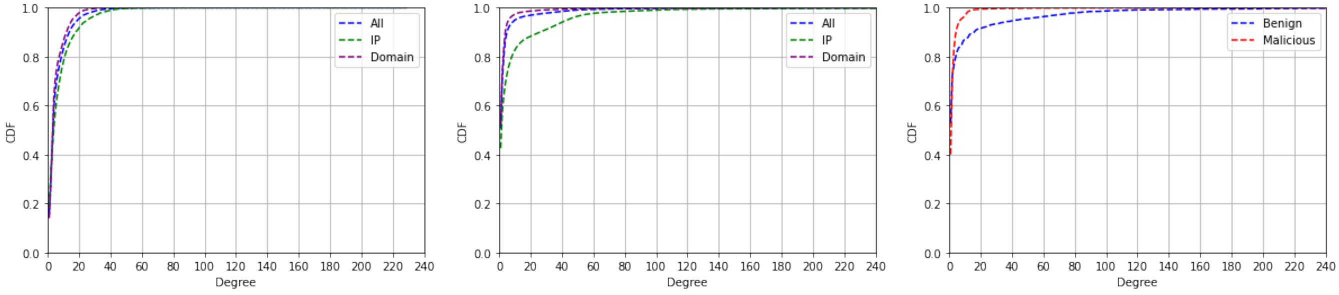
Table 2 shows the first order statistics of the two datasets. Figure 3 shows the node degree distributions for the two datasets. We make several observations from these statistics: (1) Both DNS and mDNS datasets have similar node degree distributions confirming that our sampling approach is representative, (2) The degrees of IP nodes are higher than that of domains, and (3) Benign domains are hosted on more IPs than malicious domains.

Table 2: Graph statistics of the datasets

Dataset	#Domains	#IPs	#Edges	#Malicious	#Benign
mDNS	7,495	4,505	37,285	2,827	4,668
DNS	373,475	73,593	897,588	20,354	4,963

4 NODE CLASSIFICATION RESULTS

Some GNNs are designed for homogeneous graphs that consist of only one type of node and one type of edge, such as GCN [12], GraphSAGE [8], and GAT [21]. Some others are designed for heterogeneous graphs and can deal with different types of nodes and



(a) mDNS **(b) DNS** **(c) Benign vs. Malicious**

Figure 3: Degree distribution of the two datasets

Table 3: Malicious domain detection performance comparison using different GNN models on the two datasets

Methods	mDNS						DNS					
	Acc.	AUC	F1	Prec.	Recall	FPR	Acc.	AUC	F1	Prec.	Recall	FPR
GCN	0.81	0.81	0.81	0.81	0.80	0.18	0.77	0.77	0.77	0.79	0.72	0.18
GraphSage	0.84	0.84	0.84	0.80	0.90	0.22	0.76	0.76	0.76	0.80	0.71	0.18
GAT	0.83	0.83	0.83	0.82	0.85	0.20	0.76	0.76	0.76	0.76	0.74	0.23
RGCN	0.85	0.85	0.85	0.84	0.86	0.16	0.78	0.78	0.78	0.81	0.74	0.17
HGT	0.87	0.87	0.87	0.88	0.86	0.12	0.90	0.90	0.90	0.92	0.88	0.08
HeteroSAGE	0.89	0.89	0.89	0.91	0.86	0.09	0.93	0.93	0.93	0.94	0.92	0.06
HeteroGAT	0.86	0.86	0.86	0.84	0.89	0.17	0.94	0.90	0.94	0.96	0.97	0.17

relationships, such as HGT [9], while others can handle only different types of relationships, but not node types, like RGCN [17]. It is also possible to use generic wrappers, on GNNs that are designed for homogeneous graphs, to deliver messages from source nodes to target nodes based on the bipartite GNN layer for each edge type, and compute graph convolution on heterogeneous graphs. We use such a wrapper in PyG [6] environment and generate heterogeneous counterparts of GraphSAGE [8] and GAT [21] and named them as HeteroSAGE and HeteroGAT, respectively.

In this section, we introduce experimental results on the two datasets described in this paper, namely mPDNS-Net and PDNS-Net, on above-mentioned different variants of GNNs designed for homogeneous and heterogeneous graphs. We compare their performance according to several metrics including accuracy, area under the curve, F1-score, precision, and false positive rate as shown in Table 3. For heterogeneous to homogeneous conversion, all features with the same feature dimensionality across different types are merged into a single representation and the missing dimensions are filled with zero values. For training the GNN models, we use an Adam optimizer with the learning rate of 5 x 10^{-3}, weight decay of 1 x 10^{-3}, and the number of epochs of 200. The designed models consist of two GNN layers with the hidden dimension size of 64 and a linear layer for the classification. Models are implemented using PyTorch Geometric (PyG) [6] version 2.0.3 built on top of PyTorch version 1.10.0.

We make several surprising observations that require further attention from the research community. For mPDNS-Net, while heterogeneous models perform better than homogeneous models, the improvement in performance is marginal. An important research direction is to construct heterogeneous models that perform much better on small heterogeneous graphs compare to the homogeneous counterparts. In comparison to the heterogeneous models on the smaller dataset mPDNS-Net, as expected, similar models perform better on the larger dataset PDNS-Net as it is likely to capture richer associations and interactions among domain nodes. However, it is surprising to observe that such performance gain is not achieved for the homogeneous models. We assess that further study is required to diagnose the poor performance of homogeneous models on the large dataset PDNS-Net.

5 CONCLUSION

Graph learning plays a critical role in advancing the machine learning tasks on interconnected graph data. Currently, while there are many publicly available homogeneous graph datasets, only a handful of heterogeneous graph datasets are available albeit being small in size. We address this by constructing PDNS-Net, the largest publicly available heterogeneous graph dataset for the malicious domain classification. Our preliminary results on both homogeneous and heterogeneous GNN models on PDNS-Net show that the research community needs to do more to improve the performance of GNNs on such heterogeneous large datasets.

REFERENCES

[1] 2022. Alexa: The Web Information Company. https://www.alexa.com/topsites. Accessed January 2022.

[2] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. 2012. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *Presented as part of the 21st USENIX Security*. USENIX, Bellevue, WA, 491–506. https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/antonakakis

[3] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez. 2017. Classifying phishing URLs using recurrent neural networks. In *eCrime*. 1–8.

[4] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.

[5] Farsight Security, Inc. 2022. DNS Database. https://www.dnsdb.info/.

[6] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[7] S. Geisler, D. Zugner, A. Bojchevski, and S. Gunnemann. 2021. Attacking Graph Neural Networks at Scale. In *Deep Learning for Graphs at AAAI*.

[8] W. Hamilton, Z. Ying, and J. Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

[9] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[10] C. Hubler, H. Kriegel, K. Borgwardt, and Z. Ghahramani. 2008. Metropolis Algorithms for Representative Subgraph Sampling. In *ICDM*. 283–292.

[11] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gomez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *CCS*. ACM, New York, NY, USA, 569–586.

[12] T. Kipf and M. Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[13] J. Leskovec and C. Faloutsos. 2006. Sampling from Large Graphs. In *KDD*. 631‚Äì636.

[14] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *Proceedingsof theSIGKDD Conference. Paris,France*.

[15] M. Nabeel, I. M. Khalil, B. Guan, and T. Yu. 2020. Following Passive DNS Traces to Detect Stealthy Malicious Domains Via Graph Inference. *ACM Trans. Priv. Secur.* 23, 4, Article 17 (July 2020), 36 pages.

[16] R. Roberts, Y. Goldschlag, R. Walter, T. Chung, A. Mislove, and D. Levin. 2019. You Are Who You Appear to Be: A Longitudinal Study of Domain Impersonation in TLS Certificates. In *CCS*. 2489–2504.

[17] M. Schlichtkrull, T. Kipf, P. Bloem, R. van¬†den Berg, I. Titov, and M. Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*. 593–607.

[18] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. 2017. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2017), 17–37. https://doi.org/10.1109/TKDE.2016.2598561

[19] R. Silva, M. Nabeel, C. Elvitigala, I. Khalil, T. Yu, and C. Keppitiyagama. 2021. Compromised or Attacker-Owned: A Large Scale Classification and Study of Hosting Domains of Malicious URLs. In *USENIX Security*. 3721–3738.

[20] Xiaoqing Sun, Mingkai Tong, and Jiahai Yang. 2019. HinDom: A Robust Malicious Domain Detection System based on Heterogeneous Information Network with Transductive Classification. In *RAID*.

[21] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *stat* 1050 (2017), 20.

[22] VirusTotal, Subsidiary of Google. 2022. VirusTotal. https://www.virustotal.com/.

[23] Florian Weimer. 2005. Passive DNS Replication. In *FIRST*. 98.