

HGDom: Heterogeneous Graph Convolutional Networks for Malicious Domain Detection

1st Xiaoqing Sun*, 2nd Jiahai Yang*, 3rd Zhiliang Wang*, 4rd Heng Liu⁺

*Institute for Network Sciences and Cyberspace, Tsinghua University

*National Research Center for Information Science and Technology, Beijing, China

sxq16@mails.tsinghua.edu.cn, {yang, wzl}@cernet.edu.cn

⁺China Electronics Cyberspace Great Wall Co., Ltd, Beijing, 102209, China

liuheng@cecgw.cn

Abstract—As a fundamental component of the Internet, Domain Name System (DNS) is widely abused by attackers in various cybercrimes, making malicious domain detection an essential task in network defenses. However, some well-crafted attacks with tricky techniques can not only bypass blacklists but also make some machine learning-based detection systems infeasible. In this paper, we design HGDom, an accurate and robust malicious domain detection system based on a heterogeneous graph convolutional network method. First, we jointly analyze domain features as well as the complex relations among domains, clients, and IP addresses. To capture richer information, we introduce a Heterogeneous Information Network (HIN) to model the DNS scene. Then, we propose a novel representation method named MAGCN. With a meta-path-based attention mechanism, it can handle node features and the graph structure in HIN at the same time. To our best knowledge, this is the first work to apply GCN in cyber security analysis. Comprehensive experiments over DNS data from TUNET and CERNET2 are conducted to validate the effectiveness and superiority of our proposed methods. The comparison results show that HGDom outperforms state-of-the-art approaches with promising performance. Besides, the system is decided to be deployed in production to assist with network security management for CERNET2.

Index Terms—Malicious Domain Detection, Graph Convolutional Networks, Heterogeneous Information Networks, Metapath

I. INTRODUCTION

Domain Name System (DNS) is a fundamental protocol of the Internet and provides a hierarchy of unique mappers, translating easy-to-remember domain names to IP addresses. The ability to isolate the underlying network changes from the application layer makes DNS an important attack vector in various cybercrimes, such as hosting phishing webpages, sending spam messages and facilitating command and control (C&C) communications, etc. Many organizations and security vendors construct blacklists or sinkholes to block domains involved in malicious activities. However, with evasion tactics like Fast-Flux, Domain-Flux, or Double-Flux, attackers can easily make these static methods infeasible [1]. Therefore, how to effectively detect malicious domains is always an essential topic in security research.

Many approaches have been proposed for malicious domain detection [2], which, according to different data sources and

algorithms, fall into three categories: 1) DNS traffic monitoring, 2) domain name analysis and 3) graph-based mining. Researches in category one [3]–[6] extract spatial and temporal features from DNS traffic to train domain classifiers while those in category two [7]–[9] mainly focus on the different character distributions of domain names. However, these systems may easily be evaded by weasel attackers, as they rely on local features by treating each domain individually. For example, it is easy for attackers to change request intervals or name patterns [10], which commonly serve as major features of the classifier. To tackle this issue, some researchers find that attackers cannot easily falsify the natural relations generated in DNS and hold the intuition that a domain which has strong associations with known malicious domains is likely to be malicious. Thus, they turn to discover associations among domains by constructing client-domain graphs [11], [12], domain-IP graphs [13]–[15], or even a heterogeneous network model [16]. However, with traditional graphic analysis methods, these work can only focus on relations among domains but ignore the attributes of domains themselves.

In this paper, we propose a system named *HGDom* to handle malicious domains detection with the consideration of both domain local features and their global associations. HGDom is designed based on observations from three perspectives: i) character distribution of domain names, ii) resources aggregation of attackers and iii) query behaviors of clients. To fuse richer information, we model the DNS scene as a Heterogeneous Information Network (HIN) of clients, domains, IP addresses, and different types of relations among them. Then meta-paths are elaborately extracted to help uncover higher-level semantics hiding in the HIN. We plan to simultaneously handle domain features and the various associations among domains with a Graph Convolutional Network (GCN), as it has achieved great success on non-Euclidean data like graphs [17]. However, the existing GCN methods only focus on homogeneous networks, which is not suitable for most real-world systems, as there are always plenty of interacting, multi-typed components. Therefore, in this work, we propose a variation of GCN model named *MAGCN* (GCN Meta-path-based Attention mechanism) to handle HIN. During the process, an attention mechanism is applied to adaptively learn the importance of each subgraph extracted according to different

meta-paths. With DNS traffic collected from two real-world networks TUNET, the campus network of Tsinghua University, and CERNET2, the second generation of China Education and Research Network, we empirically show that HGDome outperforms other existing graph mining-based malicious domain detection systems. Specifically, we make the following contributions:

- We propose a malicious domain detection system named *HGDome*. By modeling DNS as a HIN and applying a heterogeneous GCN method, HGDome can fully exploit the DNS scene and discover malicious domains accurately.
- We design a new deep learning method named *MAGCN*. With an attention mechanism based on meta-paths, it can jointly handle node features and the structural information in HIN.
- We implement a prototype of HGDome and demonstrate the effectiveness and superiority of our proposed approaches by extensive experiments over two real-world datasets collected from TUNET and CERNET2.

The structure of this paper is organized as follows. We introduce the background and motivation of HGDome in Section II. Section III presents HGDome’s architecture and the technical details of its key components. Experimental results are shown in Section IV. We systematically review the related works about malicious domain detection, GCN, and HIN in Section V, and we conclude this paper in Section VI.

II. BACKGROUND AND MOTIVATION

To avoid being blocked by blacklists, attackers have developed techniques like Fast-Flux, Domain-Flux, Double-Flux, etc. to improve flexibility. The key point of these techniques is to hide the real location of malicious services by continuously changing mapping relations between domains and IP addresses. For instance, DGAs are commonly used to provide bulks of alternative domain names to shield the C&C channel. Besides, sophisticated attackers also try to evade detection system by modifying temporal and spatial features. For example, they obscure the character distribution of domain names, change request intervals, or TTL values. Fortunately, though with tricky evasions, attacks still leave traces in DNS-related data on three aspects: domain names, clients, and attackers. We propose HGDome based on the following three intuitions.

(1) Domain names. DGA is essentially a random string generation algorithm. Therefore, the character distributions of DGA-generated domains are different from those of the legal ones. Figure 1 illustrates some examples of legal domains selected from Alex Top 1M [18], a list of commonly queried domains on the Internet, together with some domains generated by five different DGAs. We can see that legal domains are more readable with some practical meanings while DGA domains look completely random.

(2) Clients. Victims infected by the same attackers tend to query the same (or partially overlapping) set of malicious domains. Meanwhile, the uninfected clients have few reasons

googleusercontent.com	keorqovrrnstrsfo.com	vkiteswyai.biz
amazon.in	swvdsxxewgksbk.net	ycgyrmladkm.org
dropbox.com	swvdsxxewgksbk.net	zjeuuzcq.ws
amazon.de	rqqixtowyplih.info	wdhyeh.org
(a) legal domains	(b) matsnu	(c) conficker
ovenbdjnihhdlb.net	dkbmkbdfclcdbae.com	tdkqed6rgfq9jf3i.biz
linsmnokksa.info	denncdbalbdbkbl.website	fr53cl73gzgfi154.com
xkhvdfdpvj.com	cmfocoddcdaffde.co	naj6fo32gadm1igg.net
lwiuoljgmhnd.net	mlalffbcakakbbc.com	gwes012c1sf45s7.biz
(d) crypto	(e) padcrypt	(f) chinad

Fig. 1. Illustration of legal domains (a) and five different types of DGA domains (b-f)

to query these domains, as they mostly exist for the sole purpose of holding malicious services.

(3) Attackers. Due to economic constraints, it is hard for attackers to own many network resources. Therefore, though intending to be dynamic, they have to reuse resources, creating intrinsic associations among malicious domains. For example, when analyzing DNS response traffic, we can find a few DGA generated domains are all resolved to a certain IP address.

In a word, the character distribution of domain names, query behaviors of clients, and resources aggregation of attackers can all be utilized for malicious domain detection. To jointly consider such three intuitions, HGDome leverages a HIN model to represent clients, domains, IP addresses, and the diverse relations among them. Besides, we need a novel method to simultaneously handle domain features and the graph structure in this HIN.

III. THE OVERVIEW OF HGDOME

Figure 2 illustrates the architecture of HGDome. After collecting DNS-related data (e.g., DNS traffic, DNS logs, passive DNS datasets, etc.) from the local area network, we construct a HIN to represent the DNS scene. Some nodes are pruned to eliminate noises and reduce complexity. Then our proposed algorithm MAGCN is applied to this HIN for node classification. During this process, multiple meta-paths are built to extract subgraphs, and an attention mechanism is trained to guide the GCN model. Finally, HGDome generates a detection report telling whether a domain is malicious or not. In the following sections, we describe the key steps of HGDome in details, including Data collection, HIN construction, meta-paths extraction, and the MAGCN algorithm.

A. Data Collection

As shown in Figure 3, various kinds of data are generated during the recursive process of domain query. To gain more representative information reflecting the actual situation in the network, we conduct passive data collection and mainly utilize three types of data: i) DNS traffic. With various fields like src, rcode, TTL, etc., it reflects communications among clients, the resolver and higher-level DNS servers in details; ii) passive DNS (pDNS) dataset. To gain more macro perspectives and avoid privacy disclosure, we can aggregate the raw traffic data

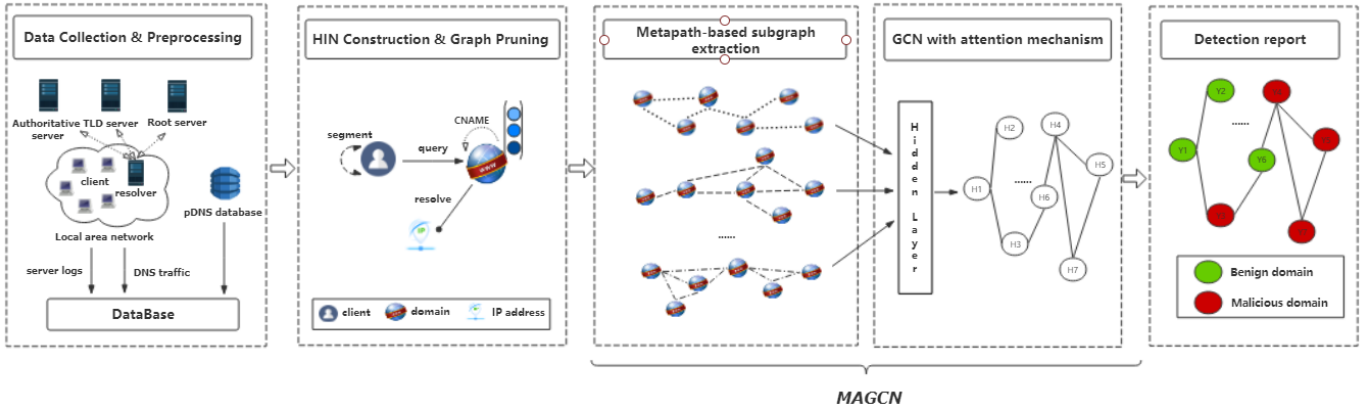


Fig. 2. The system architecture of HGDom

and construct a pDNS dataset to record timestamps and counts of resolution relations. For example, record $\langle T_f, T_l, n, d, i \rangle$ depicts the resolution from domain d to IP i is first captured at T_f and observed n times until the last time T_l . Organizations like Farsight Security [19], Virustotal [20] and Qihu 360 [21] all provide good API services for their pDNS dataset access; iii) DNS log. DNS servers record domain queries in log with fields like time, domain name, source IP, etc. Logs from the local resolver are widely used to reflect clients query behaviors in a network.

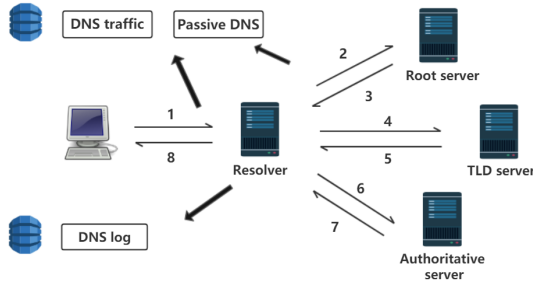


Fig. 3. Domain query process and the related data

Note that HGDom can construct its HIN model with only DNS response traffic. We utilize DNS logs and pDNS dataset for data verification and enrichment. For example, DNS logs can verify information about clients' query behaviors while pDNS dataset can enrich the mapping relation between domains and IP addresses with longer time and broader space.

B. HIN Model Construction

Most real-world systems are heterogeneous with plenty of interacting, multi-typed components. Recently, instead of simply modeling these systems as homogeneous networks, researchers turn to HIN to handle heterogeneity and represent richer semantics. In HGDom, we start the DNS excavation by first constructing a HIN to model it.

Definition 1: Heterogeneous Information Network (HIN) [22]. Given a directed graph $G = \langle V, E, X \rangle$, where V is the set of nodes, E is the set of links. Functions $\varphi : V \rightarrow Z$ and

$\psi : E \rightarrow R$ map types for nodes and links respectively, where Z is the set of object types and R is the set of link types. We regard G as a HIN if $|Z| \geq 2$ or $|R| \geq 2$, and $T_G = \langle Z, R \rangle$ is the schema of G . When $|Z| = |R| = 1$, G degenerates to a homogeneous network. Node $v_i \in V$ in a HIN may have some attributes. We use $X = \{x_i | v_i \in V\}$ to denote the set of attributes for all nodes, where x_i represents attributes of node v_i . Note that different types of nodes are usually associated with different feature vectors.

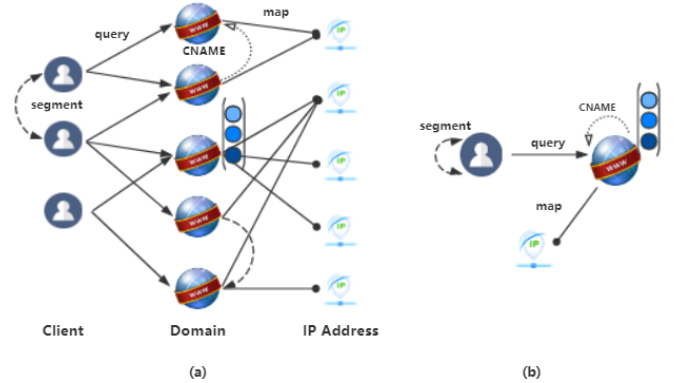


Fig. 4. The HIN model(a) and its network schema(b) in HGDom

TABLE I
DESCRIBE RELATIONS WITH ADJACENT MATRICES

Matrix	Element	Description
Q	$q_{i,j}$	$q_{i,j}=1$ if client i queries domain j , otherwise $q_{i,j}=0$.
M	$m_{i,j}$	$m_{i,j}=1$ if domain i maps to IP j , otherwise $m_{i,j}=0$.
S	$s_{i,j}$	symmetric, $s_{i,j}=s_{j,i}=1$ if client i and j are from the same network segment, otherwise $s_{i,j}=s_{j,i}=0$.
D	$d_{i,j}$	symmetric, $d_{i,j}=d_{j,i}=1$ if domain i and j appear in a CNAME record, otherwise $d_{i,j}=d_{j,i}=0$.

Figure 4 displays the HIN instance and its network schema in HGDom, which contains three types of nodes (i.e. $c \leftarrow$ clients, $d \leftarrow$ domains, $ip \leftarrow$ IP addresses) and four

types of relations: i) *query*, clients query domains, ii) *map*, domains map to IP addresses, iii) *segment*, two clients belong to the same network segment, iv) *cname*, domain *a* is the canonical name of domain *b*. Table I explains these relations with adjacent matrices. For features of domain nodes, we capture the character distributions of domain names by directly processing the name strings with one-hot encoding.

Raw data directly collected from network is dirty with noisy nodes. To improve performance and efficiency, we preprocess the HIN by applying graph pruning according to the following conservative rules. These nodes are either unlikely to be malicious or of little help in information propagation.

- **Inactive clients.** Clients querying less than K_f (for example, $K_f = 2$) domains are considered inactive and are discarded for inutilty in mining domains' associations.
- **Large clients.** Clients querying more than $K_c\%$ (for example, $K_c = 90$) of domains are mostly forwarders or proxies and should be removed to reduce noises.
- **Irregular domains.** Domains fail to meet the name rules (*RFC 1035*) and domains queried by only one client are removed for lack of helpful information.
- **Popular domains.** Domains queried by more than $K_q\%$ (for example, $K_q = 50$) of clients are removed as they are in very low risk of being malicious; otherwise, they will cause significant attack events easily detected by IDS.
- **Rare IPs.** IP address that maps to only one domain is discarded for being unhelpful in label propagation.

Note that some attackers may try to cover up their traces by reducing activities. Domains with clear malicious labels are kept even if they belong to the unusual category, so do their related clients and IP address nodes.

C. Meta-path Generation

Definition 2: Meta-path [23]. Given a HIN $G = \langle V, E, X \rangle$, with node type set Z and link type set R , a meta-path P defines the composite relation $R_1 \circ R_2 \circ \dots \circ R_L$ between Z_1 and Z_{L+1} , where $\circ$ is a relation composition operator. P is denoted as $Z_1 \xrightarrow{R_1} Z_2 \xrightarrow{R_2} \dots \xrightarrow{R_L} Z_{L+1}$, where L is the length of the meta-path. We can also use $Z_1 Z_2 \dots Z_{L+1}$ for short when no ambiguity is caused.

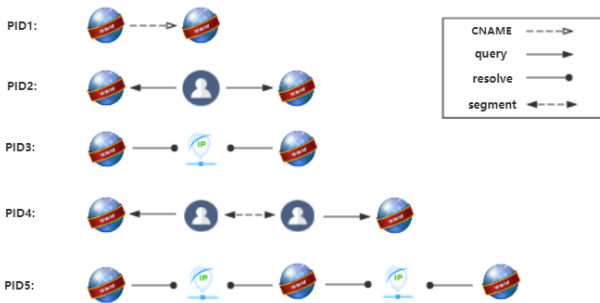


Fig. 5. Meta-paths designed in HGDom

As HGDom aims at detecting malicious domains, we only choose symmetric meta-paths where $Z_1 = Z_{L+1} = \text{domain}$.

As shown in Figure 5, we design five meta-paths to model the associations among domains. Among them, PID1 describes the fact that a domain tends to fall into the same category as its canonical name (CNAME) domain. PID2 is composed of the query behavior between clients and domains. It represents the fact that the malicious domain sets queried by victims of the same attacker are likely to partially overlap while benign clients have no reason to query them. PID3 is based on the mapping relations between domains and IP addresses. As the ownership of IP resource is relatively stable, domains resolved to the same IP address in a period tend to belong to the same category. In addition, take plenty of observations about cyber attacks into consideration, we design PID4 and PID5 to further excavate higher semantics hiding in the HIN. PID4 implies the observation that adjacent clients are inclined to be vulnerable to the same attackers in some cases, for example, a spam attack aims at a subnet or malware propagates in a network segment. PID5 describes domains resolved to IP addresses that have once been mapped to the same domain, which is chosen based on the observation that attackers tend to reuse their domain or IP resources due to funding limits. These five meta-paths, together with features of domain nodes, represent the three aforementioned intuitions about character distribution, resources aggregation, and query behaviors of malicious domains.

D. Proposed method: MAGCN

With the HIN model, a meta-path schema and label information for part of the domains, the task in HGDom can be generalized as a transductive classification problem in HIN [24] as bellow.

Transductive Classification in HIN. Given a HIN $G = \langle V, E, X \rangle$, V_T is designated to represent the target set of nodes which need to be estimated, for example, the domain nodes in HGDom. $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$ respectively denote the feature set and label set for all instances in V_T . We use a subset $V_L \subseteq V_T$ to denote the labeled nodes and $V_U = V_T - V_L$ to denote the unlabeled ones. Then the task of transductive classification in HIN is to predict labels for nodes in V_U , namely to estimate:

$$Pr(Y_{V_U} | X, Y_{V_L}) \propto \prod_{v_i \in V_U} Pr(Y_i | x_i, Y_{N_i})$$

where $N_i \subseteq V_T$ is composed of nodes associated with v_i and $Y_{N_i} = \{Y_i | v_i \in N_i\}$ denotes their labels.

The problem is challenging because the node features and diverse relations in HIN need to be considered simultaneously. To tackle this issue, we propose MAGCN, which can be regarded as a variant of basic GCN models. In the following content, we first briefly introduce the basic GCN model, then present the key points of MAGCN in details.

Basic GCN model. Recently, GCNs have achieved promising performance in graph mining, as they can leverage node features together with the network structural information. One of the representative work is done by Kipf and Welling [25]

(shown in Figure 6). They propose a 1-order approximation of Chebyshev polynomials and define the convolution layer as:

$$H^{(l+1)} = \sigma(D^{-1/2} \tilde{A} D^{-1/2} H^{(l)} W^{(l)})$$

where $\tilde{A}$ is the adjacency matrix with self-connections. D is a diagonal matrix where $D_{ii} = \sum_j \tilde{A}_{ij}$. σ denotes an activation function. $H^{(l)}$ is the matrix of the l -th hidden layer. $W^{(l)}$ is the trainable parameters of l -th layer.

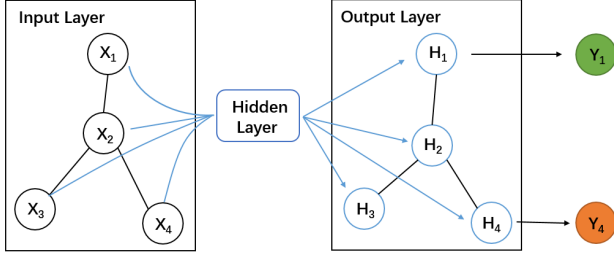


Fig. 6. Basic Graph Convolutional Network [25]

After that, many improvements and extensions are proposed, which are mostly based on the key idea that the representation of a node can be generated by exploiting its neighbors with an iterative aggregation method. However, the existing studies focus on homogeneous networks, and it is not suitable to directly employ them in HGDom. Therefore, to fully reflect our observations in DNS data, we propose a heterogeneous GCN method named MAGCN. As shown in Figure 2, MAGCN includes two phases: i) subgraph extraction, which translates the HIN to a set of homogeneous networks where we can apply convolution operations; ii) attention-based aggregation, which supports to learn the final representation by aggregating diverse subgraphs with different importance.

Subgraph extraction. HGDom aims at predicting labels for domains. Therefore we can simplify the HIN model to a set of subgraphs with domain nodes according to the five metapaths $P = \{PID_1, \dots, PID_5\}$ designed in Section 4.2. We define the subgraph $G'_i = (V_T, A'_i)$, where $A'_i \in R^{|V_T| \times |V_T|}$ denotes associations among nodes in V_T based on the metapath PID_i . As shown in Table II, the adjacent matrix A'_i of each subgraph can be calculated from the commuting matrix C of each meta-path in original HIN.

TABLE II
DESCRIBE META-PATHS WITH COMMUTING MATRICES

PID	meta-path	Commuting Matrix C
1	$d \xrightarrow{D} d$	D
2	$d \xrightarrow{Q^T} c \xrightarrow{Q} d$	$Q^T Q$
3	$d \xrightarrow{M} ip \xrightarrow{M^T} d$	MM^T
4	$d \xrightarrow{Q^T} c \xrightarrow{S} c \xrightarrow{Q} d$	$Q^T S Q$
5	$d \xrightarrow{M} ip \xrightarrow{M^T} d \xrightarrow{M} ip \xrightarrow{M^T} d$	$MM^T MM^T$

Attention mechanism. We have five subgraphs characterizing associations among domains over diverse views which are

not of equal importance in malicious domain detection. Thus, attention mechanism is employed to incorporate attention coefficients into MAGCN, as it has recently achieved promising performance in many graph classification or link prediction tasks [26]. Instead of averaging the results generated from each subgraph, we aggregate them with weights estimated adaptively and get

$$H^{(l+1)} = \sigma(X \cdot W_0 + \sum_{i \in |P|} f(\alpha_i) \cdot A'_i \cdot H^{(l)} \cdot W_{PID_i})$$

where $H^{(l)}$ is the representation after l layers aggregation, $X \in R^{N \times U}$ is the feature matrix of domain nodes with $N = |V_T|$ denoting the number of domains, U denoting the dimension of features. $A'_i \in R^{N \times N}$ is the adjacent matrix of subgraph G'_i . $W_0 \in R^{U \times k}$ and $W_{PID_i} \in R^{k \times k}$ are trainable parameters with k denotes the embedding size. $f(\alpha_i) = \exp(\alpha_i) / \sum_{i \in |P|} \exp(\alpha_i)$ is the softmax function and $\alpha_i, i \in |P|$ are also trainable parameters.

To estimate these trainable parameters effectively, we use a standard logistic loss function:

$$\min L(W_0, W_{PID}, \alpha, w) = - \sum_{i=1}^{N_L} \log \phi(y_i \cdot (w^T h_i))$$

where $\phi(x) = 1/(1 + \exp(-x))$ is the logistic function and $N_L = |V_L|$ denotes the number of labeled domains.

Note that, though designed in the malicious domain detection task, MAGCN can also be employed in other scenarios for node classification in HIN. The main differences between MAGCN and the basic GCN are i) MAGCN extends the applications of GCN from homogeneous networks to heterogeneous ones; ii) With the meta-path-based attention mechanism, MAGCN can comprehensively aggregate information from diverse perspectives. Algorithm 1 summarizes the overall process of MAGCN.

Algorithm 1 GCN with Metapath-based Attention mechanism

Input: HIN instance $G=(V, E, X)$, a metapath schema P , target node type T , embedding size k , the depth of layers L

Output: Representation of nodes $H \in R^{|V_T| \times k}$

for PID_i in P **do**

$A'_i \leftarrow \text{SubgraphExtract}(G, PID_i)$

$H^{(0)} \leftarrow X_T = \{x_i | v_i \in V_T\}$

for l in $\text{range}(0, L)$ **do**

$H^{(l+1)} \leftarrow \sigma(X \cdot W_0 + \sum_{i \in |P|} f(\alpha_i) \cdot A'_i \cdot H^{(l)} \cdot W_{PID_i})$

return $H^{(L)}$

IV. EXPERIMENTS

In this section, we first validate the superiority of MAGCN by comparing it with three baseline methods, then investigate its sensitivity to some hyperparameters. Finally, we compare our HGDom system with there other existing malicious domain detection approaches.

A. Setup

The experiments are conducted on two real-world DNS datasets: i) The first is collected from TUNET, the campus network of Tsinghua University. According to statistics, we find that about 1.5 million unique domains are queried by over 25 million clients per day. We employ this dataset to evaluate the performance of MAGCN. ii) The second dataset collected from CERNET2 is much bigger, as we intend to test the effectiveness of HGDom over longer periods. CERNET2 is the second generation of China Education and Research Network. It is the first IPv6 national backbone network in China and the world's largest next-generation Internet backbone network with pure IPv6 technology. At present, CERNET2 has 41 core nodes distributed in 36 cities with 10G~100Gbps bandwidth and provides IPv6 access services for over 10 million users in about 600+ research institutes. We capture DNS traffic in CERNET2 at Tsinghua node during one month.

We construct HIN instance based on fields like *ip.src*, *dns.a*, *dns.qry.name*, etc. extracted from DNS traffic, verify information among clients and domains with DNS logs and enrich the resolution information among domains and IP addresses with pDNS dataset from Qihu 360 [21]. We discard unhelpful nodes according to the graph pruning rules mentioned in Section 4.1 by empirically setting $K_f = 3$, $K_c = 0.5$, and $K_q = 25$. For label information, we regard domains whose 2LD (second-level domain) appears in Alexa Top 10k [18] and our local whitelist as benign. We use multiple sources like DGArchive [1], Malware-domains.com [27], Malware-domainlist.com [28], etc. to label malicious domains. Integrated services like Google Safe Browsing [29] and VirusTotal [20] are also employed to supplement label information. Besides, we manually investigate dubious domains which have both benign and malicious information. We implement MAGCN and a prototype of HGDom in Python 2.7 based on Euler [30], a large-scale distributed graph learning framework open-sourced by Alibaba. Table III shows the evaluation metrics used in the experiments. For each experiment, we randomly select 70% domains for training, leaving the remained 30% as test samples and conduct 5-fold cross-validation for 10 times to get the average results.

B. Performance evaluation of MAGCN

There exist some other approaches to handle node classification task in graphs. In this set of experiments, we compare MAGCN with the following three representative methods over the TUNET dataset. Note that, for methods on homogeneous graphs, we ignore the heterogeneity of our model and directly deal with the HIN without distinguishing between different types of nodes and links.

- **DeepWalk [31]**. As a homogeneous network embedding method, DeepWalk generates node sequences with truncated random walks and gets node representation vectors through Skip-gram model.
- **Metapath2vec [32]**. It extracts structural information in HIN by performing random walks on meta-paths.

TABLE III
EVALUATION METRICS

Metric	Description
TP	malicious domains labeled as malicious
FP	benign domains labeled as malicious
TN	benign domains labeled as benign
FN	malicious domains labeled as benign
accuracy	$(TP + TN)/(TP + FP + TN + FN)$
precision	$TP/(TP + FP)$
recall	$TP/(TP + FN)$
F1	$2 \times (precision \cdot recall)/(precision + recall)$
ROC	a curve plotting TPR against FPR with various thresholds
AUC	area under the ROC curve

Then it also utilizes the Skip-gram model to learn node representations.

- **Basic GCN [25]**. It is one of the classic Graph Neural Network methods and its core idea has already been introduced in Section 4.3.

Table IV shows the performance of each algorithm, and Figure 7 displays their ROCs. From the results, we can see that the basic GCN model achieves better performance than DeepWalk, as it not only considers the graph structure but also deals with node features at the same time. Metapath2vec works better than DeepWalk and GCN, which we think is because, in the HIN instance of malicious domain detection, information hidden in diverse relations is of greater importance than that contained in domains themselves. Our method MAGCN outperforms all the compared methods, as it can jointly handle node features together with the heterogeneous structural information, and it is able to estimate the importance of each type of meta-path adaptively.

TABLE IV
COMPARISON ON MAGCN WITH BASELINE METHODS

Method	accuracy	precision	recall
DeepWalk	0.7557	0.6926	0.6983
Metapath2vec	0.9384	0.8915	0.8471
Basic GCN	0.8750	0.8302	0.8424
MAGCN	0.9892	0.9752	0.9817

C. Parameter Sensitivity

There are two hyperparameters in MAGCN, namely the embedding size k and the depth of hidden layers L . We investigate their impacts on the performance of HGDom.

Figure 8 and Figure 9 respectively illustrate the ACC and F1-score under different values of k and L . As we can see, the performance of HGDom remains relatively stable when the embedding size k varies exponentially from 8 to 256, which means with proper regularizations, the hyperparameter k has little impact on malicious domain detection. Meanwhile, the depth of hidden layers L does have an impact

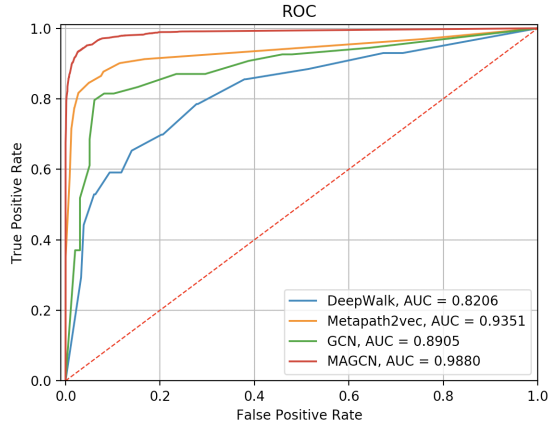


Fig. 7. ROC for DeepWalk, Metapath2vec, basic GCN and MAGCN

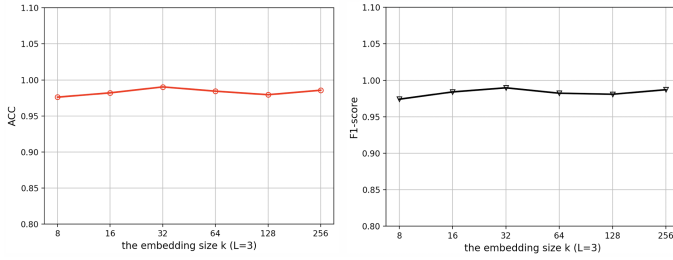


Fig. 8. ACC(a) and F1-score(b) with varying embedding size k

on HGDom. At first, HGDom performs better with deeper neural architectures, as it can aggregate richer information from neighbors. However, after the number of layers exceeds a certain threshold, HGDom's performance stops rising and suffers some drops. It is because convolution operations in a graph tend to aggregate information of a node with that of its adjacent nodes, therefore, in theory, after infinite times of iterations, the representation of all nodes will converge [33] and thus be difficult to distinguish. Take performance and computation complexity into consideration, we recommend setting the depth of hidden layer L around 4 in HGDom.

D. Comparisons of HGDom with other detection methods

In this set of experiments, we compare HGDom with the following three state-of-the-art malicious domain detection systems on DNS data collected from CERNET2.

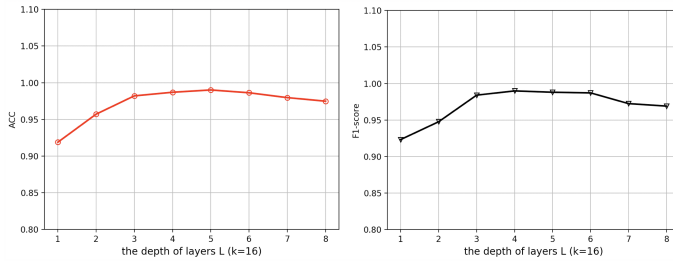


Fig. 9. ACC(a) and F1-score(b) with varying depth of hidden layers L

- **Manadhata et al [11].** It excavates proxy logs, constructs a host-domain graph, labels part of the graph with ground truth, then employs belief propagation to calculate the probability of an unlabeled domain being malicious. We will mention this system as MDS-I for short.
- **Khalil et al [13].** This system constructs a IP-domain resolution graph from passive DNS data, then applies a label propagation algorithm to discover malicious domains. We will mention this system as MDS-II for short.
- **HinDom [16].** This is a system we proposed recently. It models the DNS scene as a HIN, focuses on relations among domains, clients, and IP addresses and then makes malicious domain detection with a traditional transductive node classification method.

To investigate these systems' performance over a longer period, we collect DNS-related data from CERNET2 in one month and split the data into four consecutive weeks. In each week, we gather label information from the first six days and try to discover malicious domains newly coming in the seventh day. Note that in this experiment, we rebuild MDB-I and MDB-II based on our understanding of the papers, as neither their datasets nor system implementations are available. Table V presents the ACC and F1-score of each system. We can see that HinDom and HGDom outperform the other two systems, as they are able to fuse higher-level semantics in DNS scene with the help of HIN model, while MDB-I and MDB-II can only capture no more than half of the information due to the limitations of using bipartite graph. Besides, HGDom works better than HinDom, which is because the traditional transductive classification method used in HinDom only focus on the relations among nodes and cannot handle the character distribution features of domains as felicitously as HGDom.

TABLE V
PERFORMANCE OF DIFFERENT GRAPH-BASED DETECTION SYSTEMS

		Week 1	Week 2	Week 3	Week 4
MDS-I	ACC	0.9158	0.9221	0.9362	0.9231
	F1-score	0.8947	0.9286	0.9274	0.9241
MDS-II	ACC	0.9236	0.9361	0.9275	0.9153
	F1-score	0.9279	0.9398	0.9211	0.8958
HinDom	ACC	0.9583	0.9558	0.9590	0.9603
	F1-score	0.9528	0.9501	0.9517	0.9588
HGDom	ACC	0.9847	0.9913	0.9797	0.9885
	F1-score	0.9812	0.9886	0.9741	0.9800

V. RELATED WORK

Malicious Domain Detection. Notos [3] and Exposure [4] assign reputation scores to domains by analyzing Passive DNS, a dataset stores DNS resolution record and timestamps. Kopis [5] gets a more global view by utilizing traffic among top-level-domain servers while Pleiades [6] focuses on NXDomain traffic with clustering and classification methods. Taking privacy into consideration, some works try to detect domain

generation algorithms (DGAs) by focusing on the character distribution of domain names. FANCI [7] trains SVM and Random Forest models with 21 character-related features to detect DGA in NXDomain traffic while D3N [8] analyzes NXDomain traffic with three different deep neural network models. Ryan R et.al [9] detect several DGA families by combining RNN architecture with domain registration side information. While the above methods treat each domain individually, some studies try to be more robust and get comprehensive results by excavating associations among domains. Manadhata et al. [11] and Segugio [12] focus on “*who is querying what*” and apply belief propagation algorithm on a client-domain graph. Khalil et al. [13], [14] dig into passive DNS by building a domain-IP graph for detection. Futai Zou et al. [15] try to utilize information on both sides by constructing two graphs: a DNS query response graph and a passive DNS graph. However, with the bipartite graph model, they cannot handle multi-type components or complex relations, leaving much valuable information untapped. Our recent work HinDom [16] manages to fuse more information based on a heterogeneous information network (HIN) model. Nevertheless, due to the limitations of traditional graphic analysis methods, these work can only utilize relations among domains, without taking domains’ features into consideration. In this paper, we propose a more accurate and robust system named HGDom to fully exploit the DNS scenario and detect malicious domains by considering both domains’ features and their associations.

Graph Convolutional Networks. As deep learning revolutionizes various fields, researchers put forward Graph Convolution Network (GCN) to extend its applications from Euclidean data (e.g. text, image and voice) to complex networks. The key point of GCNs is to learn a function to generate nodes’ representations by iteratively aggregating feature information from their neighbors. GCNs have achieved promising performance in many tasks. For example, researchers build graphs with atoms edged by chemical bonds and use GCN to learn molecular fingerprints [34]. Rex Ying et.al [35] construct a graph of pins and boards on Pinterest and use GCN to make high-quality recommendations for users. DeepMem [36] uses GCN on code segments graphs for robust kernel data structure detection. As far as we know, our research is the first work to apply GCN in the field of cyber security analysis.

Heterogeneous Information Network. Recently, many researchers start to focus on Heterogeneous Information Network (HIN) as it can model diverse relations among multi-type components in complex systems. HIN has been applied to various fields, such as information fusion, link prediction, recommender system etc [37]. Many studies are proposed for representation learning in HIN. Research like ESIM [38], metapath2vec [32] and HIN2vec [39] guide random walk to obtain node sequence, then get node representation with models like Skip-gram or CBOW. PTE [40] first decomposes the HIN to homogeneous networks and fuses the results after representing them separately. HNE [41] and BL-MNE [42] choose to handle nonlinear relations with the help of deep

learning models. Unfortunately, these methods only focus on excavate associations and cannot be directly applied to our task, which needs to exploit domains’ features and associations simultaneously. To solve this problem, we propose a variant of GCN named MAGCN, which can handle node attributes and graph structure in HIN at the same time.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a malicious domain detection system called HGDom, which is designed on characteristics of malicious domains from three perspectives, namely character distribution of names, resources aggregation of attackers, and query behaviors of clients. To make accurate and robust detection, HGDom naturally models the DNS scene as a HIN for richer information fusion. Then we propose MAGCN and apply it to the HIN to classify domain nodes with the joint consideration of both domains’ features and their associations. The key point of MAGCN is a variant of GCN model with a meta-path-based attention mechanism. Adequate experiments are conducted on DNS data collected from TUNET and CER-NET2 to prove the accuracy and superiority of HGDom, where MAGCN outperforms the state-of-the-art network embedding methods, and HGDom makes better detection comparing with the existing graph mining-based systems. At present, HGDom only contains three components (i.e., clients, domains, and IP addresses). For part of our future work, we plan to add other types of DNS-related data like WHOIS, the registration information dataset, for more comprehensive analysis. Besides, we intend to propose more advanced data mining methods to further improve the efficiency and scalability of HGDom in the future.

ACKNOWLEDGMENT

We thank Mingkai Tong, Hui Zhang, Shize Zhang for constructive comments on experiments and data processing. Additionally, we thank Information Technology Center of Tsinghua University for authorizing the use of their data in our experiments. This work is supported by the National Key Research and Development Program of China under Grant No.2017YFB0803004.

REFERENCES

- [1] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, “A comprehensive measurement study of domain generating malware,” in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 263–278.
- [2] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, “A survey on malicious domains detection through dns data analysis,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 67, 2018.
- [3] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, “Building a dynamic reputation system for dns,” in *USENIX security symposium*, 2010, pp. 273–290.
- [4] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, “Exposure: A passive dns analysis service to detect and report malicious domains,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 16, no. 4, p. 14, 2014.
- [5] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon, “Detecting malware domains at the upper dns hierarchy,” in *USENIX security symposium*, vol. 11, 2011, pp. 1–16.

- [6] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of dga-based malware," in *USENIX security symposium*, vol. 12, 2012.
- [7] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "{FANCI}: Feature-based automated nxdomain classification and intelligence," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1165–1181.
- [8] M. Tong, X. Sun, J. Yang, H. Zhang, S. Zhu, X. Liu, and H. Liu, "D3n: Dga detection with deep-learning through nxdomain," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2019, pp. 464–471.
- [9] R. R. Curtin, A. B. Gardner, S. Grzonkowski, A. Kleymenov, and A. Mosquera, "Detecting dga domains with recurrent neural networks and side information," *arXiv preprint arXiv:1810.02023*, 2018.
- [10] H. S. Anderson, J. Woodbridge, and B. Filar, "Deepdga: Adversarially-tuned domain generation and detection," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*. ACM, 2016, pp. 13–21.
- [11] P. K. Manadhata, S. Yadav, P. Rao, and W. Horne, "Detecting malicious domains via graph inference," in *European Symposium on Research in Computer Security*. Springer, 2014, pp. 1–18.
- [12] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks," in *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*. IEEE, 2015, pp. 403–414.
- [13] I. Khalil, T. Yu, and B. Guan, "Discovering malicious domains through passive dns data graph analysis," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 663–674.
- [14] I. M. Khalil, B. Guan, M. Nabeel, and T. Yu, "A domain is only as good as its buddies: Detecting stealthy malicious domains via graph inference," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. ACM, 2018, pp. 330–341.
- [15] F. Zou, S. Zhang, W. Rao, and P. Yi, "Detecting malware based on dns graph mining," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 102687, 2015.
- [16] X. Sun, M. Tong, J. Yang, L. Xinran, and L. Heng, "Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, 2020.
- [17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.
- [18] Amazon Web Services, Inc., "AWS — Alexa Top Sites - Up-to-date lists of the top sites on the web," <https://aws.amazon.com/alexa-top-sites/>, 2019, [Online].
- [19] Dnsdb.info, "DNSDB," <https://www.dnsdb.info>, 2019, [Online].
- [20] Virustotal.com, "VirusTotal," <https://www.virustotal.com>, 2019, [Online].
- [21] Passivedns.cn, "Sign In-passiveDNS," <https://passivedns.cn>, 2019, [Online].
- [22] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 797–806.
- [23] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.
- [24] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 570–586.
- [25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [26] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *arXiv preprint arXiv:1807.07984*, 2018.
- [27] Malwaredomains.com, "DNS-BH – Malware Domain Blocklist by Risk-Analytics," <http://www.malwaredomains.com>, 2019, [Online].
- [28] Malwaredomainlist.com, "MDL," <https://www.malwaredomainlist.com>, 2019, [Online].
- [29] Google, "Google Safe Browsing," <https://developers.google.com/safe-browsing>, 2019, [Online].
- [30] Alibaba, "Euler," <https://github.com/alibaba/euler>, 2019, [Online].
- [31] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.
- [32] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2017, pp. 135–144.
- [33] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [34] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1263–1272.
- [35] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 974–983.
- [36] W. Song, H. Yin, C. Liu, and D. Song, "Deepmem: Learning graph neural network models for fast and robust memory forensic analysis," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 606–618.
- [37] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.
- [38] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, "Meta-path guided embedding for similarity search in large-scale heterogeneous information networks," *arXiv preprint arXiv:1610.09769*, 2016.
- [39] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1797–1806.
- [40] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1165–1174.
- [41] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 119–128.
- [42] J. Zhang, C. Xia, C. Zhang, L. Cui, Y. Fu, and S. Y. Philip, "Bl-mne: emerging heterogeneous social network embedding through broad learning with aligned autoencoder," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 605–614.