Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

HANDOM: Heterogeneous Attention Network Model for Malicious Domain Detection

Qing Wang[a,b], Cong Dong[c], Shijie Jian[d], Dan Du[a,b], Zhigang Lu[a,b], Yinhao Qi[a,b], Dongxu Han[a,b], Xiaobo Ma[e], Fei Wang[f], Yuling Liu[a,b,*]

[a] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[b] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[c] Zhongguancun Laboratory, Beijing, China
[d] The First Research Institute of the, Ministry of Public Security of P.R.C, Beijing, China
[e] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[f] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 24 May 2022
Revised 12 October 2022
Accepted 8 December 2022
Available online 9 December 2022

Keywords:
Malware domain detection
Spatial-Temporal contextual correlation
Heterogeneous attention network
Statistical-and-Structural information

ABSTRACT

Malicious domains are crucial vectors for attackers to conduct malicious activities. With the increasing numbers in domain-based attack activities and the enhancement of attacker evasion methods, the detection of malicious domains has become critical and increasingly difficult. Statistical feature-based and graph structure-based detection methods are mainstream technical approaches. However, highly hidden domains can escape feature detection, and the detection range of graph structure-based methods is limited. Based on these, we propose a malicious detection method called HANDOM. HANDOM combines statistical features and graph structural information to neutralize their limitations, and uses the Heterogeneous Attention Network (HAN) model to jointly handle both information to achieve high-performance malicious domain classification. We conduct experimental evaluations on real-world datasets and compare HANDOM with machine learning methods and other malicious detection methods. The results present that HANDOM has superior and robust performance, and can identify highly hidden domains.

1. Introduction

The Domain Name System (DNS) Mockapetris (1989) serves as the infrastructure of the Internet, enabling the mapping of domains to IP addresses. As the Internet grows, attackers use dynamic DNS-based agile techniques to achieve highly variable FQDN and IP address mappings for malicious activities, such as launching APT attacks, manipulating botnets Hao et al. (2009), spreading malware On (2016), locating Command and Control Server (C&C Server) Abley (2014); Antonakakis et al. (2012); Kara et al. (2014); Zhao et al. (2015), performing covert communication, etc. Therefore, the detection of malicious domains as the main attack medium is crucial Zhauniarovich et al. (2018).

Existing malicious domain detection methods are mainly divided into two categories: statistical feature-based detection methods and graph structure-based detection methods. Statistical feature-based detection methods analyze DNS data and select statistical features that can distinguish malicious behavior of domains, and then train models using machine learning or deep learning algorithms to discover more malicious domains. However, the effectiveness of the features gradually decreases with the development of defense evasion techniques. Researchers start to use graph structure-based detection methods. Graph structure-based detection methods consider the association relationships between domains and construct domain graphs based on structural information, to discover unknown malicious domains associated with known malicious domains in the graph. This type of detection technique is robust for defense, but cannot identify malicious domains that are not associated with known domains in the graph.

These two categories of studies complement each other well. At first glance, one can simply deploy two types of detection systems, namely, statistical feature-based and graph structure-based systems, to achieve both high adaptability and coverage. How-

[*] Corresponding author.
E-mail addresses: wangqing@iie.ac.cn (Q. Wang), dongcong@iie.ac.cn (C. Dong), jianshijie@iie.ac.cn (S. Jian), dudan@iie.ac.cn (D. Du), luzhigang@iie.ac.cn (Z. Lu), qiyinhao@iie.ac.cn (Y. Qi), handongxu@iie.ac.cn (D. Han), liuyuling@iie.ac.cn (X. Ma), xma.cs@xjtu.edu.cn (F. Wang), wangfei@ict.ac.cn (Y. Liu).

https://doi.org/10.1016/j.cose.2022.103059
0167-4048/© 2022 Elsevier Ltd. All rights reserved.

ever, concurrently deploying two types of detection systems is not cost-effective. More importantly, deploying two separate detection systems, though complementing each other, cannot seamlessly and automatically facilitate each other with their detection results. Seamlessly combining two types of detection systems to facilitate each other is of fundamental importance. The reasons are twofold. First, the statistical feature-based detection method continuously adjusts behavior patterns of malicious domains to achieve high adaptability, which can provide the graph structure-based detection method with malicious behavior pattern mining of domains that are not associated with known malicious domains in the graph. Second, the graph structure-based detection method can build strong connections between domains and discover potentially unknown malicious domains associated with them based on known malicious domains, thus achieving high accuracy and coverage detection. We develop our study based on these two types of information.

In this paper, we observe the behavioral differences between malicious and benign domains in terms of spatial-temporal correlation, which are the interactive behavior patterns of malicious domains in the attack period and the resource distribution of malicious domains are different from benign domains, and then we propose HANDOM based on these observations. HANDOM first constructs a heterogeneous graph to represent the resource sharing relationship between domains based on the correlation of spatial context, that is, the query behaviors and registration behaviors between domains, client hosts and registrants. Then, HANDOM extracts the time-series features for each domain node in the graph based on the behavioral patterns of temporal contextual correlation. Finally, HANDOM uses the HAN model to synthesize the global graph structure information and local domain features. HAN uses node-level and semantic-level attention mechanisms to obtain the final embedding vector for each domain in the graph, and transforms the malicious domain detection problem into a HAN-based node classification. Finally, we demonstrate the superiority of HANDOM in multidimensional experiments.

The main contributions of this paper are summarized as follows:

- We propose a method based on Heterogeneous Attention Network model to detect malicious domains. We model the interactive relationships between domains, client hosts, and domain registration information, and design multiple meta-paths as well as heterogeneous graphs, and extract feature attributes for each domain node in the graph. We use the HAN model to learn the maliciousness of domains and transform the malicious domain detection problem into a HAN-based node classification.
- We propose a HANDOM method that can detect highly hidden malicious domains, which uses resources sharing based on spatial context correlation and behavioral patterns information based on temporal context correlation, to capture attack behaviors of domains at both structural-local levels. Experiments prove that HANDOM is effective in detecting highly hidden malicious domains.
- We show our method's high robust performance by varying the number of training label domains. Meanwhile, we demonstrate the superiority of our method by comparing it with traditional machine learning methods and other existing detection methods.

We organize the rest of the paper as follows. Section 2 reviews related work of malicious domain detection. Section 3 describes our method motivation. Section 4 introduces the preliminary. Section 5 introduces the method architecture of HANDOM. Section 6 presents the experimental results. Section 7 shows the limitations and future work of HANDOM. Section 8 concludes the paper.

2. Related works

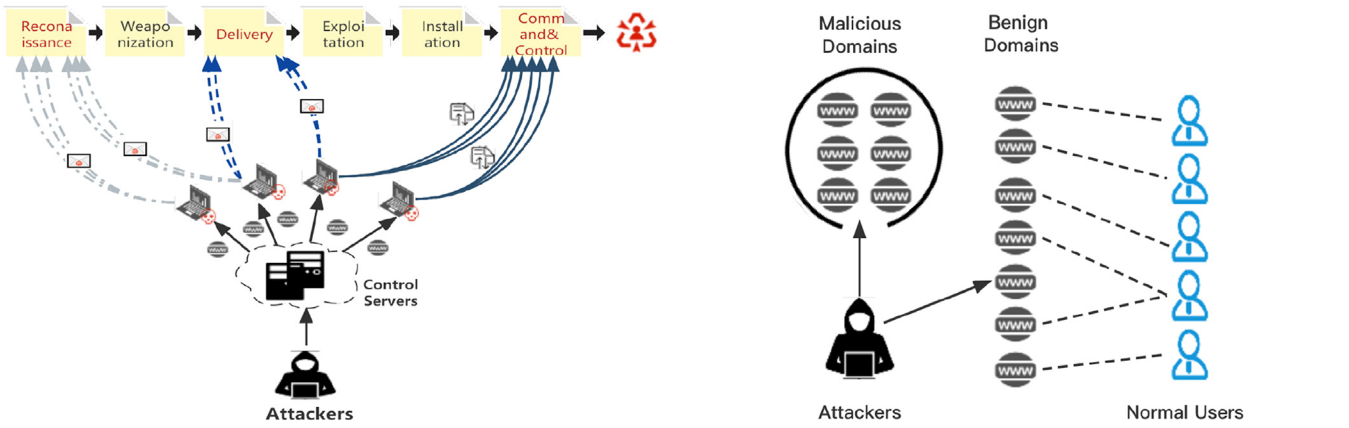
With the development of attacker technology and changes in domain production mechanisms, traditional blacklist and whitelist-based detection methods cannot cope with the large amount of DNS data generated in the network. Researchers investigate detection models that can automatically detect malicious domains. There are two mainstream technical methods for detecting malicious domains: statistical feature-based and graph structure-based methods.

2.1. Statistical feature-based methods

In statistical feature-based detection methods, researchers observe the differences between malicious and benign domains in DNS data, and extract statistical features from DNS data that can effectively distinguish malicious domains, and then use machine learning or deep learning algorithms to train malicious domain detection models.

Researchers usually determine the maliciousness of domains based on their behavioral features during network activities. For example, Bilge et al. Bilge et al. (2014) proposes EXPOSURE, which extracts four types of network traffic features from passive DNS data and uses machine learning algorithms to detect any domain associated with malicious activities. Samuel et al. Schüppen et al. (2018) uses machine algorithms such as Random Forest (RF) to identify unknown Algorithmically Generated Domains (AGDs) by extracting domain character-level features from NXDomain data. Kountouras et al. Kountouras et al. (2016) extracts correlation features in DNS queries and uses only a small number of known malicious domains to calculate the similarity between unknown domains and known malicious domains. Vissers et al. Vissers et al. (2017) is the first study to shift the focus of detection to malicious activities, which uses a hierarchical clustering algorithm to classify domains according to different types of malicious activities. He et al. He et al. (2019) extracts domain character features, PDNS features, and the domains relationship features by a modified graph embedding algorithm from Passive DNS data. Experiments have proven that these features achieve better results. Park et al. Park et al. (2022) extracts features based on domains linguistic patterns, and then uses an unsupervised approach to detection malicious domains. In addition to using machine learning algorithms for detection, deep learning-based detection methods are popular because they eliminate manual feature extraction. For example, Bharathi et al. Bharathi and Bhuvana (2019) uses the Long Short-Term Memory (LSTM) model and the bidirectional LSTM model to automatically extract useful features of AGD domains and use them for training the detection model. Some researchers use deep learning algorithms in conjunction with machine learning algorithms, Vinayakumar et al. Vinayakumar et al. (2019) uses a deep learning-based framework I-DGA-DC-Net combined with classical machine learning algorithms for AGD detection.

Statistical feature-based detection methods have been effective at first. As the attacker's evasion tactics increase, the attacker disguises the malicious domain as a benign domain by changing its features, such as the character-level distribution features of the domain and the number of domain-mapped IP addresses, thus causing poor robustness of these features. Attackers easily evade statistical feature-based detection methods that rely on features alone, so researchers have proposed to detect domains by using association relationships that describe malicious behaviors between malicious domains, which is the graph structure-based detection method.



(a) Behavioral association of domain query records

(b) Resource association between domain registration

Fig. 1. Behavioral differences between malicious and benign domains.

2.2. Graph structure-based methods

The graph structure-based approach first constructs association rules through the relationships between domains, and then forms a domain graph based on the association rules. By giving a portion of information about known malicious domains, a graph inference algorithm is used to infer the unknown malicious domains in the graph associated with known malicious domains.

Researchers form domain-IP bipartite graphs using mapping relationships between domains and their mapped IP addresses Khalil et al. (2016); Liang et al. (2020); Peng et al. (2019), or domain-host bipartite graphs using query relationships for client hosts accessing domains Lee and Lee (2014); Oprea et al. (2015); Rahbarinia et al. (2015), and then build domain homogeneous graphs containing only one association relationship and one type of node on top of these graphs. MalShoot Peng et al. (2019) builds a domain graph based on the association of domain-mapped IP addresses, and uses graph embedding techniques to embed DNS resolution data of domains for training classification. Oprea et al. Oprea et al. (2015) constructs a domain host bifurcation graph by studying suspicious communications between hosts inside an enterprise network and external domains. Rahbarinia et al. Rahbarinia et al. (2015) proposes Segugio, which constructs a domain-host dichotomous graph based on the association between the client host and its queried domains, and uses machine learning algorithms to train the detection classifier.

Since homogeneous graphs can only represent one type of nodes or edges, which cannot express complex DNS scenarios information. To address this limitation, researchers use heterogeneous graphs to model the various relationships between domains Sun et al. (2019, 2020); Xsa et al. (2020). Sun et al. Sun et al. (2019) uses a heterogeneous information network (HIN) model that represents the relationships between client hosts, domains and IP addresses, and uses a transduction classification method to detect malicious domains in the HIN. Based on Sun et al. Sun et al. (2019), Sun et al. Sun et al. (2020) proposes a heterogeneous graph convolutional network-based method, which combines the graph convolutional networks (GCN) and attention mechanisms to detect malicious domains.

Graph structure-based detection methods are less likely to be circumvented by exploiting the association relationships between domains. However, these detection methods have limitations in detection scope and can only identify malicious domains associated with known malicious domains in the graph. To address the boundaries of the above research methods, HANDOM combines statistical feature-based and graph structure-based methods to achieve high adaptability and high detection coverage. HANDOM builds the heterogeneous graph to represent strong correlation relationships between domains to avoid evasion by attackers, and extracts temporal correlation-based domain features to uncover more malicious domains with the same malicious behavior patterns.

3. Motivation

Attackers use dynamic DNS-based agile techniques such as Fast-Flux and Domain-Flux to hide malicious services' actual locations. As shown in Fig. 1(a), attackers may use DNS techniques to manipulate domains at different times in the Cyber Kill Chain (e.g., reconnaissance, delivery, command and control) for highly stealthy attacks such as sending spams, hiding C&C servers, and transmitting data through DNS covert channels. Highly stealthy malicious domains have the following characteristics: using public platforms such as Content Delivery Networks (CDNs) for domain to IP address mapping to circumvent resource connections; disguising their network communication traffic as legitimate traffic and reducing their activity trajectory to show less frequent activity. In this way, it disguises itself as a benign domain to evade detection systems. Many new malicious domains with short survival cycles are also registered to evade detection. These evasion methods make the detection of highly stealthy malicious domains more difficult.

The main motivation of this paper is that even if an attacker performs evasive means in carrying out malicious activities, there still consist temporal and spatial correlation between the behaviors of malicious domains. This correlation is difficult to circumvent simultaneously by attackers for cost reasons. By distinguishing the behavioral differences between malicious and benign domains in terms of spatial-temporal relevance, we can solve the problems of highly stealthy malicious domain detection under attackers' circumvention means. We analyze the behavioral differences from the following two aspects, as shown in Fig. 1.

Spatial context-based correlation of resources sharing. We found that based on the attacker's behaviors in the Cyber Kill

Chain, the attacker's control host and the infected host group usually have spatial correlation when conducting malicious activities, such as attackers use overlapping infected host groups at different stages of the attack to achieve malicious behaviors, which leads to anomalous correlations between the query records of domains. On the other hand, attackers usually register a large number of domains in the same period in order to save costs as shown in Fig. 1(b), the domains belonging to the same registration batch share the same registration information, which leads to the resources sharing correlation between domains. In contrast, benign domains are accessed by hosts with a more random distribution and more unique registration information.

Temporal context-based correlation of behavioral patterns. Correlation based on temporal contextual information. When conducting malicious activities, there are usually strong temporal correlations between malicious domains manipulated by attackers, as well as similar abnormal behaviors based on time, such as malicious hosts accessing domains with a certain regularity and periodicity, with fixed query time patterns. Each malicious organization has its specificity in domain query patterns, from which extract common features among malicious domains by analyzing the time series of domain queries from multiple organizations. In contrast, benign domains exhibit more randomness in being accessed by hosts without specific query times and behavior patterns.

Therefore, based on the above two distinct behavior differences in spatial-temporal correlations between malicious and benign domains, we propose HANDOM to mine the query patterns of domains and resource association relationships. HANDOM constructs rich heterogeneous graph structures from spatial correlations based on domain query records as well as resource distributions, then mines query patterns of malicious domains, and extracts time series features from temporal correlation-based anomalous behaviors, and then uses the HAN model to embed these two kinds of information.

The reason why we choose HAN model is that HAN can well solve the problem of mutual integration of graph structure information and statistical features. HAN is a novel semi-supervised heterogeneous graph neural network based on hierarchical attention Wang et al. (2019), which can classify nodes for heterogeneous graphs Huang et al. (2020); Long et al. (2020); Zhao et al. (2020). Since C&C domains can easily circumvent existing domain detection methods because they are highly stealthy and can remain dormant for a long time without being detected, and attackers will involve C&C domains in as few resource associations as possible. Using the HAN model not only supports the resource association between domains, but also analyzes the behavior pattern of each domain node, which can well present the malicious information of domains through two dimensions. HAN based on a two-layer attention mechanism can effectively combine global graph structure information as well as local node feature information to learn the maliciousness of each domain in the graph and classify it correctly. The node-level attention-based mechanism in HAN can learn various types of nodes associated with the domain separately, and effectively handle the feature attributes of nodes in the graph; the semantic-level attention-based mechanism can effectively learn the importance of different meta-paths, and thus can deal with multiple types of nodes and multiple association relations in the heterogeneous graph scenarios we constructed in a hierarchical manner.

4. Preliminary

In this section, we first describe the definition of the heterogeneous graph, meta-path, and formulate the malicious domain detection problem based on the HAN model.

Definition 1. (Heterogeneous Graph.) A heterogeneous graph is a graph containing two or more types of objects or links, given a heterogeneous graph G = (V, E), where V is the set of objects in the graph G, E is the set of links, and the node type mapping function ϕ: $V \to Z$ and the link type mapping function Φ: $E \to R$ denote the types of nodes and links, respectively, where Z is the set of object types and R is the set of link types, where $|Z|$ + $|R| >$ 2. In the heterogeneous graph, objects can be connected by different links, which are called meta-paths.

Definition 2: (Meta-Path.) Different meta-paths are used to represent complex association relations between different objects. For example, the meta-path between object Z_1 and Z_i can be represented by the relation $R = R_1 \circ R_2 \circ \dots \circ R_{i-1}$, where $\circ$ denotes the composition operator on relations, which can be described as:

$$Z_1 \xrightarrow{R_1} Z_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{i-1}} Z_i \tag{1}$$

which can also be abbreviated as Z_1 Z_2 ... Z_i.

Definition 3. (Heterogeneous Attention Network). The HAN model is a semi-supervised graph neural network for the heterogeneous graph. The input of the HAN contains the following three types of data: the heterogeneous graph $G = (V, E)$, where V and E are the sets of nodes and links; the meta-path set P which is composed of relations R; the node feature matrix X denotes the feature vector of nodes V, where x_i denotes the set of features of node v_i. HAN uses a two-level attention mechanism structure: node-level attention and semantic-level attention mechanism to learn node embeddings in HAN for specific tasks, such as node classification tasks.

Definition 4. (Malicious Domain Detection Based on HAN Model.) We formulate the malicious domain detection problem as a node classification problem on the HAN model. We apply the following information into the HAN model: the heterogeneous graph G constructed based on the association relations between domains; the set of meta-paths P constructed based on the set of association relations R of domains; the feature matrix X of domains. Given the label of nodes, HAN uses the two-layer attention mechanism to learn the weights of each node and each meta-path, then get the final malicious preference features of each domain node as the semantic-specific embedding Z, Z is used in combination with the cross-entropy loss function for the node of malicious domains classification task.

5. Method design

5.1. Method overview

The method architecture of HANDOM is shown in Fig. 2, which contains three parts: graph construction, feature design and HAN-based malicious domain detection model. By inputting DNS log data and Whois data, HANDOM first constructs a heterogeneous graph and the meta-path set, which can represent the associations between host, domain, and domain registration information in DNS scenarios. HANDOM then extracts time series-based and registration-informed domain features for each domain node in the graph, which is used to mine the behavior patterns of malicious domains. After that, HANDOM inputs both the heterogeneous graph, the meta-path set, and the feature matrix to the HAN model, and generates a domain graph containing only domain nodes based on the meta-path set. Finally, we define a cross-entropy loss function for HAN's semi-supervised classification model, and train the HAN model by giving a portion of the labeled domain nodes in the domain graph. The trained HAN-based model can detect whether the unknown domains in the domain graph are malicious or not.

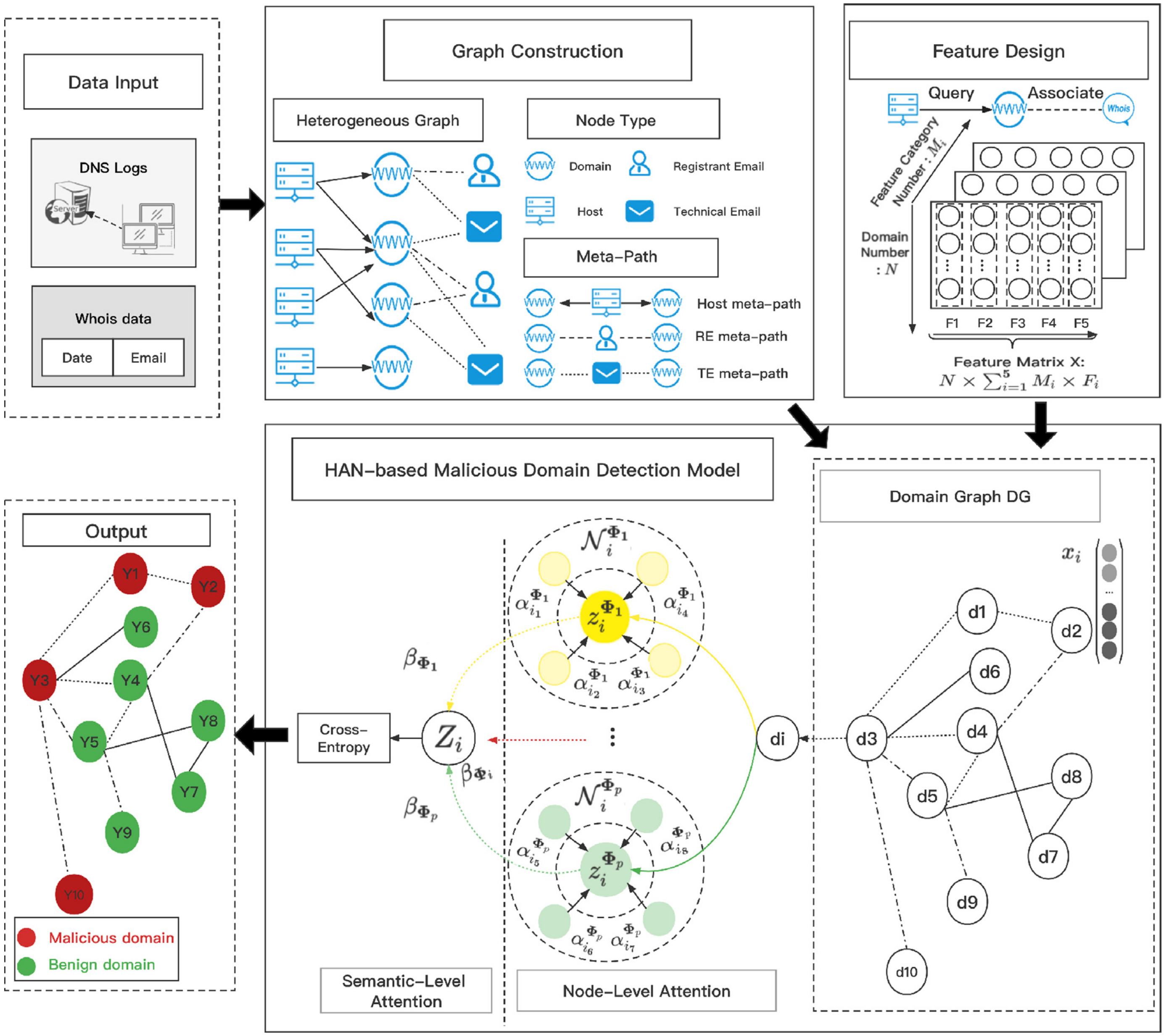


Fig. 2. The architecture of HANDOM.

5.2. Graph construction

Based on the two major spatial contextual association facts that attackers use overlapping clusters of infected hosts at different stages of the attack and malicious domain registration information is cross-used, we construct the following graph association rules: i) An association exists between domains that are queried by the same host within the time window. ii) An association exists between domains associate with the same registrant or technical email within the time window. Then HANDOM uses the graph association rules and following two types of data to construct the heterogeneous graph *G*. The first type of data is DNS log data: the query records of client hosts accessing domains within the time window are extracted from the real network environment, mainly including the information of hosts accessing domains and their query timestamps, which are used to construct query relationships of client hosts accessing domains. The second type of data is Whois data: Whois data includes the entire lifecycle of the domain and registration information, such as domain registration date, domain survival time, registrant email address, and technical email address, which is used to construct resource association relationships between domains.

The constructed heterogeneous graph $G = (V, E)$ contains four types of nodes V and three types of connection relationships E. Nodes include four types: domains, client hosts, registrant emails, and technical emails. Connection relationships $R = (R1, R2, R3)$ contain three kinds of association relations: the query relationship $R1$ between domains and hosts, and the association relationship $R2$ between domains and registrant emails, the association relationship $R3$ between domains and technical emails. We use three matrices: Q_{ij}, AR_{ij}, AT_{ij} to represent connection relationships, and extract the following three symmetric meta-paths (the node types of both the start and endpoints of the meta-path are domains) based on the above three association relationships, including *Host*

Table 1
Description of relationships and meta-paths.

Relationship	Description	Meta-path
$R1$	Q_{ij} represents the query relation between host and domain, if client i queries domain j, Q_{ij} =1, else Q_{ij} =0	*Host* meta-path: domain $\xrightarrow{Q}$ *Host* $\xrightarrow{Q^T}$ domain
$R2$	AR_{ij} represents the association relation between registrant email and domain, if the registrant email i is associated with domain j, AR_{ij} =1, else AR_{ij} =0	Registrant Email(RE) meta-path: domain $\xrightarrow{AR}$ RE $\xrightarrow{AR^T}$ domain
$R3$	AT_{ij} represents the association relation between technical email and domain, if technical email i is associated with domain j, AT_{ij} =1, else AT_{ij} =0	Technical Email(TE) meta-path: domain $\xrightarrow{AT}$ TE $\xrightarrow{AT^T}$ domain

meta-path based on $R1$, Registrant Email (RE) meta-path based on $R2$, and Technical Email (TE) meta-path based on $R3$, as shown in Table 1.

DNS log data obtained directly from the real network contains a large amount of dirty data, as there are large proxy servers querying domains resulting in lots of traffic records, and clients querying misspelled domain records, etc. These traffic are ineffective for malicious domain detection and consume a lot of computing resources Rahbarinia et al. (2015); Sun et al. (2019).

We prune the constructed graphs by filtering rules to remove redundant data and provide high-quality data support for subsequent detection: **Inactive clients and email addresses.** Clients with queries less than K_f (we set K_f=2) domains and email addresses that map to only one domain are considered as inactive, and we remove them because we focus on domains with actual query behaviors, not misoperations due to syntax errors or query errors. **Public email addresses.** We define email addresses with specific company names such as Microsoft, Google, and Alibaba as public email addresses, such as domains@microsoft.com. We remove email addresses whose registrant email addresses or technical email addresses are public information. Public email addresses not only map a large number of malicious domains, but are also used by benign domain registrants, so this type of email address nodes incur significant resource consumption and do not contain substantial malicious association information. **Single email addresses.** Email addresses that map to only one domain are discarded as they do not contribute to label propagation. We follow the above rules for graph construction and keep all domains with malicious information and associated nodes.

5.3. Feature design

By analyzing the DNS logs, we found those client hosts used by attackers are divided into different divisions of labor, such as scanning, random access, obtaining the address of the Command and Control communication server, etc. Hosts infected by the same attacker tend to query the exact malicious domains, and the query records with similar regularity in time. Since the interaction behaviors of attackers in the Cyber Kill Chain are time-related, we perform a time series analysis of the collection of DNS queries. There are resource associations in the malicious domains registered by the attackers, so we also perform a domain registration analysis of Whois data.

We tracked DNS logs and Whois data within time window periods, statistically analyzed domains' behavior patterns, then extracted time-series features and domain registration features from the following five facts to indicate domain's malicious behaviors.

As shown in Table 2, where the feature categories of F1-F4 are time series features, the feature category of F5 is domain name registration features. We provide a brief description of these feature names in the table and explain in detail how to calculate these feature values in the text. Time series analysis focuses on the interdependence of time-based data series. The mean, peak, variance, and standard can be used as discrete measures for statistical analysis of time series.

Table 2
Feature Description.

Data Source	Feature Category	Sub - Category
DNS Logs	F1	a : Total number of hosts. b : Discrete indicators of $T1$, mean, peak,.
	F2	a : Total number of network segments. b : Discrete indicators of $T2$, mean, peak,.
	F3	a : Mean and peak and discrete indicators of $T3$. b : Maximum and minimum value of $T3$. c : Difference value of maximum and minimum of $T3$.
	F4	a : Discrete indicators of $T4$ under hour time units. b : Maximum counts divide total counts of $T4$.
Whois Data	F5	a : Survival cycle length of per domain. b : Longest and shortest survival period of per domain. c : Number of email addresses. d : Character similarity of each nameserver.

The number of hosts visiting the domain. Based on the principle that malicious domains are usually visited by more malicious hosts, we extract the feature category: F1. We extract the list of hosts visiting the domain within the time window, calculate the number of each host visiting the domain under each time interval, and extract the following features: the total number of hosts visiting the domain (F1-a); the number distribution sequence $T1$ formed by different hosts visiting the domain based on time, from which extracts the discrete indicators of $T1$ (F1-b), such as the mean, variance, peak, to reflect the trend of the number of hosts visiting the domain over time. These features indicate the regularity of the distribution of hosts accessing malicious domains, and are used to detect malicious domains accessed by a large number of infected hosts and malicious domains with abnormal host access frequency distribution.

Resource diversity of hosts. Based on the fact that the same attacker usually attacks neighboring client hosts, and a large number of infected hosts typically belong to the same network segments, we extract the segment feature category: F2. Within the time window, we extract the list of different network segments to which the hosts visit the domain at each time interval, which is called the segment diversity sequence $T2$. Calculate the following features from $T2$: the number of different network segments accessing the domain(F2-a); the mean, variance, peak, and other discrete indicators of $T2$ (F2-b). These type of features can obtain the diversity of host resources accessing the domain to detect malicious domains with concentrated resource distribution.

Domain query time change rate. When conducting malicious activities, sudden cut-in and cut-out operations by attackers can lead to a significant increase or decrease in the number of domain queries, so we extract the feature category F3. Within the time window, we extract the time series $T3$ of domain queries difference value, statistically analyze the domain query time change rate in daily and second units, and extract the following features: statistical values such as variance, and peak in domain queries difference value list $T3$ (F3-a); the maximum and minimum query time difference (F3-b); and the difference between the maximum and minimum query time difference (F3-c), to detect the malicious domains with unstable queries and surprise malicious domains with unstable and burst features.

Domain query time period preference. Based on the fact that attackers usually have their attack preferences, if attackers tend to launch attacks at night, malicious domains will have frequent query records at night to extract feature category F4. Extract the time series list $T4$ of the number of times the domain is queried within the time window, calculate the number of queries in each time interval, and extract the following features: mean, variance and peak value in the query count sequence list $T4$ under hour time unit (F4-a); the ratio of maximum access counts to the total access counts under the hour time unit (F4-b). The set of malicious domains with similar query time preferences is detected by the feature that malicious domains have similar query time preferences among them.

Domain registration feature. The full lifecycle of a domain refers to the whole process of domain registration, update and cancellation, which can happen many times. Malicious domains usually have frequent registration, cancellation and other survival update operations, with a short survival period, thus extracting the feature category F5. We calculate the average survival cycle length of the domain (F5-a), which is calculated by dividing the sum of all survival cycles of the domain by the number of cycles; the longest and shortest survival period of the domain (F5-b); the number of domain email addresses (F5-c); the character similarity between domain nameservers (F5-d). The calculation of feature F5-d uses the Levenstein distance, which is a type of edit distance and calculates the minimum number of edit operations required to convert from one string a to another string b. Allowed edit operations include replacing one character with another, inserting a character or deleting a character. Given two strings a and b, the formula for calculating the Levenstein distance between a and b is as follows:

$$\begin{aligned}&\mathrm{lev}_{a,b}(i,j)\\&=\begin{cases}\max(i,j) & \text{if } \min(i,j)=0\\ \min\begin{cases}\mathrm{lev}_{a,b}(i-1,j)+1\\ \mathrm{lev}_{a,b}(i,j-1)+1\\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i\neq b_j)}\end{cases} & \text{otherwise.}\end{cases}\end{aligned} \tag{2}$$

where i denotes the first i characters of a and j denotes the first j characters of b. F5-d is calculated and averaged for each nameserver of the domain in turn according to the Levinstein distance. This is used to detect malicious domains with frequent survival updates and short lifecycles, and abnormal registration information.

In this study, a month is used as the time window, and days and periods are set as time intervals respectively. The whole day is divided into the following two types of time periods: (a) $[0-4], (4-24]$; (b) $[0-12], (12-24]$. The four categories of F1-F4 feature values under each day and each time period within one month are extracted.

We extract features x_i for each domain node in the graph G, the number of features for each domain node is $\sum_{i=1}^{5} F_i * M_i$, where F_i denotes the feature category and M_i denotes the number of sub-features under each category. Feature matrix X represents the features of all domain nodes in the graph and contains N rows and $\sum_{i=1}^{5} F_i * M_i$ columns, the rows represent the number of domain nodes and the columns represent the number of features of each domain node. We embed feature matrix X into the HAN model.

5.4. HAN-based malicious domain detection model

To effectively handle the complex statistical-and-structural information of domain nodes, we formulate the malicious domain detection problem as a node classification problem on the HAN model. In the HAN model, we only focus on domain nodes. To better represent our HAN-based domain detection problem, we extract the domain graph DG containing only domain nodes from the heterogeneous graph G based on the set P of the three symmetric meta-paths mentioned above. The DG graph is only intended to explain the node classification problem based on the domain graph and is not involved in the computation of the actual HAN model. We take the nodes V and edges E in the heterogeneous graph G, the set of meta-paths $P = \{Host, RE, TE\}$ and the feature matrix X as inputs to the model HAN. Given the label information of $k\%$ of the domain nodes in the DG graph and input them as domain training samples y_L into the HAN model. The purpose of the HAN model is to learn the embedding of y_L and predict the labels of the unlabeled domain nodes in the DG graph. The overall process of the HAN-based malicious domain detection model is shown in Algorithm 1.

Algorithm 1: The overall process of HAN-based malicious domain detection model.

```
input  : Heterogeneous graph $G$=($V, E$),
         Meta-path set $P = \{Host, RE, TE\}$,
         feature matrix $X$,
         Domain training samples $y_L$.
output: HAN-based malicious domain detection model.

for  $\Phi \in \{Host, RE, TE\}$ do
    for $i \in V$ do
        $\mathcal{N}_i^{\Phi} \leftarrow DomainNeighborNodes(G, \Phi)$for $j \in \mathcal{N}_i^{\Phi}$ do
            Weight coefficient $a_{ij}^{\Phi}$:
            $a_{ij}^{\Phi} \leftarrow \mathrm{softmax_j}\left(\mathrm{att}_{node}\left(x_i, x_j, \Phi\right)\right)$
            $\leftarrow \frac{\exp\left(\sigma\left(\mathrm{a}_{\Phi}^{\mathrm{T}}\cdot\left[\mathrm{x}_i \| \mathrm{x}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i^{\Phi}}\exp\left(\sigma\left(\mathrm{a}_{\Phi}^{\mathrm{T}}\cdot\left[\mathrm{x}_i \| \mathrm{x}_k\right]\right)\right)}$
        Node-level embedding $z_{\mathrm{i}}^{\Phi}$:$z_i^{\Phi} \leftarrow \sigma\left(\sum_{j\in\mathcal{N}_i^{\phi}} a_{ij}^{\Phi}\cdot x_j\right)$
    Weight $\beta_{\Phi}$:$\beta_{\Phi} \leftarrow \mathrm{att_{sem}}\left(Z_{\Phi_{Host}}, Z_{\Phi_{RE}}, Z_{\Phi_{\mathrm{TE}}}\right)$
    $\omega_{\Phi} \leftarrow \frac{1}{|V|}\sum_{i\in V} q^T\cdot\tanh\left(W\cdot Z_i^{\Phi}+b\right)$
    $\beta_{\Phi} \leftarrow \frac{\exp\left(\omega_{\Phi_i}\right)}{\sum_{i=1}^{P}\exp\left(w_{\Phi_i}\right)}$
    Fuse the semantic-level embedding: $\mathrm{Z} \leftarrow \sum_{i=1}^{P}\beta_{\Phi_i}\cdot Z_{\Phi_i}$
Train HAN by the Cross-Entropy loss function:
$L \leftarrow -\sum_{l\in y_L} Y^l \ln\left(C\cdot Z^l\right)$
Return the HAN-based malicious domain detection model.
```

In Algorithm 1, HAN uses two attention mechanisms to present domain behavior differences: i) Node-level attention mechanism. HAN first uses the node-level attention mechanism for each node to learn the importance of its related neighboring nodes based on each meta-path and assigns different weight values to them. For example, based on the $Host$ meta-path, for domain nodes connected to more host nodes associated with malicious domains, HAN gives a higher weight value to that domain than domain nodes associated with hosts of benign domains. ii) Semantic-level attention mechanism. HAN uses the semantic-level attention mechanism to learn the weight of each meta-path. For example, if the meta-path of the *Host* is more meaningful for detecting malicious domains than the meta-path of *RE* or *TE*, then HAN assigns a higher weight value to the *Host* meta-path.

Node-level attention mechanism. HAN first finds the neighboring nodes $\mathcal{N}_i^{\Phi}$ of each domain node i based on meta-path Φ. Then it uses $att_{node}\left(x_i, x_j, \Phi\right)$, which is a node-level attention neural network, to calculate the importance between each neighboring node j to node i, and finally uses the softmax function to obtain the weight coefficient $a_{ij}^{\Phi} = \frac{\exp\left(\sigma\left(\mathbf{a}_{\Phi}^{T} \cdot [\mathbf{x}_i \| \mathbf{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp\left(\sigma\left(\mathbf{a}_{\Phi}^{T} \cdot [\mathbf{x}_i \| \mathbf{x}_k]\right)\right)}$ of node pair(i, j), where $\mathbf{a}_{\Phi}$ is the node-level attention vector. Aggregate the weight coefficients of all neighbor nodes of node i, and get the domain node-level embedding $z_{\mathrm{i}}^{\Phi} = \sigma\left(\sum_{j \in \mathcal{N}_i^{\phi}} a_{ij}^{\Phi} \cdot x_j\right)$.

Semantic-level attention mechanism. For a given set of meta-path $P = \{Host, RE, TE\}$, after the node-level attention mechanism, HAN gets three sets of embedding under each meta-path: $Z_{\Phi_{Host}}, Z_{\Phi_{RE}}, Z_{\Phi_{\mathrm{TE}}}$. After getting the semantic-layer embedding set of meta-paths, HAN uses the semantic-level attention neural network $att_{sem}\left(Z_{\Phi_{Host}}, Z_{\Phi_{RE}}, Z_{\Phi_{\mathrm{TE}}}\right)$ to learn the weight coefficient β_{Φ} of each meta-path. HAN first uses a Non-linear transformation to transform the semantic embeddings to derive the important value of each meta-path: $\omega_{\Phi} = \frac{1}{|V|} \sum_{i \in V} q^T \cdot \tanh\left(W \cdot Z_i^{\Phi} + b\right)$, where V is the set of domain nodes, W is the weight matrix, b is the bias vector, and q is the semantic-level attention vector. Then HAN gets the weight coefficient by using the softmax function $\beta_{\Phi} = \frac{\exp\left(\omega_{\Phi_i}\right)}{\sum_{i=1}^{P} \exp\left(w_{\Phi_i}\right)}$. Then HAN integrates the two levels of attention to embed the final malicious preference features by aggregating all the node-level weights as well as the semantic-level weights to obtain the semantic-level embedding as the semantic-level embedding $\mathrm{Z} = \sum_{i=1}^{P} \beta_{\Phi_i} \cdot Z_{\Phi_i}$.

Finally, the detection model is trained by learning the final embedding of the labeled nodes set y_L and using the cross-entropy loss function to minimize the cross-entropy between the label and prediction value of y_L: $L = -sum_{l \in y_L} . Y^l \ln\left(C \cdot Z^l\right)$, where C denotes the parameters of the classifier and Z^l denotes the embedding of the labeled nodes, and Y^l denotes the label of the labeled nodes. We can use the trained detection model to predict the labels y_U of unlabeled nodes of the graph.

6. Experiments

In this section, we first evaluate the performance of HANDOM with different proportions of initial training labeled samples, and analyze the importance of each meta-path in the graph. Then we analyze the performance of HANDOM with machine learning methods. Finally, we demonstrate the superiority of our method by comparing HANDOM with other malicious domain detection methods.

6.1. Experimental settings

Datasets. In this paper, we use DNS log data within the time window as well as Whois data of domains to evaluate our proposed method. Our DNS log dataset comes from Qi An Xin Technology Group Inc's open-source dataset (https://datacon.qianxin.com/opendata), which contains large real network traffic captured on Qi An Xin's DNS servers by deploying sensors. Each record of DNS logs represents the timestamp of the host accessing domains and the number of times, which provides detailed information about the hosts in terms of domain requests, and the domains and hosts in DNS logs are encrypted. Whois data related to domains represent the registration information of the domain, such as the registrant, the registration email, registration time, etc. This information can be obtained from the Whois domain name registration database and top-level domain name zone files.

Table 3
Performance comparison of different label proportions.

Label Proportion	Precision(%)	Recall(%)	F-Score(%)	Accuracy(%)
90%	94.59	97.22	95.89	99.55
70%	94.12	95.14	94.62	99.25
50%	90.78	94.81	92.75	99.00
30%	90.11	94.89	92.44	99.02
10%	90.75	93.55	92.13	98.98

To verify that HANDOM can detect highly stealthy malicious domains, the malicious domains in the dataset are all APT domains, which are used by 9 Advanced Persistent Threat (APT) organizations and can be considered as 9 malicious family domains. These domains are manually labeled by researchers based on blacklists and whitelists and threat intelligence databases. Due to privacy protection, these domains cannot be disclosed in this paper for illustrative purposes only.

We set the time window of one month to extract a one-month of DNS logs and Whois information, and generate the heterogeneous graph and nodes features according to our method. Then, we randomly slice the nodes in the graph, where k% of the nodes in the graph are randomly selected as training nodes, and the remaining nodes are used as test nodes to validate the performance of the HAN model generated by the training.

We use the metrics as shown in the following formulas to evaluate our method performance. TN indicates the number of benign domains correctly classified; FP indicates the number of benign domains classified as malicious; FN indicates the number of malicious domains classified as benign; TP indicates the number of malicious domains correctly classified.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

In addition to these conventional evaluation metrics, due to the existence of data imbalance in the dataset, we additionally use the classification accuracy metric Matthews Correlation Coefficient (MCC) to evaluate the classification accuracy of each method under imbalanced data.

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}} \tag{7}$$

6.2. Performance evaluation results of HANDOM

We randomly select k% (where k is equal to 10, 30, 50, 70, 90) of the domain nodes on the HAN model as initial training samples to train the model. HAN still needs a validation set to train the parameters of the model, so we then select m% (where m is approximately equal to 10) of the dataset as the validation set, and the remaining nodes in the graph as the test set to detect the model performance.

As shown in Table 3, when training samples reach 90% in the graph, the metric values of HANDOM are basically above 95%, with precision, recall, F-Score, and accuracy of 94.59%, 97.22%, 95.89%,

Table 4
Data distribution description.

Category	Train		Test		
Label Proportion	Benign	Malware	Benign	Malware	TN - FP - FN - TP (Result)
90%	11,909	809	634	36	632 - 2 - 1 - 35
70%	10,050	660	2493	185	2480 - 13 - 3 - 182
50%	8799	572	3744	273	3722 - 22 - 13 - 260
30%	5019	336	7624	509	7571 - 53 - 26 - 483
10%	2514	163	10,029	682	9964 - 65 - 44 - 638

and 99.55% respectively. As the number of training samples decreases, there is a small decrease in each evaluation metric value of HANDOM. When HANDOM is trained with only a small number of training label samples (about 10% of training samples in the graph), it also has a good performance with recall, F-Score and accuracy above 90% respectively.

Table 4 shows the distribution of benign and malicious domains in the training and testing samples in the dataset, and the actual classification results of TN, FP, FN, and TP. To highlight the difference in the number of nodes in the training and test sets, we count the number of nodes in the validation sets into the training sets. The number of benign domains is much larger than malicious domains in both training set and test sets, and the detection range of malicious domains by the malicious information in the graph shrinks as the training samples gradually decrease.

When there are only 163 malicious domains in the training sample, HANDOM can still detect the remaining 638 unknown malicious nodes in the graph. This indicates that the extracted features of domain nodes can extend the detection range, which can detect unknown domains with same malicious patterns. When the proportion of training samples is large enough to reach 90%, HANDOM can achieve lower false positives and false negatives through tighter correlation between malicious domains, with only 2 false positives (we call them NO_benign1 and NO_benign2) and 1 false negative (we call it NO_malware).

We further analyze the reasons for these misclassifications and explain why there are highly hidden malicious domains that remain undetected despite having sufficient training samples. The NO_malware domain belongs to a small-scale APT family that contains few malicious domains, and the resources of this family are distributed more scattered, with a mixture of resources associated with benign domains. The access records of NO_malware are more random, with no fixed time or regularity, so HANDOM and other machine learning methods treat it as a benign domain. NO_benign1 and NO_benign2 are mostly accessed by malicious hosts, then their association relationships are biased towards malicious associations; as the host resources are concentrated in the same network segments, the access time is fixed time periods, thus their feature values are biased toward malicious values, so they are judged as malicious domains by HANDOM. Therefore, it is possible to circumvent HANDOM when the malicious domain circumvents both the temporal-based resource associations and malicious query patterns.

Considering the cost of tagging domains in practical applications and detection efficiency, we set the training label proportion of 50% in an attempt to simulate the detection results in a real network environment. We analyze the importance of each meta-path in the heterogeneous graph.

Table 5
Performances comparison of HANDOM under each meta-path.

Meta-Path	Precision(%)	Recall(%)	F-Score(%)	Accuracy(%)
Host	90.23	87.91	89.05	98.53
RE	95.14	87.04	90.91	98.83
TE	94.76	87.04	90.73	98.81
RE+TE	94.96	89.74	92.28	98.98
Host+RE+TE	92.20	95.24	93.69	99.13

From Table 5, the results show that each HANDOM constructed based on *RE* or *TE* has higher precision and relatively lower recall compared to the HANDOM constructed based on meta-path *Host*. This is because there is a broader association between hosts and domains, and thus are prone to misclassify some benign domains accessed by infected hosts, leading to false alarm rates. Email addresses are more obscure, which cannot detect more relevant malicious domains, but they lead to fewer false positives. HANDOM constructed based on combined *RE* and *TE* meta-paths has higher metric values than these three separate meta-path, because more malicious resources get aggregated. When these three meta-paths are combined, we obtain the highest performance of HANDOM, with precision, recall, F-Score, and accuracy of 92.20%, 95.24%, 93.69%, and 99.13% respectively.

6.3. Comparison with machine learning methods

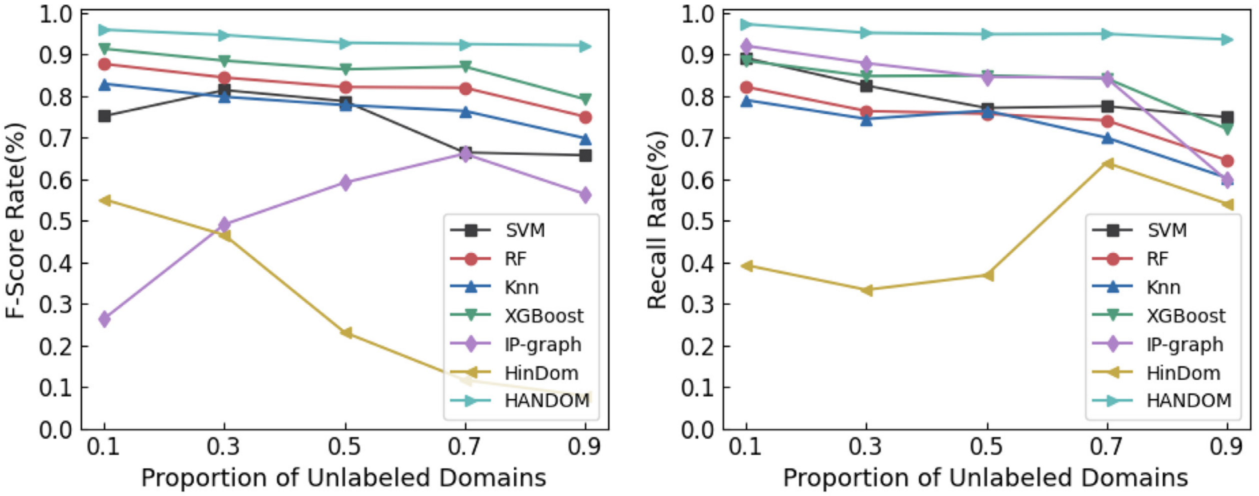
To better interpret the advantages of our proposed HANDOM, we compare our approach with traditional machine learning methods (SVM, RF, KNN, XGBoost). We apply the statistical features used in HANDOM to machine learning methods, and select the number of training samples randomly in proportion. As for the evaluation metrics, since the domain nodes of HANDOM are inherently imbalanced, with significantly more benign domains compared to malicious domains (about 90% of benign domains), any metric used to describe performance needs to take into account this imbalance. Therefore, Table 6 focuses on the selection of the recall metric that can indicate the correct classification of malicious domains and the F-Score metric that serves as a reconciled average of recall and precision. The detection results for each method under different proportions are presented in Table 6 and Fig. 3.

From Table 6, with different proportions of training samples, the detection performance of HANDOM always outperforms SVM, RF, KNN, XGBoost. In the beginning, HANDOMs F-Score and Recall are much higher than other methods at the proportion of 90% of the training label instances (F-Score: 95.89%, Recall: 97.22%), and XGBoost achieves the best performing method of machine learning with F-Score: 91.30%, Recall: 88.42%. The performance of SVM, RF, KNN and XGBoost gradually decrease as the labeled instances proportion decreases, with detection performance all dropping below 80% with only 10% of training samples, while the performance of HANDOM remains impressive (F-Score: 92.13%, Recall: 93.55%). Fig. 3 visualizes the results, the metric values of HANDOM are consistently higher than other machine learning methods. As the proportion of unlabeled training domains increases, the line graphs of HANDOM's F-Score and Recall values have minimal downward trend fluctuations, while the remaining methods all have a larger downward trend in performance values, which proves the robustness of HANDOM. Thus, these results verify the proficiency of HANDOM.

As we can see from Fig. 4, the four performance metrics (precision, recall, F-Score, accuracy) of HANDOM are all higher than SVM,

Table 6
Performances comparison of different machine learning methods.

Method	SVM		RF		KNN		XGBoost		HANDOM	
Proportion	F-Score	Recall	F-Score	Recall	F-Score	Recall	F-Score	Recall	F-Score	Recall
90%	75.14	89.04	87.64	82.11	82.87	78.95	91.30	88.42	**95.89**	**97.22**
70%	81.36	82.44	84.39	76.34	79.75	74.43	88.45	84.73	**94.62**	**95.14**
50%	78.72	77.08	82.10	75.68	77.78	76.43	86.36	84.86	**92.75**	**94.81**
30%	66.37	77.49	81.94	74.06	76.36	69.93	87.03	84.19	**92.44**	**94.89**
10%	65.70	74.87	75.02	64.55	69.83	60.32	79.22	72.09	**92.13**	**93.55**



(a) The F-Score of HANDOM and different methods (b) The Recall of HANDOM and different methods

Fig. 3. Comparison of F-Score and Recall values of different methods.

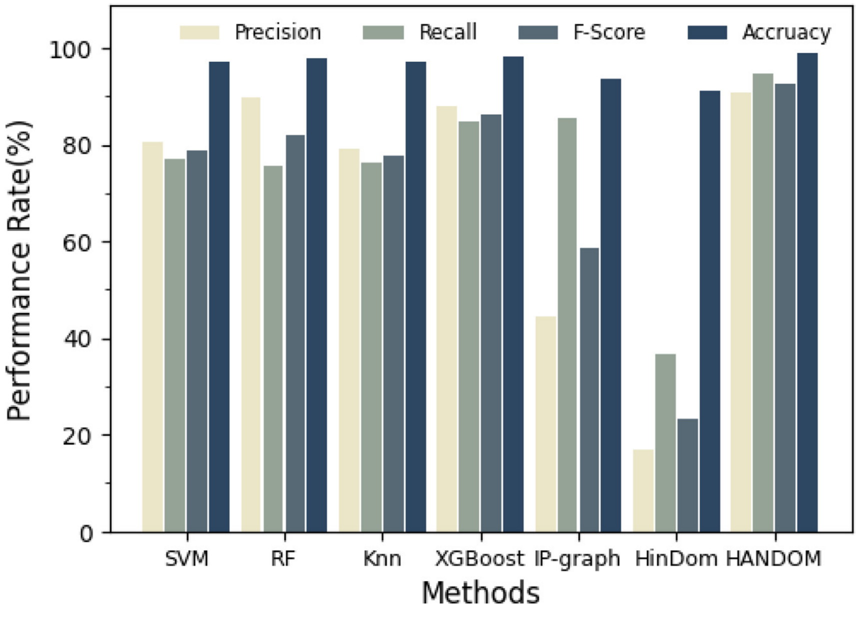


Fig. 4. Performance evaluation on different methods.

RF, KNN, and XGBoost method, when the proportion of training samples is 50%. This gives a reference to the detection effectiveness of HANDOM and these machine learning methods in practical applications. Further analysis reveals that the number of benign domains is much more than malicious domains when the training sample is only 50%, and HANDOM's precision and recall are much higher than other detection methods, which proves that HANDOM is better at handling imbalanced datasets compared to other methods.

From Fig. 5, HANDOM shows the best MCC rates among all detection methods under different proportions of training samples, RF and XGBoost have the second highest MCC effect, and SVM has the worst effect. This proves that HANDOM can handle the data imbalance problem very well. While the machine learning-based

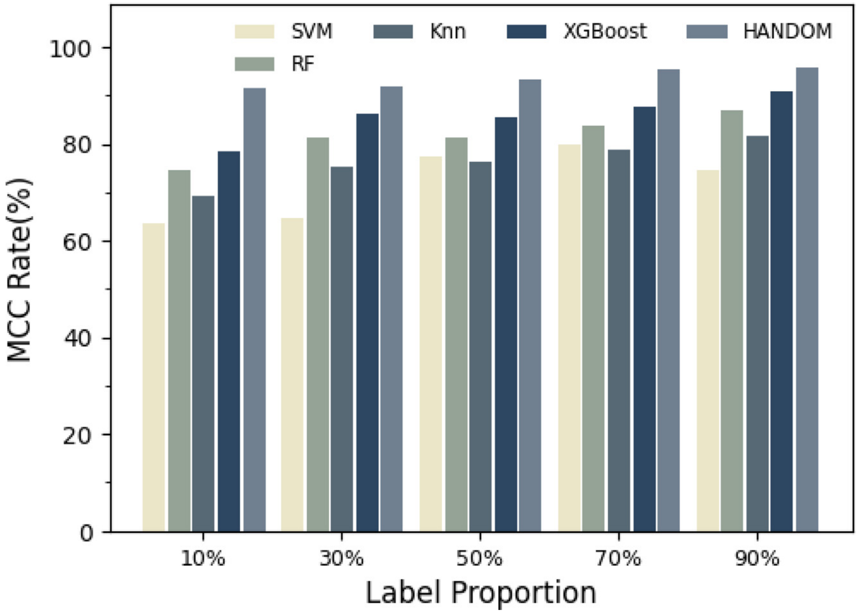


Fig. 5. MCC Rate of different methods.

detection methods are unable to solve the data imbalance problem due to their settings, which leads to their slightly worse MCC rates.

In general, our proposed HANDOM outperforms all the above machine learning methods in malicious domain detection. The advantages of HANDOM can be attributed to the following reasons. HANDOM considers both global correlation relations between domains and local node features. It also relies on resource correlation between domains compared to traditional machine learning methods that use only statistical features, the correlations based on graph structure make it impossible to avoid detection between malicious domains with associations; HANDOM based on graph structure can solve the data imbalance problem well and detect unknown nodes in a graph with high performance with only a small number of labeled samples, thus can be better applied to detection in real network environments.

6.4. Comparison with other malicious domain detection methods

To verify the superiority of our method, we enlarge the comparison scope to compare HANDOM with other malicious domain detection methods Khalil et al. (2016); Sun et al. (2019). Issa Khalil et al. Khalil et al. (2016) proposes an IP-graph method that constructs a domain graph based on the association between domains and their resolved IP addresses, and then calculates the association weight between each unknown malicious domain and the given malicious domains, which is determined by the IP address and IP's AS number, and finally obtains the malicious score for each unknown malicious domain. Sun et al. Sun et al. (2019) proposes a HinDom method that proposes six different meta-paths based on the association between domains, and uses transduction classification to detect unknown domains in the graph. HANDOM and the two compared methods Khalil et al. (2016); Sun et al. (2019) are both graph-based detection methods, which first construct graph models according to association rules, and then apply inference algorithms on the graph for malicious domain node detection. The difference is that HANDOM not only uses the graph model, but also extracts five types of features for domain nodes based on it, which greatly enriches the information of each domain node in the graph. However, these two types of methods Khalil et al. (2016); Sun et al. (2019) do not perform feature extraction on the nodes in the graph.

Table 7
Distribution of nodes of the graph constructed by each method.

Method	Domain Nodes	Benign Nodes	Malicious Nodes
Dataset	18,765	17,920	845
IP-graph	2445	2000	245
HinDom	15,070	14,530	540
HANDOM	13,388	12,543	845

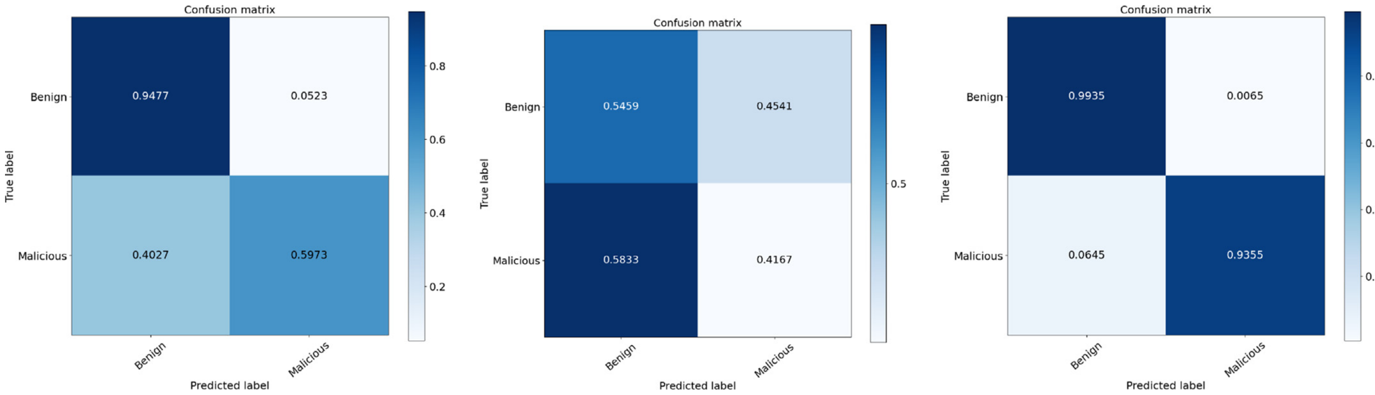
Since the source codes and the datasets are not provided for the above two types of methods, we reproduce them using our datasets according to the information provided in the paper and compare them with our method. We construct each domain graph according to the graph composition rules of each method, and the distribution of nodes in each graph is shown in Table 7. As can be seen from Table 7, the total number of domains in the initial experiment dataset is 18765, of which 17,920 are benign domains and 845 are malicious domains. Compared with the domain graphs constructed by the other two methods, HANDOM's heterogeneous graph covers all the malicious domains and removes the most benign domain nodes. This indicates that HANDOM's graph composition rules have better malicious domain coverage, as well as the ability to obtain more streamlined computational efficiency.

We randomly select k% of domain nodes in the graph as the training samples to train these methods, and the experimental results are shown in Table 8 and Fig. 3. In conjunction with the above analysis, Table 8 selects three metrics, F-Score, recall, and MCC to evaluate the performance of these three methods together. The F-Score, recall, and MCC values of HANDOM at different ratios are higher than IP-Graph and HinDom. For example, when the label proportion is 90%, HANDOM obtains F-Score, recall, and MCC values of 95.89%, 97.22%, and 95.66%, respectively, which are much higher than the results of the two detection methods.

As shown in Fig. 3, the F-Scores of both IP-graph and HinDom are lower than HANDOM and machine learning methods using the statistical features of HANDOM. Recall values of HinDom are lower than machine learning methods, and recall values of IP-graph maintain good performance, but when the training samples are small, the recall value of IP-graph is lower than machine learn-

Table 8
Performance comparison of different malicious detection methods.

Method	IP-graph			HinDom			HANDOM		
Proportion	F-Score	Recall	MCC	F-Score	Recall	MCC	F-Score	Recall	MCC
90%	26.44	92.00	36.38	55.06	39.26	61.07	**95.89**	**97.22**	**95.66**
70%	49.06	87.84	52.52	46.51	33.33	52.13	**94.62**	**95.14**	**95.51**
50%	59.09	84.55	59.25	23.13	36.85	21.93	**92.75**	**94.81**	**93.24**
30%	66.06	84.30	64.62	11.66	63.89	11.15	**92.44**	**94.89**	**91.95**
10%	56.41	59.73	51.86	7.86	54.07	4.82	**92.13**	**93.55**	**91.60**



(a) IP-graph (b) HinDom (c) HANDOM

Fig. 6. Confusion matrix of different methods under the proportion of 10%.

ing methods and HANDOM. These results demonstrate the ability of our selected statistical features to mine malicious behavior patterns, as it can detect more highly concealed malicious domains.

Fig. 4 proves that our HANDOM takes advantage of the two detection methods, on the whole with respect to the precision, recall, F-Score, and accuracy, when the training label proportion of 50%. And it can be seen that the performance of these two methods is slightly lower than the other four machine learning methods in each metric, which also proves that our selected features have a higher ability to detect malicious domains than these two methods.

Fig. 6 shows the confusion matrixs for each method when the proportion of training samples is 10%. These confusion matrix visualize the misclassification cases of benign and malicious domains for each method. In the extreme case (when the training sample is small), HANDOM has the least misclassification cases among the three methods, which indicates that HANDOM has a better ability to reduce false positives and positive positives even in the case of insufficient label data.

We analyze the reasons for the poor experimental results of these two types of methods and the excellent performance of our method and why HANDOM can detect more highly hidden malicious domains. i) The graph constructed by the IP-graph method has a low node coverage, and can only associate 13% of the domain nodes in the dataset. When the training samples are sufficient, IP-graph can detect more malicious domains, but the number of misreported malicious domains is higher, which leads to a low F-Score value. The ratio of the number of benign and malicious domains among the nodes associated with IP-graph is not much different, but the MCC value is low, which indicates that it cannot handle imbalanced data. ii) HinDom has the highest domain node coverage but captures fewer malicious domain nodes, which leads to an unbalanced distribution of domain nodes in the graph, resulting in extremely low MCC values. HinDom's graph has more associations of benign domain nodes and fewer malicious associations exist, and HinDom relies only on the correlation between domains, resulting in the inability to effectively capture more potentially highly hidden malicious domains in the graph which leads to its lowest performance.

Both above types of detection methods use association relationships between domains for detection. Unlikely, HANDOM uses both structural association information between domains and incorporates statistical features. HANDOM depends not only on the structural information of the graph but also on the statistical features extracted from malicious behaviors to enhance the malicious information of domains. As seen in Fig.reffig:random-ml-mask5, the detection performance of the machine learning methods that rely only on our selected statistical features are higher than these two detection methods that rely on graph structures. And the rules of constructing the domain graph of HANDOM enable it to capture more information about malicious domains and remove other redundant nodes to reduce resource computation. The combination of graph structure information and node statistical features of HANDOM not only expands the detection range but also improves the detection accuracy, thus achieving significant detection of malicious domains.

7. Limitations and future work

HANDOM has a limitation that it does not provide real-time detection. Unlike DGA domains, which are short effective and need to be blocked as soon as possible. HANDOM mainly targets long-acting malicious domains such as APT domains, which are long effective and highly covert. HANDOM needs a suitable time window to collect domain behavior data, to achieve the balance between detection efficiency and detection performance.

We provide an analysis of how to circumvent HANDOM. Two types of conditions need to be satisfied for an attacker to circumvent HANDOM: there is no malicious association between the attacker's malicious domains, which means malicious resources cannot be reused; the behavioral habits between malicious domains need to be disguised as characteristics of benign domains. Both of these involve cost issues, and thus circumventing HANDOM requires a greater loss of resources for the attacker.

In the future work, based on the malicious domains identified by HANDOM, we can go further to analyze the size of the malicious family to which each malicious domain belongs, as well as the attributes and scale of malicious activities, to discover the attacks faster and achieve a comprehensive blocking of malicious activities at the early stage of their development.

8. Conclusion

In this paper, we propose an effective malicious domain detection method called HANDOM. More specifically, HANDOM first constructs a heterogeneous graph to represent resource sharing relationships between domains, and then extracts time-series features based on behavioral patterns for each domain in the graph. To better combine statistical features and structural information, HANDOM uses the HAN model to handle both types of information. HAN uses the node-level and semantic-level to learn the malicious embedding of each domain and achieve the effective classification of domain nodes. We test HANDOM in the real network data and experimental results show that our proposed HANDOM has superior performance. In addition, comparative results on multiple scopes show that HANDOM outperforms traditional machine learning models as well as the existing detection methods, and can achieve effective detection of highly hidden malicious domains.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This research is partially supported by National Key Research and Development Program of China (NO.2021YFF0307203, No.2019QY1303, No.2019QY1302), and the NSFC No.61902376, and the Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDC02040100), and the National Engineering Research Center of Classified Protection and Safeguard Technology for Cybersecurity under Grant C21640-3, and the NIM R&D Project under Grant 35-AKYZD2008-3. This work is also supported by the Program of Key Laboratory of Network Assessment Technology, the Chinese Academy of Sciences; Program of Beijing Key Laboratory of Network Security and Protection Technology.

References

Abley, J., 2014. Dns security introduction and requirements.

Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Dagon, D., 2012. From throw–away traffic to bots: detecting the rise of dga-based malware. In: Usenix Conference on Security Symposium.

Bharathi, B., Bhuvana, J., 2019. Domain name detection and classification using deep neural networks: 6th international symposium, SSCC 2018, bangalore, india, september 1922, 2018, revised selected papers. Security in Computing and Communications.

Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C., 2014. Exposure: a passive dns analysis service to detect and report malicious domains. Acm Trans. Inf. Syst. Secur. 16 (4).

Hao, S., Syed, N.A., Feamster, N., Gray, A.G., Krasser, S., 2009. Detecting spammers with snare: Spatio-temporal network-level automatic reputation engine. In: 18th USENIX Security Symposium, Montreal, Canada, August 10–14, 2009, Proceedings.

He, W., Gou, G., Kang, C., Liu, C., Li, Z., Xiong, G., 2019. Malicious domain detection via domain relationship and graph models. In: 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), pp. 1–8. doi:10.1109/IPCCC47392.2019.8958718.

Huang, Q., Yu, J., Wu, J., Wang, B., 2020. Heterogeneous graph attention networks for early detection of rumors on twitter.

Kara, A.M., Binsalleeh, H., Mannan, M., Youssef, A.M., Debbabi, M., 2014. Detection of malicious payload distribution channels in dns. In: IEEE International Conference on Communications.

Khalil, I., Yu, T., Guan, B., 2016. Discovering malicious domains through passive DNS data graph analysis. In: Chen, X., Wang, X., Huang, X. (Eds.), Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2016, Xi'an, China, May 30, - June 3, 2016. ACM, pp. 663–674. doi:10.1145/2897845.2897877.

Kountouras, A., Kintis, P., Lever, C., Chen, Y., Nadji, Y., Dagon, D., Antonakakis, M., Joffe, R., 2016. Enabling network security through active dns datasets. Springer International Publishing.

Lee, J., Lee, H., 2014. Gmad: graph-based malware activity detection by dns traffic analysis. Comput. Commun. 49 (aug.1), 33–47.

Liang, Z., Zang, T., Zeng, Y., 2020. Malportrait: Sketch malicious domain portraits based on passive DNS data. In: 2020 IEEE Wireless Communications and Networking Conference, WCNC 2020, Seoul, Korea (South), May 25–28, 2020. IEEE, pp. 1–8. doi:10.1109/WCNC45663.2020.9120488.

Long, Y., Zhang, Y., Wu, M., Peng, S., Li, X., 2020. Predicting drugs for covid-19/sars-cov-2 via heterogeneous graph attention networks. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM2020).

Mockapetris, P.V., 1989. Dns encoding of network names and other types. ietf request for comments.

On, 2016. network-level clusters for spam detection.

Oprea, A., Li, Z., Yen, T.F., Sang, C., Alrwais, S., 2015. Detection of early-stage enterprise infection by mining large-scale log data. In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).

Park, K.H., Song, H.M., Yoo, J.D., Hong, S.-Y., Cho, B., Kim, K., Kim, H.K., 2022. Unsupervised malicious domain detection with less labeling effort. Comput. Secur. 116, 102662. doi:10.1016/j.cose.2022.102662. https://www.sciencedirect.com/science/article/pii/S016740482200061X.

Peng, C., Yun, X., Zhang, Y., Li, S., 2019. Malshoot: Shooting malicious domains through graph embedding on passive DNS data: Methods and protocols. Methionine Dependence of Cancer and Aging.

Rahbarinia, B., RobertoPerdisci, Antonakakis, M., 2015. Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks. In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks.

Schüppen, S., Teubert, D., Herrmann, P., Meyer, U., 2018. FANCI : Feature-based automated nxdomain classification and intelligence. In: Enck, W., Felt, A.P. (Eds.), 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15–17, 2018. USENIX Association, pp. 1165–1181. https://www.usenix.org/conference/usenixsecurity18/presentation/schuppen.

Sun, X., Tong, M., Yang, J., 2019. Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification.

Sun, X., Yang, J., Wang, Z., Liu, H., 2020. Hgdom: Heterogeneous graph convolutional networks for malicious domain detection. In: NOMS 2020 - IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, April 20–24, 2020. IEEE, pp. 1–9. doi:10.1109/NOMS47738.2020.9110462.

Vinayakumar, R., Soman, K.P., Poornachandran, P., Akarsh, S., Elhoseny, M., 2019. Improved DGA domain names detection and categorization using deep learning architectures with classical machine learning algorithms. Cybersecurity and Secure Information Systems.

Vissers, T., Spooren, J., Agten, P., Jumpertz, D., Janssen, P., Wesemael, M.V., Piessens, F., Joosen, W., Desmet, L., 2017. Exploring the ecosystem of malicious domain registrations in the.eu tld. Springer, Cham.

Wang, X., Ji, H., Shi, C., Wang, B., Cui, P., Yu, P., Ye, Y., 2019. Heterogeneous graph attention network. In: The World Wide Web Conference.

Xsa, B., Zwa, B., Jya, B., Xl, C., 2020. Deepdom: malicious domain detection with scalable and heterogeneous graph convolutional networks. Comput. Secur. 99.

Zhao, G., Xu, K., Xu, L., Wu, B., 2015. Detecting apt malware infections based on malicious dns and traffic analysis. IEEE Access 3, 1132–1142.

Zhao, J., Liu, X., Yan, Q., Li, B., Sun, L., 2020. Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning. Comput. Secur. 102 (6), 102152.

Zhauniarovich, Y., Khalil, I., Yu, T., Dacier, M., 2018. A survey on malicious domains detection through DNS data analysis. ACM Comput. Surv. 51 (4), 67:1–67:36. doi:10.1145/3191329.

Qing Wang is a PhD candidate in the Sixth Research Laboratory of Institute of Information Engineering, Chinese Academy of Sciences. She received the B.S. degree in software engineering from Henan University in China. Her research areas are network security situational awareness, malicious domain detection, and data mining.

Cong Dong received his B.S. degree in Information Management and Information System (Confidentiality Direction) from Tianjin University in China. Now he is pursuing the PhD degree in the Sixth Research Laboratory of Institute of Information Engineering, Chinese Academy of Sciences. His research areas are network security situational awareness and knowledge graph.

Shijie Jian received her B.S. degree in Information Security from the University of Science and Technology Beijing and her master's degree in Cyberspace Security from the University of Chinese Academy of Sciences in China. Her research areas are network security situational awareness and intrusion detection.

Dan Du received her M.S. degree in Computer Technology from University of Chinese Academy of Sciences in China. Now she is pursuing the PhD degree in the Sixth Research Laboratory of Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include network security situational awareness and attack-defense modeling.

Zhigang Lu received his PhD degree from the Graduate School of Chinese Academy of Sciences in China. He is currently a senior engineer at the Institute of Information Engineering, Chinese Academy of Sciences, and an associate professor at the Institute of Cyberspace Security, Chinese Academy of Sciences. His research areas are network security situational awareness, network attack detection, mobile terminal security, etc.

Yinhao Qi received his B.S. degree in the School of Information Engineering from Zhengzhou University in China. Now he is pursuing the PhD degree in the Sixth Research Laboratory of Institute of Information Engineering, Chinese Academy of Sciences. His research areas are network security situational awareness and anomaly detection in blockchain.

Dongxu Han received his master's degree from North China Electric Power University in China. He is currently an engineer at the Institute of Information Engineering, Chinese Academy of Sciences, and is pursuing a PhD in the field of network security. His research areas are network attack detection, network security situational awareness.

Xiaobo Ma is a professor in the School of Computer Science and Technology, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University in China. His current research interests include network security monitoring, DNS security, encrypted traffic analysis.

Fei Wang is a professor in the Institute of Computing Technology, Chinese Academy of Sciences in China. His current research interests include situational confrontation and spatiotemporal data prediction.

Yuling Liu received his PhD degree from the Institute of Software Research, Chinese Academy of Sciences in China. He is now a senior engineer at the Institute of Information Engineering, Chinese Academy of Sciences. His research areas are network security situational awareness, network security big data analysis, and security measurement and certification.