HinCTI: A Cyber Threat Intelligence Modeling and Identification System Based on Heterogeneous Information Network

Yali Gao, Xiaoyong Li, *Member, IEEE,* Hao Peng, Binxing Fang, and Philip S. Yu, *Fellow, IEEE*

Abstract—Cyber attacks have become increasingly complicated, persistent, organized, and weaponized. Faces with this situation, drives a rising number of organizations across the world are showing a growing willingness to leverage the open exchange of cyber threat intelligence (CTI) for obtaining a full picture of the fast-evolving cyber threat situation and protecting themselves against cyber-attacks. However, modeling CTI is challenging due to the explicit and implicit relationships among CTI and the heterogeneity of cyber-threat infrastructure nodes involved in CTI. Owing to the limited labels of cyber threat infrastructure nodes involved in CTI, automatically identifying the threat type of infrastructure nodes for early warning is also challenging. To tackle these challenges, a practical system called *HinCTI* is developed for modeling cyber threat intelligence and identifying threat types. We first design a threat intelligence meta-schema to depict the semantic relatedness of infrastructure nodes. We then model cyber threat intelligence on heterogeneous information network (HIN), which can integrate various types of infrastructure nodes and rich relations among them. Following, we define a meta-path and meta-graph instances-based threat Infrastructure similarity (MIIS) measure between threat infrastructure nodes and present a MIIS measure-based heterogeneous graph convolutional network (GCN) approach to identify the threat types of infrastructure nodes involved in CTI. Moreover, through the hierarchical regularization strategy, our model can alleviate the problem of overfitting and achieve good results in the threat type identification of infrastructure nodes. To the best of our knowledge, this work is the first to model CTI on HIN for threat identification and propose a heterogeneous GCN-based approach for threat type identification of infrastructure nodes. With *HinCTI*, comprehensive experiments are conducted on real-world datasets, and experimental results demonstrate that our proposed approach can significantly improve the performance of threat type identification compared to the existing state-of-the-art baseline methods. Our work is beneficial to greatly relieve security analysts from heavy analysis work and efficiently protect organizations against cyber-attacks.

Index Terms—Cyber threat intelligence, threat type identification, heterogeneous information network, graph convolutional network, threat infrastructure nodes.

1 Introduction

NOWADAYS, to obtain the overall picture of the fast-evolving cyber threat situation and protect themselves from the complicated, persistent, organized, and weaponized cyber-attacks, a rising number of organizations across the world are showing an increasing willingness to leverage the open exchange of cyber threat intelligence (CTI) [1]. CTI is evidence-based knowledge about an existing or emerging threat to assets and can be used to inform decisions regarding a subject's response to the threat [2]. As we know, cyber criminals usually make full use of network infrastructures (e.g., domain names and Internet Protocol or IP addresses) to conduct cyber-attacks. The Pyramid of Pain model [3] indicates six levels of threat indicators for detecting attack activities, and the lower three levels are file hashes, IP addresses, and domain names. These three levels are atomic indicators and can be consumed by network security devices such as intrusion detection system (IDS), firewall, and spam filters on email servers. Through the application program interfaces (APIs) provided by the threat intelligence sharing platforms (TISPs), users can acquire huge amounts of CTI about file hashes, IP addresses, and domain names (i.e., the lower three levels of the Pyramid of Pain model that are the focus of this study). Generally, diverse intelligence sources can help depict cyber-threat infrastructure nodes from different perspectives. For instance, a domain name can be described with information not only from commercial CTI sources such as IBM X-Force Exchange Platform[1] and ThreatBook[2] but also from the related datasets such as passive domain name system (DNS) and domain name blacklist. Facing increasingly sophisticated cyber-attacks, modeling CTI provides numerous advantages [4], [5], [6], [7], such as obtaining a full picture of the fast-evolving cyber threat situation and unveiling potential groups that are behind specific attacks. Take domain name infrastructure nodes as an example, the threat types of domain names can be spam URLs, brute force login attacks, malware activity, and botnet node activity. Identifying the threat types of infrastructure nodes not only benefits the fine-grained threat warning but also facilitates targeted defensive measures. Note that we only consider CTI represented in structured data in this research. The extraction of structured data from unstructured data such as security technique reports is another important research direction [8], [9].

- *Y. Gao, X. Li (Corresponding author) and B. Fang are with the Key Laboratory of Trustworthy Distributed Computing and Service (Beijing University of Posts and Telecommunications), Ministry of Education, Beijing, 100876, China. E-mail: {gaoyalibupt, lxyxjtu}@163.com, fangbx@bupt.edu.cn*
- *H. Peng is with the State Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100083, China. E-mail: penghao@act.buaa.edu.cn*
- *P. S. Yu is with Computer Science Department, University of Illinois at Chicago, Chicago, IL 60607, and also with the Institude for Data Science, Tsinghua University, Beijing 100084, China. E-mail: psyu@uic.edu*

1. https://api.xforce.ibmcloud.com/doc
2. https://x.threatbook.cn/private_api

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2987019, IEEE Transactions on Knowledge and Data Engineering

1.1 Motivation

The modeling of CTI and the threat type identification of infrastructure nodes should undoubtedly be the most fundamental requirements for any cyber threat defense and warning system. In the past few years, academic and industry communities in the fields of cybersecurity and data mining have been attracted to this topic, and many state-of-the-art studies have been carried out, such as [7], [10] and [11]. Some of them are very creative and elaborate, but most of them face the following two key limitations that must be solved.

First, few studies have focused on the problem of limited threat type labels of infrastructure nodes involved in CTI. Owing to the high cost of manual labeling, the threat labels of cyber-threat infrastructure nodes is incomplete in the CTI database, and the labels are annotated with threat types by intelligence providers or security analysts [11]. Thus, how to accurately and effectively learn from the limited labeled infrastructure nodes and a large number of relationships among them to predict the threat types of unlabeled nodes is a paramount concern and key task for most security analysts and operators [11].

Furthermore, few studies have focused on the higher-level semantic relations among cyber-threat infrastructure nodes from the perspective of heterogeneous information network (HIN) [12]. In a large-scale CTI sharing environment, graph-based automatic analysis has attracted significant research efforts in recent years [5], [10], [13]. However, most of these works primarily focus on homogeneous information networks or bipartite graphs, which cannot discover the higher-level semantic relations among different types of nodes. As a special type of information network, HIN involves multiple types of nodes or relations, which have different semantic meanings. Such complex and semantically enriched information networks have great potential for knowledge discovery [14], [15]. However, the application of HIN in CTI mining is largely unexplored. Although some works have considered multiple types of nodes and relations, they have not considered higher-level semantics. Modeling CTI on HIN can provide an efficient and compact representation of linked cyber-threat infrastructure nodes in various semantics, such as capturing the complex relations among different types of infrastructure nodes, distinguishing different cyber-attacks based on the differences of network behaviors, and exploring how adversaries organize campaigns and adapt their techniques. Thus, a practical model for CTI on HIN, which leverages network correlations for better mining of CTI, should be further explored to relieve security analysts from heavy analysis work [16].

1.2 Our Contributions

To the best of our knowledge, we are the first to simultaneously design a HIN for CTI modeling, and propose a *m*eta-path and meta-graph *i*nstances-based threat *i*nfrastructure *s*imilarity (*MIIS*) measure-based heterogeneous graph convolutional network (GCN) approach for threat type identification of cyber-threat infrastructure nodes. The main innovations of our mechanism go beyond those of existing approaches in terms of the following three aspects:

1) *A CTI modeling approach based on HIN is proposed from the perspective of computation (meta-path and meta-graph instances-based computing).* By modeling CTI based on HIN, the proposed framework can not only integrate infrastructure nodes involved in CTI in a semantically meaningful way, including domain name, IP addresses, malware hashes, email addresses, and their relations but also extract and incorporate higher-level semantics of infrastructure nodes.
2) *A MIIS measure-based heterogeneous GCN approach is proposed to identify the threat types of infrastructure nodes.* We define a *MIIS* measure between threat infrastructure nodes, and present a *MIIS* measure-based heterogeneous GCN approach to identify the threat type of infrastructure nodes. Through hierarchical regularization, the approach can alleviate the problem of overfitting and achieve good results in the threat type identification of infrastructure nodes. This research can also promote cyber security investigations with partial or incomplete information.
3) *A practical system called HinCTI is developed for modeling cyber threat intelligence and identifying threat types*. With the system, we conduct comprehensive experiments on real-world datasets, and experimental results demonstrate that our proposed approach can significantly improve the performance of threat type identification compared with the existing state-of-the-art baseline methods.

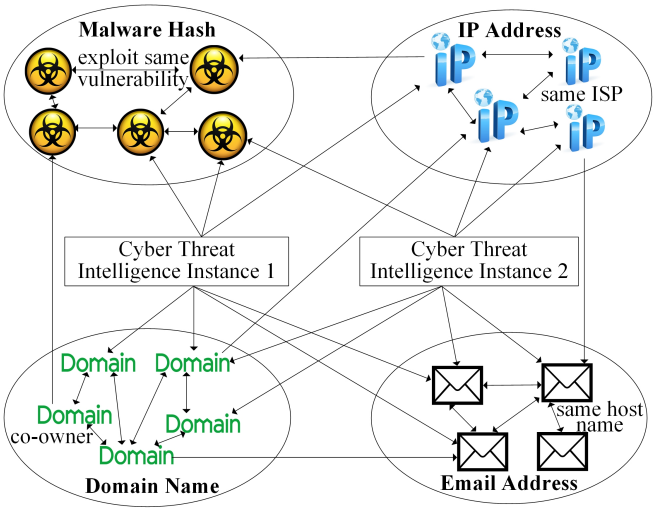

Fig. 1: Examples of two cyber threat intelligence instances involving different types of threat infrastructure nodes and edges.

These innovative designs collectively make *HinCTI* an efficient solution that can be used in the complex cyber security environment. A series of comprehensive experiments based on the real-world cyber-threat data from IBM X-Force Exchange Platform and other sources are conducted to evaluate the effectiveness and efficiency of the proposed approach. Experimental results demonstrate the superiority of the proposed approach by comparison with the state-of-the-art baseline methods.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 depicts the modeling of CTI on HIN, presents preliminary concepts, and gives an overview of the system architecture. Section 4 gives a detailed description of the proposed heterogeneous GCN-based threat type identification approach. Section 5 describes the experiments and performance results of the proposed approach by comparison with the state-of-the-art baseline methods. Section 6 summarizes the paper and outlines future work.

2 Related Work

The main contributions of our mechanism benefit from many existing representative work. In this section, we first review the typical work of modeling of CTI. We then analyze the graph-based threat identification and the network representation learning for threat identification.

2.1 Modeling of CTI

From the perspective of CTI sharing, numerous exchange formats, such as Structured Threat Information eXpression (STIX) [17], Incident Object Description and Exchange Format (IODEF) [18], and OpenIOC [19], are proposed to describe security incidents and observations related to attack campaigns. However, STIX, IODEF, and OpenIOC are not used for computational purposes. To extract and incorporate higher-level semantics of infrastructure nodes, CTI must be modeled from the perspective of computation.

The modeling of CTI based on multiple intelligence sources (e.g., IBM X-Force Exchange, and ThreatBook) can be very beneficial to discover the correlations among various cyber-attack events, facilitate the analysis of cyber attacks, and obtain a complete visibility across Kill Chain phases [20]. For instance, referring to IP and DNS registration information can be useful for malware database, and referring to malware database entries is useful for IP and DNS blacklists wherever appropriate. Likewise, a vulnerability database can refer to any malware samples, which exploit that vulnerability, and vice versa. Modi et al. [4] proposed an automated CTI fusion framework called ATIS, which considers multiple threat sources and connects apparently isolated cyber events. Gascon et al. [21] proposed MANTIS, a platform for CTI that provides a unified presentation for numerous standards and correlates threat data from different sources through a novel type-agnostic similarity algorithm based on attributed graphs. However, the similarity algorithm only considers the similarity of fingerprints (hash values) of any two objects, and the available higher-level semantics (indirect relations involving other types of nodes) are totally neglected. Boukhtouta et al. [5] presented an approach to investigate cyber-threats, in which tens of types of nodes are considered. However, the higher-level semantics among infrastructure nodes are not further analyzed.

Researchers have proposed approaches to automatically extract nodes and relations from unstructured CTI text, such as tweets, blogs, and forums [8], [9]. Liao et al. [8] proposed an approach to automatically extract Indicators of Compromises (IoCs) from blog posts in natural language. They model the problem as graph similarity problem and identify the IoC item if it has a similar graph structure as the training set. However, the identified IoCs do not preserve their roles in a malicious campaign, which makes analyzing the characteristics of campaign in different stages and correlating with field measurements difficult. Husari et al. [9] proposed TTPDrill, leveraging natural language processing (NLP) and information retrieval (IR) to extract threat actions from unstructured CTI text. However, we do not focus on the extraction of nodes and their relations from unstructured text, and we simply utilize the extraction results.

2.2 Graph-based Threat Identification

Graph-based threat identification is an important research approach in the fields of network security and data mining, and it offers the characterization of the interaction between infrastructure nodes and the identification of influential entities and groups. By leveraging the linkage information between infrastructure nodes of interest, graph-based methods can uncover the potential relationships, which are relatively harder for attackers to evade because making a cyber attack unavoidably generate plenty of links in the graph [22].

In recent years, a number of innovative graph-based threat identification methods have been developed for cyber security. However, existing research heavily focuses on homogeneous information networks, which can only perform simple correlation analysis. Manadhata et al. [13] leveraged graph inference and adapted belief propagation to detect malicious domain names. However, only the host-domain graph is constructed, and ignoring IP-domain graph and other informative graphs greatly hinders the accuracy of identification. Shi et al. [23] proposed a malicious domain name identification approach based on extreme machine learning (ELM), in which construction-based, IP-based, TTL-based, and Whois-based features are extracted to characterize a domain name and fed into ELM. However, ignoring relationships among different types of infrastructure nodes can greatly reduce the performance of identification. Some scholars developed an ontology for cyber security knowledge graphs to represent the rich relations between cyber entities [24], [25], [26]. However, the approach requires a significant amount of work to build and is somewhat difficult to use. In our previous work [27], we proposed a graph mining-based trust evaluation mechanism with multidimensional features for heterogeneous CTI. In this paper, we further analyze the higher-level relationship between heterogeneous infrastructure nodes and study the infrastructure nodes in a complex and semantically enriched HIN, which is simple to build and use.

Topic modeling techniques such as Latent Dirichlet Allocation (LDA) have been widely used for automatically identifying the topics of large amounts of source code whose purposes are unknown [28], [29]. Samtani et al. [30] applied classification and topic modeling techniques to explore the functions and characteristics of assets in hacker forums. In [31], the authors proposed AZSecure Hacker Assets Portal, in which LDA is utilized on online hacker forum source code to identify major hacker code topics. In [32], the authors leveraged topic modeling to analyze hacker community source code and explore emerging hacker assets and key hackers for proactive CTI. Given that we only consider CTI represented in structured data in this research, topic modeling-based approaches, which are usually used for textual data, are unsuitable for this task. The extraction of structured data from textual data has been studied, e.g., [8], [9]. Log analysis techniques are widely used in threat identification, such as analysis of DNS log data for detecting malicious domain names [33], [34] and analysis of system audit logs for finding entry point of an attack [9]. Pei et al. [35] presented HERCULE, which conducts community discovery on logs from multiple systems to reconstruct a complete, intuitive, and human-understandable attack story. However, the aim of our research is a re-mining of CTI data for threat identification, which is quite different from log analysis-based anomaly detection.

2.3 Network Representation Learning for Threat Identification

Network representation learning, i.e., network embedding, aims to embed network into a low dimensional space while pre-

serving the network structure and property so that the learned embedding can be easily applied by machine learning techniques. Recently, many efficient network embedding methods have been proposed to address representation learning problem for homogeneous network, such as DeepWalk [36], Node2Vec [37]. Compared to the widely studied homogeneous information network, the heterogeneous properties of HIN (i.e., containing multiple types of nodes or links) make directly apply homogeneous techniques for HIN representation learning difficult. To tackle this challenge, Dong et al. [38] proposed Metapath2Vec, which designs a meta-path-based random walk and utilizes skip-gram to perform heterogeneous graph embedding. However, Metapath2Vec can only utilize one meta-path and may ignore useful information. Fu et al. [39] proposed HIN2Vec to explore meta-paths in HINs for representation learning. Graph neural network (GNN) [40], [41] is proposed to extend the deep neural network to deal with arbitrary graph-structured data. Wang et al. [42] proposed heterogeneous graph attention network (HAN) to handle heterogeneous graph, considering node-level and semantic-level attentions. Compared to the research on areas such as bibliographic networks (classifying and clustering author and paper nodes) [38] and recommendation systems [43], [44], network representation learning has only recently been applied to the research on cybersecurity such as [45], [46].

3 CTI Modeling

In this section, we first define the problem of modeling CTI on HIN. We then introduce preliminary concepts. Finally, we give an overview of the system architecture.

3.1 CTI Modeling based on HIN

The definition and characterization of "CTI" have received substantial attention across academic communities, including network security [10] and data mining [11], [27]. A piece of CTI generally refers to cyber-attack-related evidence, involving a group of different types of threat infrastructures, such as malicious IP addresses, malicious domain names, malware hashes, and malicious email addresses. We name the above infrastructures as *threat infrastructure nodes*. Relationships exist between threat infrastructure nodes, including relationships between nodes of the same type and between nodes of different types, i.e., relationships between domain names, relationships between IP addresses, relationships between malware hashes, relationships between email addresses, and relationships among them. We name the above relationships as *threat infrastructure relations*.

Through the APIs provided by threat intelligence providers, including open-source communities such as IoC Bucket[3] and commercial CTI service providers such as ThreatBook, we can derive huge amounts of relations (i.e., *domain-IP*, *domain-malware*, *IP-malware*, *domain-email*, and *IP-email*) among different types of threat infrastructure nodes (i.e., domain names, IP addresses, email addresses, and malware hashes) to construct the cyber threat intelligence HIN. As for the relations between nodes of the same type, we extract related information from various sorts of external sources to enrich the context of threat infrastructure nodes. As shown in Fig. 1, the direct relations between two domain names can be enriched by domain-related service, such as from Whois[4] database to get relations of co-owner, co-organization, co-location of DNS, and co-registrar. The direct relations between two IP addresses can be enriched by IP-related service, such as from IP2Location[5] service to get relation of having the same internet service providers (ISPs). The direct relations between two malware hashes can be enriched by open-source malware analysis tools, such as from Common Vulnerabilities and Exposures (CVE) database to get relations of exploiting the same vulnerability. The direct relations between two email addresses can be enriched by the relation of same host name.

After extracting the above threat infrastructure nodes and threat infrastructure relationships from CTI instances and external sources, we can build a cyber threat intelligence HIN, as shown in Fig. 1, which contains four types of threat infrastructure nodes, i.e., malware hashes, IP addresses, domain names, and email addresses. The threat intelligence can be regarded as a group of threat infrastructure nodes and threat infrastructure relationships that can contribute to explain the relationship between various types of nodes. Thus, a piece of threat intelligence instance can be treated as a subgraph of the whole HIN. One particular advantage of HIN is that meta-paths (defined in Section 3.2) and meta-graphs (defined in Section 4.2) defined over node types can reflect semantically meaningful information about similarities and, thus, can naturally provide explainable results for threat analysis and identification. For instance, a relation between two domain names can be revealed by meta-path *Domain-Malware-Domain*, which describes two domain names are visited by the same malware, or by meta-path *Domain-Email-Domain* which describes two domain names registered by the same email address.

3.2 Preliminaries

Definition 1 (Cyber-Threat Infrastructure Nodes [5]). *As cyber-criminals usually make full use of network resources to conduct their malicious activities, we define that cyber-threat infrastructure nodes consist of IP addresses, domain names, malware hashes, and email addresses.*

The collected CTI from intelligence providers is generally in the form of hash values of malwares, malicious IP addresses and malicious domain names. Thus, we only consider the lower-level basic CTI and represent them as a HIN in this paper. The nodes in the graph represent cyber-threat infrastructures, i.e., domain names, IP addresses, malware hashes, and email addresses. In this paper, we investigate how to leverage the HIN to facilitate the mining of CTI datasets.

Definition 2 (HIN [47]). *A HIN is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and a relation type mapping $\psi : \mathcal{E} \rightarrow \mathcal{R}$, where V denotes the node set, and E denotes the link set. $\mathcal{A}$ denotes the node type set, and $\mathcal{R}$ denotes the relation type set, where the number of node types $|\mathcal{A}| > 1$ or the number of relation types $|\mathcal{R}| > 1$.*

Fig. 1 gives an example of two CTI instances connected with different types of nodes and relationships. After given a complex HIN for CTI modeling, describing its meta-level (i.e., schema-level) is necessary for better understanding.

Definition 3 (Meta-Schema (or Network Schema)). *Given a HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node type mapping $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and the relation*

3. https://www.iocbucket.com/

4. https://www.whois.com/

5. https://www.ip2location.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2987019, IEEE Transactions on Knowledge and Data Engineering

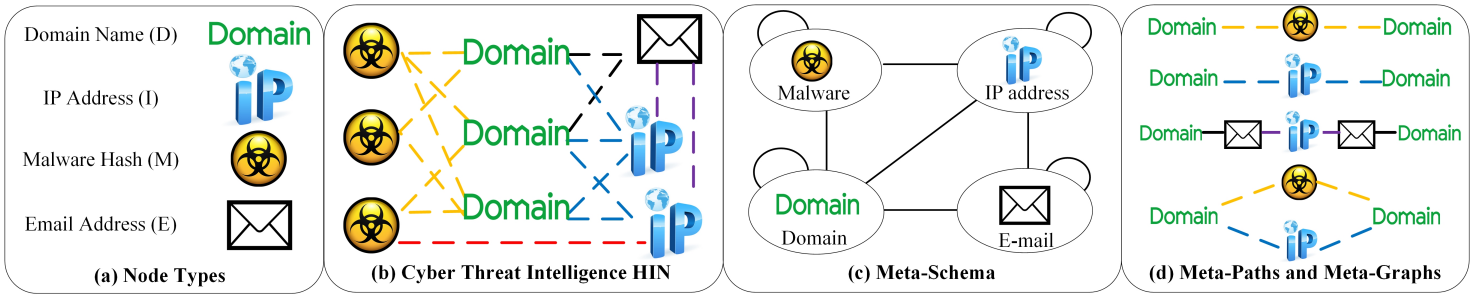

Fig. 2: CTI modeling based on HIN. (a) Four types of nodes (i.e., Domain Name (D), IP Address (I), Malware Hash (M), Email Address (E)). (b) The cyber threat intelligence HIN consists of four types of nodes and five types of relationships. Five different colored lines represent five distinct relations among various types of nodes. (c) Meta-schema of cyber threat intelligence HIN. (d) Examples of meta-paths and meta-graphs involved in *HinCTI* (e.g., *domain-malware-domain*, *domain-IP-domain*).

type mapping $\psi : \mathcal{E} \rightarrow \mathcal{R}$, the meta-schema (or network-schema) for network $\mathcal{G}$, denoted as $T_G = (\mathcal{A}, \mathcal{R})$, is a graph with nodes as node types from $\mathcal{A}$ and edges as relation types from $\mathcal{R}$.

As described in Fig. 2, CTI modeling involves four types of nodes (i.e., domain names, IP addresses, malware hashes, and email addresses), and five types of relations among different types of nodes (i.e. *R*, *S*, *G*, *C*, *N*, as shown in Table 2). Fig. 2(c) shows an example of the HIN meta-schema characterizing the relationships of threat infrastructures described in CTI. Another important concept of HIN is meta-path defined over types, which can formulate the semantics of higher-level relationships among nodes and, thus, can naturally provide explainable results for threat infrastructure modeling. Here, we follow this concept and extend it to our *HinCTI* model.

Definition 4 (Meta-Path [47]). *A meta-path P is a path defined on the graph of network schema $T_{\mathcal{G}} = (\mathcal{A}, \mathcal{R})$ and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_d} A_{d+1}$, which defines a composite relation $R = R_1 \cdot R_2 \cdot \ldots \cdot R_d$ between node types A_1 and A_{d+1}, where symbol $\cdot$ denotes the composition operator on relations, and d is the length of P.*

In general, a meta-path corresponds to a type of path within the network schema, containing a certain sequence of link types. For simplicity, we use object types connected by symbol "," to denote the meta-path when there is only one relationship between a pair of types: $P = (A_1, A_2, \ldots, A_{d+1})$. If $\forall l, \phi(v_l) = A_l$ and edge $e_l = \langle v_l, v_{l+1} \rangle$ belongs to relation type $R_l \in P$, then a meta-path instance $p = (v_1, v_2, \ldots, v_{d+1})$ between v_1 and v_{d+1} in network $\mathcal{G}$ follows the meta-path $P = (A_1, A_2, \ldots, A_{d+1})$. We further introduce semantically meaningful meta-paths that describe infrastructure node relations in Section 4.2.

The literature gives many definitions of the term "threat type identification", and they vary from team to team and from project to project. Here, we give a clear definition for describing the purpose of the paper as follows [48].

Definition 5 (Threat Type Identification). *For the collected cyber-threat infrastructure nodes without threat labels, threat type identification means to identify their threat type labels by the constructed heterogeneous GCN-based threat type identification model, leveraging those cyber-threat infrastructure nodes with threat labels and the relations among them.*

On the threat intelligence sharing platforms, a large number of threat-infrastructure nodes are without threat labels, which is incomplete for CTI consumers. Thus, predicting the threat types of nodes without threat labels leveraging the threat-infrastructure nodes and their relations involved in the large amount of basic CTI is of great significance.

3.3 System Architecture

The architecture of our proposed CTI modeling and identification system based on HIN, called *HinCTI*, is shown in Fig. 3, which mainly consists of the following four modules:

- ***CTI Modeling based on HIN.*** Through the APIs provided by various CTI providers, we can obtain a large amount of valuable CTI, involving massive threat infrastructure nodes and relationships among them. In cyber threat intelligence HIN, the more context information correlates with nodes, the more conducive for CTI analysis. Thus, to enrich the context of infrastructure nodes, we extract information from external databases to establish relations between nodes of the same type and different types, e.g., "Whois" database for both domain name and IP nodes, "CVE" database for malware nodes, and "Passive DNS" database for both domain name and email address nodes. In this way, cyber threat intelligence HIN is constructed to depict the relationships among various types of infrastructure nodes.
- ***Feature Extractor and Meta-path and Meta-graph Builder.*** Based on the meta-schema designed for cyber threat intelligence HIN, we build a set of meta-paths and meta-graphs to capture the higher-level relatedness over infrastructure nodes from different semantic meanings.
- ***Heterogeneous GCN-based Threat Type Identification.*** We first extract infrastructure node features and generate node feature matrix X. Then, meta-graph based adjacent matrices are aggregated to obtain the weighted adjacent matrix B. Finally, we leverage heterogeneous GCN to fuse X and B to learn the threat types of cyber-threat infrastructure nodes.
- ***Threat Type Identifier.*** For each newly collected unknown threat infrastructure node, the node features will be first extracted, then its related infrastructure nodes will be extracted from external sources. The relationships among these infrastructure nodes will be further analyzed. Based on the extracted features and the constructed heterogeneous GCN-based threat type identification model, the threat type of the infrastructure node will be labeled by the threat identifier. Based on the identified threat type label, security analysts can give early warning and adopt defensive strategies.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2987019, IEEE Transactions on Knowledge and Data Engineering

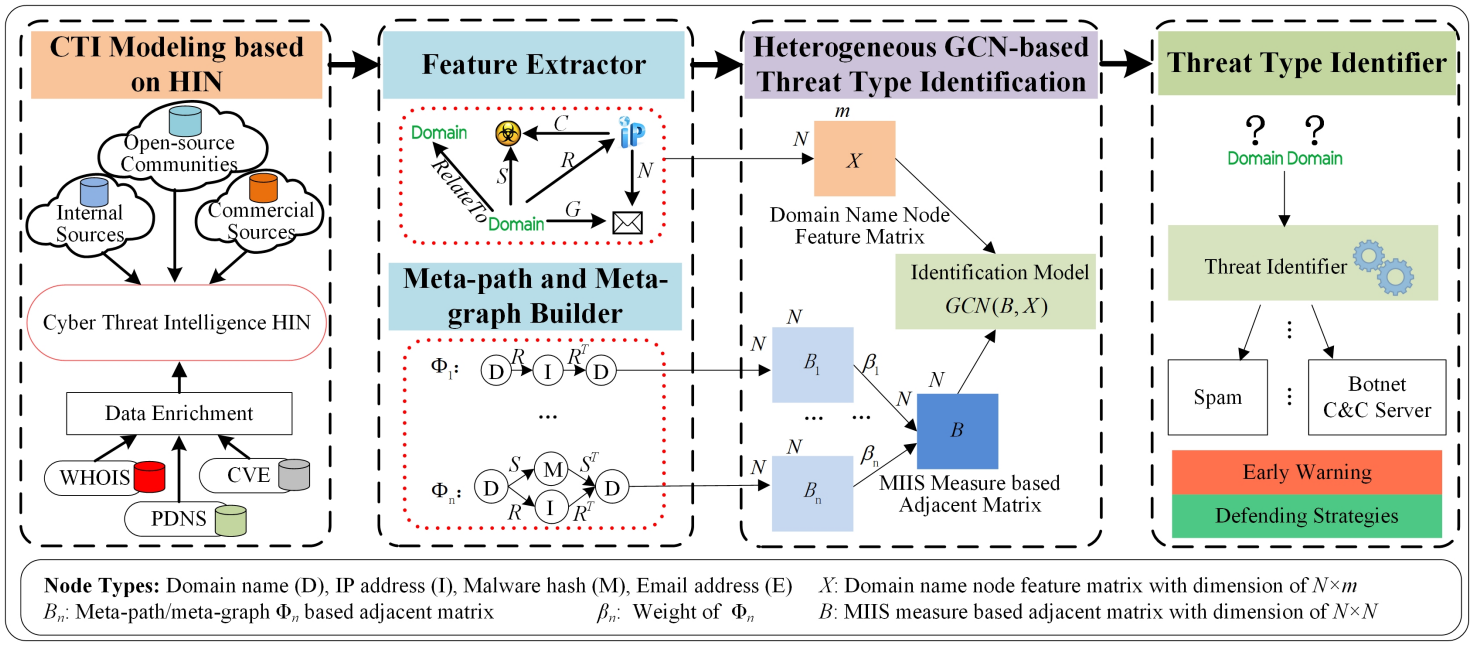

Fig. 3: System architecture of the proposed *HinCTI*. (1) Modeling of CTI on HIN, and generation of cyber threat intelligence HIN. (2) Extraction of node features and designing of a set of meta-paths and meta-graphs based on cyber threat intelligence HIN. (3) The node feature matrix X and the *MIIS* measure-based adjacent matrix B of domain name infrastructure nodes are the inputs of the heterogeneous GCN model. (4) The heterogeneous GCN model predicts the threat types of domain name infrastructure nodes, such as spam, and botnet C&C server. The threat type identification results of infrastructure nodes can be used for giving early warning and adopting defensive strategies. Note that the threat type identification task of different types of nodes (i.e., D, I, M, and E) in *HinCTI* are carried out separately, and we take the domain name infrastructure node (i.e., D) as an example.

4 Proposed Threat Type Identification Approach

In this section, we first introduce feature extraction, followed by the building of meta-paths and meta-graphs. We then describe the heterogeneous GCN-based threat type identification approach, and finally depict how the hierarchical regularization strategy alleviate the problem of overfitting. As CTI about domain names is more static and efficient than other types of infrastructure nodes in cybersecurity [3], we specifically focus on the threat type identification of domain name infrastructure nodes. Before the detailed description of the proposed approach, we first list key notations and their descriptions in Table 1.

TABLE 1: Notations and their descriptions.

Notation	Description
X	feature matrix of infrastructure nodes
m	dimension of infrastructure node features
N	number of infrastructure nodes
Φ	meta-path and meta-graph set $\Phi = \{\Phi_k \mid k = 1, 2, \ldots, n\}$
v_i	the i^{th} infrastructure node
$NumP_{\Phi_k}(v_i, v_j)$	number of meta-path and meta-graph instances under Φ_k between two infrastructure nodes v_i and v_j
$MIIS(v_i, v_j)$	meta-path and meta-graph instances-based similarity between infrastructure nodes v_i and v_j
B_k	adjacent matrix based on Φ_k
$\boldsymbol{\beta}$	weight vector of meta-path and meta-graph set Φ, where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_n]$ and β_k is the weight of Φ_k.
B	*MIIS* measure-based adjacent matrix
U_{Φ_k}	commuting matrix under Φ_k
L	threat type label set, where $L = \{l_i \mid i = 1, 2, \ldots, K\}$ and K is the number of labels.
L_i	child threat type label set of l_i, where $L_i = \{l_i^{(j)} \mid j = 1, 2, \ldots, K_i\}$, $l_i^{(j)}$ is the j^{th} child label of l_i, and K_i is the number of l_i's child labels.
W	parameter vector of labels in the final output layer of GCN model, where $W = [w_{l_1}, w_{l_2}, \ldots, w_{l_K}]$ and w_{l_i} is that of label l_i.

4.1 Feature Extraction

Node features. Domain names are frequently used by attackers to keep in touch with server. The malicious domain names have different attributes compared with benign domain names. Legitimate web owners choose a succinct domain name so that users can remember it better, whereas malicious domains are usually generated by domain name generation algorithm (DGA) in batches. That is, the average length of malicious domain names is longer than that of benign domain names [49]. Regarding the information entropy of distribution of alphanumerics within a domain name, the entropy is an expression of the disorder, and the higher the entropy, the more chaotic the distribution [50]. The character distribution of Domain-Flux based malicious domain names is usually chaotic [50]. Thus, we choose the length and information entropy of domain name as the character-based features in the threat type identification of domain names.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2987019, IEEE Transactions on Knowledge and Data Engineering

The active time of domain names for malicious purpose is considered short [23]. Whenever old domain names are deactivated by authorities, attackers register new ones rapidly and employ them for malicious purpose before they are detected and blocked by authorities, which typically makes the life time of malicious domain name much shorter. By contrast, benign web owners usually register their domain names for long-term business. Thus, we define the active time of domain name as the interval (counted in days) between registration expiration date and creation date based on the Whois data. Moreover, given that a legitimate domain name is frequently queried by users, owners of legitimate domain names will promptly update their Whois information to ensure the domain names serve users well. To the contrary, owners of malicious domain names do not update Whois information, and their update frequency is lower than that of owners of benign domain names. Following, we take the active time and update frequency of domain name as the time-related node features.

Relation-based Features. Although node features of a domain name can be used to reflect their behaviors and detect malicious domain names like "amazon-gst-sale.com", the intrinsic and complex relationships between it and its associated malwares can provide more critical information the identification. The relationships extracted among the nodes can provide a higher level of representation than that of the simple statistics, which requires more efforts for attackers to evade the detection. The area of attack will be greatly reduced if the attackers reduce communication with the related malwares, domain names, and IP addresses. Thus, to analyze the increasingly sophisticated malicious domain names, we consider not only the node features, but also the relationships summarized in Table 2, in which "element" denotes the element in the related relation matrices.

- R: To describe the relation between a domain name and the IP address it resolved to, we build the *domain-resolvedTo-IP* matrix R where each element $r_{ij} \in \{0, 1\}$ means if domain i is resolved to IP address j.
- S: To represent the relation between a malware and a domain name, we generate the *domain-visitedBy-malware* matrix S where each element $s_{ij} \in \{0, 1\}$ denotes whether domain name i is visited by malware j.
- G: To describe the relation between a domain name and an email address, we generate the *domain-registeredBy-email* matrix G where each element $g_{ij} \in \{0, 1\}$ denotes if domain name i is registered by email address j.
- C: To denote the relation that an IP address communicates with malware, we generate the *IP-communicateWith-malware* matrix C where each element $c_{ij} \in \{0, 1\}$ denotes if IP address i has communicated with malware j.
- N: To represent the relation that an IP address connects to an email address, we generate the *IP-connectTo-email* matrix N where each element $n_{ij} \in \{0, 1\}$ denotes whether IP address i has connected to email address j. Note that matrix R^T, S^T, G^T, C^T ,and N^T represent the transposed matrix of R, S, G, C, and N, respectively.

4.2 Meta-path and Meta-graph Builder

Although meta-path can be used to depict the correlations between nodes, it fails to capture a more complex relationship. Meta-graph [51] is proposed to use a directed acyclic graph of nodes to handle more complex relationship between HIN nodes, which can be defined as follows:

TABLE 2: Descriptions of relation matrices.

Matrix	Element	Description
R	r_{ij}	If *domain*$_i$ is resolved to *IP*$_j$, then $r_{ij} = 1$; otherwise, $r_{ij} = 0$.
S	s_{ij}	If *domain*$_i$ is visited by *malware*$_j$, then $s_{ij} = 1$; otherwise, $s_{ij} = 0$.
G	g_{ij}	If *domain*$_i$ is registered by *email*$_j$, then $g_{ij} = 1$; otherwise, $g_{ij} = 0$.
C	c_{ij}	If *IP*$_i$ communicates with *malware*$_j$, then $c_{ij} = 1$; otherwise, $c_{ij} = 0$.
N	n_{ij}	If *IP*$_i$ connects to *email*$_j$, then $n_{ij} = 1$; otherwise, $n_{ij} = 0$.

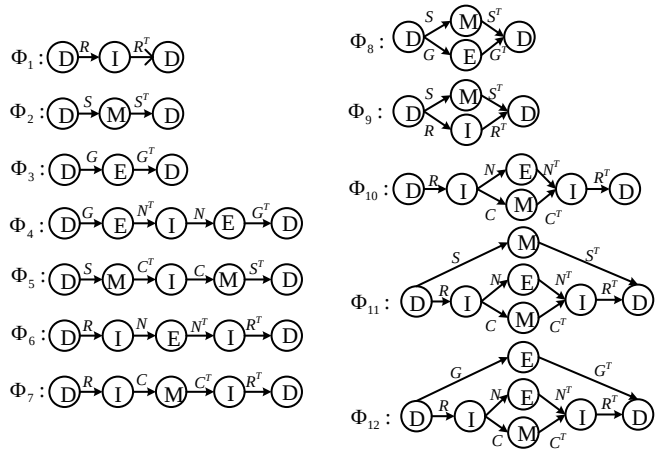

Fig. 4: Meta-paths and meta-graphs designed for threat type identification of domain name infrastructure nodes. The symbol D stands for domain name, I stands for IP address, M stands for malware hash, and E stands for email-address.

Definition 6 (Meta-Graph [51])**.** *A meta-graph Φ_i is a directed acyclic graph with single source node n_s and single target node n_t, defined on a HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with schema $T_\mathcal{G} = (\mathcal{A}, \mathcal{R})$. Formally, a meta-graph is defined as $\Phi_i = (\mathcal{V}_{\Phi_i}, \mathcal{E}_{\Phi_i}, \mathcal{A}_{\Phi_i}, \mathcal{R}_{\Phi_i}, n_s, n_t)$, where $\mathcal{V}_{\Phi_i} \subseteq \mathcal{V}$ and $\mathcal{E}_{\Phi_i} \subseteq \mathcal{E}$ are constrained by $\mathcal{A}_{\Phi_i} \subseteq \mathcal{A}$ and $\mathcal{R}_{\Phi_i} \subseteq \mathcal{R}$, respectively.*

As depicted in Fig. 4, different meta-paths and meta-graphs characterize the relatedness over threat infrastructure nodes from different aspects, i.e., with varying semantic meanings. For instance, meta-path Φ_1 depicts the relatedness over threat infrastructures through the *domain-IP* relations (i.e., two domain names are both resolved to the same IP address). Meta-path Φ_2 describes the relatedness over infrastructure nodes through the *domain-malware* relations (i.e., two domain names are both visited by the same malware). Meta-graph Φ_{11} depicts the relatedness over threat infrastructures from a more comprehensive view which incorporates both external and intrinsic connections. That is, in meta-graph Φ_{11}, two domain names are connected as they are both visited by the same malware (external connection), and their resolved IP addresses not only connect to the same e-mail address but communicate with the same malware (intrinsic connection).

In our approach, to detect the threat types of infrastructure nodes, meta-path and meta-graph are jointly considered to capture the complex relatedness among infrastructure nodes which is more expressive than pure meta-path-based or pure meta-graph-based approaches. Different meta-paths and meta-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2987019, IEEE Transactions on Knowledge and Data Engineering

graphs measure the relatedness between two infrastructure nodes from different views. That is, the more meaningful meta-paths and meta-graphs enumerated by the meta-schema, the higher accuracy of the similarity measure is. To detect the threat type of domain name infrastructure nodes, based on the meta-schema described in Fig. 2(c) and the domain knowledge of human experts in the field of cyber security, we enumerate 12 meaningful symmetric meta-paths and meta-graphs (i.e., Φ_1–Φ_{12} shown in Fig. 4) over different lengths to characterize the relatedness over domain name infrastructure nodes.

4.3 Heterogeneous GCN-based Threat Type Identification

After extracting the features of infrastructure nodes and designing the meaningful meta-paths and meta-graphs depicted in the previous subsections, we introduce the proposed *MIIS* measure-based heterogeneous GCN approach to identify the threat types of infrastructure nodes involved in CTI. This heterogeneous GCN, which simultaneously integrates node features and meaningful meta-path and meta-graph-based similarity adjacency relations, enables the representation of infrastructure nodes in a more comprehensive way. Before the definition of *MIIS*, we present the definition of number of meta-path and meta-graph instances under Φ_k, called $NumP_{\Phi_k}$, as follows:

Definition 7 (*Num*ber of meta-*P*ath and meta-graph instances under Φ_k, $NumP_{\Phi_k}$)**.** *Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its network schema $T_{\mathcal{G}} = (\mathcal{A}, \mathcal{R})$ and a symmetric meta-path or meta-graph Φ_k, the number of meta-path/meta-graph instances under Φ_k between two domain name infrastructure nodes v_i, v_j, denoted as $NumP_{\Phi_k}(v_i, v_j)$, can be defined as*

$$NumP_{\Phi_k}(v_i, v_j) = U_{\Phi_k}(v_i, v_j), \tag{1}$$

where U_{Φ_k} is the commuting matrix between domain name infrastructure nodes under Φ_k.

As for **meta-path** $\Phi_k = (A_1, A_2 \ldots, A_{d+1})$, its commuting matrix between node type A_1 and A_{d+1} can be calculated as

$$U_{\Phi_k} = Q_{A_1A_2} \cdot Q_{A_2A_3} \cdot \ldots \cdot Q_{A_dA_{d+1}}, \tag{2}$$

where $Q_{A_iA_{i+1}}$ is the adjacency matrix between type A_i and type A_{i+1}, and symbol $\cdot$ represents the matrix multiplication. However, as for **meta-graph**, the problem of calculating its commuting matrix to get the count of meta-graph instances becomes more complicated. Taking meta-graph Φ_{10} in Fig. 4 as an example, the two ways to pass through the meta-graph are path (D, I, E, I, D) and path (D, I, M, I, D). Note that D represents the type of domain name infrastructure node in cyber threat intelligence HIN. In the path (D, I, E, I, D), (I, E, I) means that two IP addresses (I) have similarities if they both connect to the same email address (E). Similarly, in the path (D, I, M, I, D), (I, M, I) means that two IP addresses (I) have similarities if they both communicate with the same malware (M). Inspired by [51], we define our logic of similarity when there are multiple ways for a flow passing through the meta-graph from the source node to the target node. When there are multiple paths, we constrain a flow to satisfy all of them, which requires one more matrix operation than simple matrix multiplication, i.e., the Hadamard product (Schur product). Taking meta-graph Φ_{10} in Fig. 4 as an example, **Algorithm 1** depicts how to calculate its commuting matrix, where $\odot$ is the Hadamard product and N, C, and R represent the IP-email, IP-malware, and domain-IP adjacency matrices, respectively, as shown in Table 2. After obtaining U_{P_r}, it is easier to obtain the whole commuting matrix $U_{\Phi_{10}}$ by the multiplication of a sequence of matrices. In practice, all the meta-paths and meta-graphs (i.e., Φ_1–Φ_{12} shown in Fig. 4) defined in this paper can be computed by multiplication operations and Hadamard product on the corresponding matrices.

Algorithm 1: Calculation of commuting matrix for $M_{\Phi_{10}}$.

1 Calculate $U_{P_1} = Q_{IE} \cdot Q_{IE}^T = N \cdot N^T$, where P_1 is the subpath (I, E, I);
2 Calculate $U_{P_2} = Q_{IM} \cdot Q_{IM}^T = C \cdot C^T$, where P_2 is the subpath (I, M, I);
3 Calculate $U_{P_r} = U_{P_1} \odot U_{P_2}$;
4 Calculate $U_{\Phi_{10}} = Q_{DI} \cdot U_{P_r} \cdot Q_{DI}^T = R \cdot U_{P_r} \cdot R^T$.

As described in Section 4.2, we design 12 meta-paths and meta-graphs (i.e., Φ_1–Φ_{12}) with different types of nodes and relations. As different meta-paths and meta-graphs can define different similarities and introduce different higher-level semantics, it is natural to incorporate all useful meta-paths and meta-graphs when identifying the threat type of infrastructure nodes. However, different meta-paths and meta-graphs have varying importance. Treating different meta-paths and meta-graphs equally is unpractical and weakens the semantic information provided by the meaningful meta-paths and meta-graphs. For example, domain name D_1 can either connect to domain name D_2 via meta-path (D_1, E_1, D_2) (both registered by the same email address E_1) or connect to domain name D_3 via meta-path (D_1, M_1, D_3) (both visited by the same malware M_1). When considering more on the source of threat, meta-path (D, E, D) usually plays a more important role than that of (D, M, D); however, it will be the other way around when considering more on the behavior of threat. Thus, given that different meta-paths and meta-graphs depict the relatedness over threat infrastructures in very diverse ways, to explore the complementary nature of these different aspects, we propose to leverage a meta-path and meta-graph-based weighted adjacent matrix to incorporate different semantics. Here, we define a similarity with weights for any two threat infrastructure nodes v_i and v_j, which is denoted as $MIIS(v_i, v_j)$ and defined as follows:

Definition 8 (*MIIS*)**.** *Given a meta-path and meta-graph set, denoted as $\Phi = \{\Phi_k | k = 1, 2, \ldots, n\}$, the MIIS measure between two infrastructure nodes v_i and v_j can be defined as*

$$MIIS(v_i, v_j) = \sum_{k=1}^{n} \beta_k \frac{2 \times NumP_{\Phi_k}(v_i, v_j)}{NumP_{\Phi_k}(v_i, v_i) + NumP_{\Phi_k}(v_j, v_j)}, \tag{3}$$

where $NumP_{\Phi_k}(v_i, v_j)$ is the number of meta-path and meta-graph instances between infrastructure nodes v_i and v_j under Φ_k, $NumP_{\Phi_k}(v_i, v_i)$ is that between v_i and v_i, $NumP_{\Phi_k}(v_j, v_j)$ is that between v_j and v_j. We use the parameter vector $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_n]$ to denote the weights of Φ, where β_k is the weight of meta-path/meta-graph Φ_k and satisfies $\beta_k \geq 0, \sum_{k=1}^{n} \beta_k = 1$.

The *MIIS* measure is defined from the perspective of two parts: the semantic overlap, which is defined by the number of paths between threat infrastructures v_i and v_j, and the semantic broadness, which is defined by the number of path instances between themselves (i.e., paths from v_i to v_i, and paths from v_j to v_j). The weight vector $\boldsymbol{\beta}$, which can be learned automatically,

is leveraged to incorporate the meta-path and meta-graph-based node similarities together.

After calculating the similarity of any two domain name infrastructure nodes by the *MIIS* measure, we can construct a matrix B with dimension of $N \times N$, where N is the number of domain name nodes and $B_{ij} = B_{ji} = MIIS(v_i, v_j)$. According to the description in Section 4.1, we can derive the domain name node feature matrix X with dimension of $N \times m$. Doing so is an obvious way to leverage the popular two-layer GCN [52] architecture to identify the threat types of infrastructure nodes. Here, the category labels represent the threat types of infrastructure nodes. The input of the GCN-based identification model is B and X, with $B \in \mathbb{R}^{N \times N}$, $X \in \mathbb{R}^{N \times m}$, which contains the m-dimensional original domain name node features. We first calculate $\hat{B} = \widetilde{D}^{-\frac{1}{2}} \widetilde{B} \widetilde{D}^{-\frac{1}{2}}$, where $\widetilde{B} = B + I_N$ is the *MIIS* measure-based adjacency matrix with added self-connections, I_N is the identity matrix, and $\widetilde{D}$ is diagonal matrix with $\widetilde{D}_{ii} = \sum_j \widetilde{B}_{ij}$. Then, the forward model takes the following simple form:

$$Z = f(X, B) = softmax(\hat{B}\, ReLU\, (\hat{B} X W^{(0)}) W^{(1)}), \quad (4)$$

where $ReLU$ denotes an activation function defined as $ReLU(\cdot) = max(0, \cdot)$, and the *softmax* activation function is applied row-wise, which is defined as $\textit{softmax}(x_i) = e^{x_i} / \sum_j e^{x_j}$. The neural network weights $W^{(0)} \in \mathbb{R}^{m \times h}$ is an input-to-hidden trainable weight matrix for a hidden layer with h feature maps; the neural network weights $W^{(1)} \in \mathbb{R}^{h \times K}$ is a hidden-to-output trainable weight matrix, where K is the number of threat type labels. Both are trained using gradient descent, and we perform batch gradient descent using the full dataset for every training iteration, which is a viable option as long as datasets fit in memory. Stochasticity in the training process is introduced via dropout [53].

Given a set of threat type labeled threat infrastructures, our model optimizes the cross-entropy H between the true label distribution and the predicted distribution as follows:

$$H = -\sum_{i \in \mathcal{Y}_L} \sum_{k=1}^{K} (l_k(v_i) \ln Z_k(v_i) + (1 - l_k(v_i)) \ln(1 - Z_k(v_i))), \quad (5)$$

$\mathcal{Y}_L$ is the set of domain name infrastructure node indices that have labels, K is the number of labels in the hierarchy, $l_k(v_i)$ is a binary label to indicate whether infrastructure node v_i belongs to label k, and $Z_k(v_i)$ refers to the probability of neural network prediction of label k for infrastructure node v_i.

4.4 Hierarchical Regularization

If we simply treat each label as an independent decision, then Eq. (5) can be used directly to train the neural network. However, there is usually a hierarchy structure among the threat type labels, in which a parent label contains several child labels. Fig. 5 shows examples of threat label hierarchy of all threat infrastructure nodes. The parent label "BotNet" contains multiple child labels, such as BruteForce, spam, Command and Control (C&C) server, backdoor, etc. Thus, introducing hierarchical dependencies among labels can improve the performance of threat type identification. That is, when a leaf label (which has no child label in the hierarchical structure) has few training examples, the decision can be regularized by its parent label. Inspired by [54] and [55], we use a hierarchical regularization over the final output layer of GCN model. As a simplification, the hierarchical dependencies among labels encourage the parameters of labels

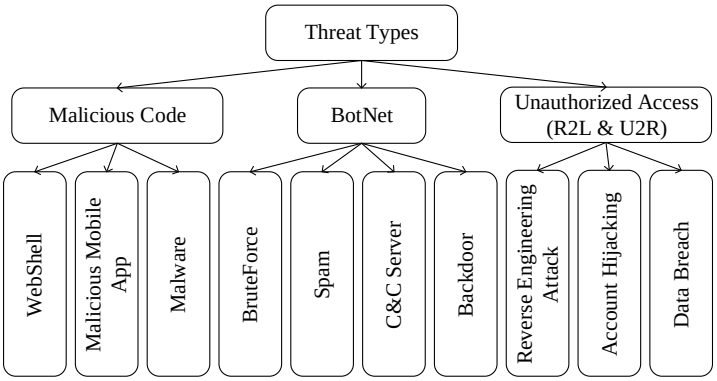

Fig. 5: Simplified example of threat label hierarchy of all threat infrastructure nodes, where R2L represents unauthorized access from a remote machine to a local machine and U2R represents unauthorized access to local superuser privileges by a local unprivileged user.

with hierarchical relationships to be similar. For instance, in Fig. 5, there is an edge between labels "BotNet" and "C&C Server", so the parameters of these two labels tend to be similar to each other.

Formally, we denote the threat type label set as $L = \{l_i | i = 1, 2, \ldots, K\}$, where K is the number of labels. As we focus on the hierarchical relationships of labels, we denote $L_i = \{l_i^{(j)} | j = 1, 2, \ldots, K_i\}$ as the child label set of label l_i where K_i is the number of l_i's child labels. We denote $W = [w_{l_1}, w_{l_2}, \ldots, w_{l_K}]$ as the parameters of labels in the final output layer of GCN model, where w_{l_i} is that of label l_i. We then use the following hierarchical regularization strategy to regularize the parameters in the final output layer:

$$\lambda(W) = \sum_{i=1}^{K} \sum_{j=1}^{K_i} \frac{1}{2} ||w_{l_i} - w_{l_i^{(j)}}||^2. \quad (6)$$

Finally, we use the following loss function with hierarchical regularization to optimize the parameters:

$$J = H + C\lambda(W), \quad (7)$$

where C is the penalty parameter.

From above, the overall process of *HinCTI* can be briefly described as **Algorithm 2**.

4.5 Analysis of the Proposed Approach

The proposed HinCTI can deal with various types of infrastructure nodes and relations and fuse rich semantics in a heterogeneous graph. Information can transfer from one type of node to another via diverse relationships. Benefitted from such a cyber threat intelligence HIN, diverse semantics can enhance the threat identification of infrastructure nodes. We then give the analysis of computational complexity of our proposed approach as follows. With regard to *MIIS* measure, multiplying adjacency matrices of a meta-path/meta-graph in a natural sequence way can be inefficient. However, the classic matrix chain multiplication problem can be optimized by dynamic programming [56] in $O(d^3)$, where d is the length of a meta-path/meta-graph which is usually very small. With regard to GCN training, inspired by [52], we leverage TensorFlow [57] for an efficient

Algorithm 2: Overall process of *HinCTI*.

Input: The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; meta-path and meta-graph set $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_n\}$; feature matrix of infrastructure nodes X; the set of node indices that have labels in the training set $\mathcal{Y}_L$ and their labels $L = \{l_i | i = 1, 2, \ldots, K\}$.

Output: Predicted labels of nodes in the testing set.

for $\Phi_i \in \{\Phi_1, \Phi_2, \ldots, \Phi_n\}$ **do**
  Calculate commuting matrix U_{Φ_k} using Eq. (2) and **Algorithm** 1;
  Calculate the number of meta-path and meta-graph instances using Eq. (1);
end
Calculate $MIIS(v_i, v_j)$ using Eq. (3), and get B;
Fuse X and B using Eq. (4);
Calculate cross-entropy $H \leftarrow -\sum_{i \in \mathcal{Y}_L} \sum_{k=1}^{K} (l_k(v_i) \ln Z_k(v_i) + (1 - l_k(v_i)) \ln(1 - Z_k(v_i)))$;
Calculate hierarchical regularization term $\lambda(W) \leftarrow \sum_{i=1}^{K} \sum_{j=1}^{K_i} \frac{1}{2} ||w_{l_i} - w_{l_i^{(j)}}||^2$;
Calculate loss function $J \leftarrow H + C\lambda(W)$;
Back propagation and update parameters in heterogeneous GCN-based threat type identification model;
return The predicted labels of nodes in the testing set.

TABLE 3: Statistics of the evaluation datasets.

Node Type	#Train	#Validataion	#Test	#Class
Domain name	2,827	354	353	47
IP address	3,360	420	420	23
Malware hash	1,670	209	208	15
Email address	1,215	152	152	3

GPU-based implementation of Eq. (4) using sparse-dense matrix multiplications. The computational complexity of evaluating Eq. 4 is $O(|\varepsilon| mhK)$, which scales linearly in terms of the number of graph edges denoted as $|\varepsilon|$.

5 Experiments

In this section, we conduct comprehensive experimental studies to demonstrate the effectiveness of the presented practical system *HinCTI*, which integrates the above proposed approach.

5.1 Experimental Setup

Datasets. We collect real-world data from two popular threat intelligence sharing platforms, namely, IBM X-Force Exchange Platform and VirusTotal[6], and enrich the data as described in Section 3.1. Although the collected data set involves 126,933 infrastructure nodes, only 11,340 nodes are left after preprocessing due to crawler constraints and data sparsity. Labels for 10,833 infrastructure nodes are crawled from the intelligence companies, and the remaining 507 unlabeled infrastructure nodes are labeled by three recruited security researchers with the help of third-party analysis tools. The statistics of the evaluation datasets is described in Table 3, including number of nodes for train, validation, and test and number of classes (i.e., number of threat types) for different types of infrastructure nodes.

Baselines. We compare our proposed approach with the following baselines, including state-of-the-art network representation learning methods and several traditional threat type identification methods.

- Node2Vec [37] + SVM: A random walk-based network embedding method for homogeneous graphs. Here, we use $p = 1$ and $q = 1$ and ignore the heterogeneity of nodes and perform Node2Vec on the whole heterogeneous graph.
- Metapath2Vec [38] + SVM: A heterogeneous graph embedding method, which performs meta-path-based random walk and utilizes skip-gram to embed the heterogeneous graphs. Here we test all the meta-paths and report the best performance.
- HAN [42] + SVM: A semi-supervised heterogeneous graph neural network, which considers node-level attention and semantic-level attention to learn the importance of nodes and meta-paths, respectively.
- HinCTI-: The HinCTI model that does not consider hierarchical regularization.

6. https://www.virustotal.com

TABLE 4: Metrics involved in performance evaluation of threat type identification methods.

Metrics	Description
TP_t	# of infrastructure nodes correctly classified as the t^{th} label in label set L
FP_t	# of infrastructure nodes mistakenly classified as the t^{th} label in label set L
FN_t	# of infrastructure nodes in the t^{th} label in label set L mistakenly classified
Pre	$TP/(TP + FP)$
Rec	$TP/(TP + FN)$
F_1	$2 \times Pre \times Rec \,/\, (Pre + Rec)$
$Macro\text{-}F_1$	averaged F_1 of all different labels in the hierarchy
$Micro\text{-}F_1$	a type of F_1 score considering the overall precision and recall of all labels in the hierarchy

Evaluation Metrics. To quantitatively evaluate the threat type identification performance of different methods, we follow [54] to use $Macro\text{-}F_1$ score and $Micro\text{-}F_1$ score as our evaluation metrics. The metrics involved in performance evaluation are shown in Table 4. We apply 10-fold cross-validation and report the average performance measures in terms of $Macro\text{-}F_1$ and $Micro\text{-}F_1$ scores with significance level $\alpha = 0.05$. $Macro\text{-}F_1$ is a type of F_1 score that evaluates averaged F_1 of all different labels in the hierarchy. Let TP_t, FP_t, FN_t denote the true-positives, false-positives, and false-negatives for the t^{th} label in label set L respectively. $Macro\text{-}F_1$ can be defined as:

$$\begin{aligned} Macro\text{-}F_1 &= \frac{1}{|L|} \sum_{t \in L} \frac{2 \times Precision_t \times Recall_t}{Precision_t + Recall_t}, \\ Precision_t &= \frac{TP_t}{TP_t + FP_t}, Recall_t = \frac{TP_t}{TP_t + FN_t}. \end{aligned} \tag{8}$$

$Micro\text{-}F_1$ is another type of F_1 score that considers the overall precision and recall of all labels. $Micro\text{-}F_1$ can be defined as:

$$\begin{aligned} Micro\text{-}F_1 &= \frac{2 \times Precision \times Recall}{Precision + Recall}, \\ Precision &= \frac{\sum_{t \in L} TP_t}{\sum_{t \in L} TP_t + \sum_{t \in L} FP_t}, \\ Recall &= \frac{\sum_{t \in L} TP_t}{\sum_{t \in L} TP_t + \sum_{t \in L} FN_t}. \end{aligned} \tag{9}$$

TABLE 5: Performance evaluation of different meta-paths and meta-graphs.

ID	Meta-paths included	$Macro$-F_1	$Micro$-F_1
Φ_1	/	0.7244	0.7646
Φ_2	/	0.7115	0.7594
Φ_3	/	0.7076	0.7588
Φ_4	/	0.7450	0.7764
Φ_5	/	0.7047	0.7469
Φ_6	/	0.7247	0.7682
Φ_7	/	0.7144	0.7604
Φ_8	Φ_2 & Φ_3	0.7307	0.7746
Φ_9	Φ_1 & Φ_2	0.7361	0.7764
Φ_{10}	Φ_6 & Φ_7	0.7366	0.7823
Φ_{11}	Φ_2 & Φ_6 & Φ_7	0.7451	0.7892
Φ_{12}	Φ_3 & Φ_6 & Φ_7	0.7424	0.7833

Based on the experimental setup described, we conduct experiments of threat type identification of infrastructure nodes on the operating system Ubuntu 18.04.2, Intel(R) Core(TM) i5-6600K CPU@ 3.50GHz and NVIDIA GeForce GTX 1080 Ti GPU. The software platforms are TensorFlow-gpu 1.13.1 and Python 3.7.3.

5.2 Evaluation of Different Meta-paths and Meta-graphs

In this set of experiments, based on the dataset described in Section 5.1, we evaluate the performance of different correlations among threat infrastructures depicted by different meta-graphs (i.e., Φ_1–Φ_{12}). In the experiments, given a meta-graph Φ_k, we calculate the Φ_k based on *MIIS* measure described in Section 4.3 and leverage hierarchical regularization described in Section 4.4 to learn the threat type label of the nodes with type of domain names in the HIN. The optimal identification results of different meta-graphs are presented in Table 5. Different meta-graphs show different performances in threat type identification. Each of them represents a specific semantics in the task of threat type identification.

From Table 5, we can also observe that: (1) some meta-paths, e.g., Φ_4, perform well on the testing set, whereas other meta-paths do not perform well on their own, such as Φ_5, which may be because the semantics of the meta-path cannot reflect the problem of threat type identification of infrastructure nodes well. (2) The approach based on meta-graph is generally more expressive than that based on pure meta-path in terms of depicting more complex and comprehensive relationships among nodes and thus achieve better identification performance. a) The performance of Φ_{10}, which integrates Φ_6 and Φ_7, outperforms that of both Φ_6 and Φ_7. b) The relationships among nodes depicted by the meta-graphs consisting of complicated correlations (e.g.,Φ_{10}–Φ_{12}) can provide much higher-level semantics and obtain better identification results than others (e.g., Φ_1-Φ_3). Exploring the performance when different meta-paths and meta-graphs are incorporated together for the identification is meaningful, which is evaluated in the next set of experiments.

5.3 Performance Evaluation of *HinCTI*

In this set of experiments, we evaluate our proposed approach *HinCTI* by comparisons with several typical network representation learning methods combined with the SVM algorithm (i.e., Node2Vec [37] + SVM, Metapath2Vec [38] + SVM, and HAN [42] + SVM) and HinCTI-. For node2vec, we ignore the heterogeneous property of HIN and directly feed the HIN for representation learning. For metapath2vec, we test meta-path Φ_1-Φ_7 to guide the random walks in metapath2vec and report the best performance. For node2vec and metapath2vec, which are random walk-based methods, we set window size to 5, walk length to 100, and walks per node to 500. To facilitate the comparisons, we use the experimental procedure provided in [37], [42] and implement the algorithm according to the description of [38]. For a fair comparison, the dimension of embedding is set to 64. The learned node representation vector and node features are input of SVM algorithm to identify the threat types of infrastructure nodes.

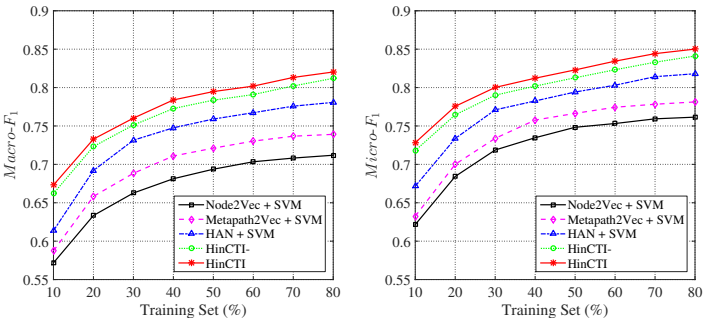

Fig. 6: Performance comparisons of different threat type identification approaches in terms of $Macro$-F_1 score (Left) and $Micro$-F_1 score (Right).

We randomly select a portion of the samples described in Section 5.1 (ranging from 10% to 80%) as the training set, 10% of samples as validation set, and the remaining 10% of samples as the testing set. Fig. 6 shows the comparison results of *HinCTI* and several typical network representation learning methods in the task of threat type identification of infrastructure nodes in terms of $Macro$-F_1 score (left) and $Micro$-F_1 score (right). On the whole, the proposed model *HinCTI* consistently and significantly outperforms all these typical network representation learning methods: an improvement of approximately 4%–11% in $Macro$-F_1 and 3%–10% in $Micro$-F_1. That is, *HinCTI* can identify the threat type of infrastructure nodes better than those of the existing state-of-the-art network representation learning methods. The success of *HinCTI* lies in the proper consideration and accommodation of the heterogeneous property of HIN (i.e., the multiple types of nodes and relations) and the advantage of meta-path and meta-graph-guided similarity computing for infrastructure nodes.

In addition, as shown in Fig. 6, we compare HinCTI (the red line) and HinCTI- (the green line). HinCTI, which considers hierarchical regularization, consistently achieves approximately 1% improvement in terms of both $Macro$-F_1 and $Micro$-F_1, which can demonstrate the effectiveness of hierarchical regularization leveraged in our system.

Furthermore, from Table 5 and Fig. 6, we observe that compared with any node representations learned based on individual meta-path or meta-graph (i.e., Φ_1–Φ_{12}), the proposed HinCTI, which efficiently incorporates different meta-paths and meta-graphs together and learns higher-level semantics of node representations, can significantly improve the performance of threat type identification of infrastructure nodes: an improvement of more than 6% in $Macro$-F_1 and $Micro$-F_1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.2987019, IEEE Transactions on Knowledge and Data Engineering

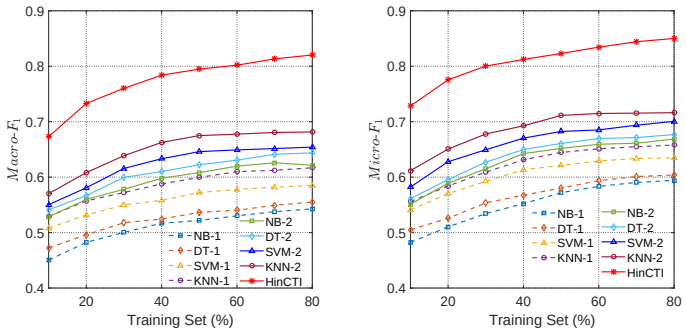

Fig. 7: Performance comparisons among *HinCTI* and traditional identification methods on $Macro\text{-}F_1$ score (Left) and $Micro\text{-}F_1$ score (Right). "NB-1", 'DT-1'", "SVM-1", and "KNN-1" represent the algorithms that take the original node features as input. "NB-2", 'DT-2'", "SVM-2", and "KNN-2" represent the algorithms that HIN-related nodes and relations are also leveraged as features for algorithms to learn.

5.4 Comparisons among *HinCTI* and Traditional Identification Methods

In this set of experiments, we compare *HinCTI* with four other typical identification methods, i.e., Naive Bayes (NB), Decision Tree(DT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). In NB-1, DT-1, SVM-1, and KNN-1, we take the original node features discussed in Section 4.1 as input. In NB-2, DT-2, SVM-2, and KNN-2, we put all HIN-related nodes and relations as features for algorithms to learn. All the algorithms are implemented in Python and trained and executed with best parameter values. For SVM, we use GridSearchCV in sklearn to obtain the best combination of parameters.

The experimental results are shown in Fig. 7. The proposed HinCTI significantly outperforms all these traditional identification methods. Compared to the performance results of NB-1, NB-2 achieves roughly 7%–9% improvements in $Macro\text{-}F_1$ and $Micro\text{-}F_1$. Compared to the performance results of DT-1, DT-2 achieves around 6%–8% improvements in $Macro\text{-}F_1$ and 5%–8% in $Micro\text{-}F_1$. Compared to the performance results of SVM-1, SVM-2 achieves approximately 5%–7% improvements in $Macro\text{-}F_1$ and 4%–7% in $Micro\text{-}F_1$. Similarly, compared to the performance results of KNN-1, KNN-2 achieves nearly 4%–6% improvements in $Macro\text{-}F_1$ and 6%–7% in $Micro\text{-}F_1$. That is, HIN-related nodes and relations leveraged by machine learning methods can help improve the performance of threat type identification of infrastructure nodes, which demonstrates that rich semantics encoded in different types of relations can bring more information.

Moreover, compared to the performance results of NB-2, DT-2, SVM-2, and KNN-2, *HinCTI* achieves approximately 10%–20% improvements in $Macro\text{-}F_1$ and 11%–19% in $Micro\text{-}F_1$. *HinCTI* is significantly better than the best baseline methods we compared. The reason is that the inputs of traditional identification algorithms are simply flat features, i.e., the simple combination of different features. By contrast, in *HinCTI*, we design the expressive representation and build the connection between the higher-level semantics of the infrastructure node data and their threat type labels. To identify the threat types of the increasingly sophisticated threat infrastructures, *HinCTI* using meta-graph based approach over HIN can build the higher-level semantic and structural connection between threat infrastructures with a more expressive and comprehensive view and thus achieves better identification performance.

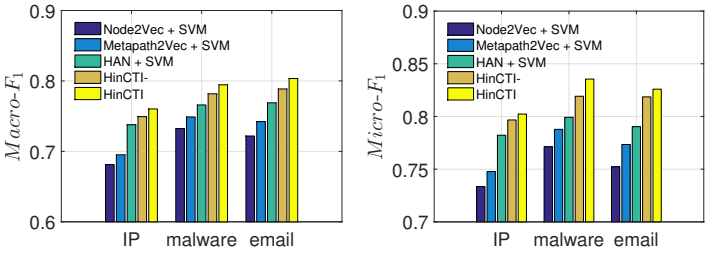

Fig. 8: Performance of *HinCTI* for other types of infrastructure nodes on $Macro\text{-}F_1$ score (Left) and $Micro\text{-}F_1$ score (Right).

5.5 Performance of *HinCTI* for Other Types of Nodes

The proposed approach is relatively general in threat type identification and can be applied to nodes of domain names and other types of nodes. For the threat type identification of other infrastructure nodes considered in this research (i.e., IP address, malware hash, and email address), Fig. 8 shows preliminary results of *HinCTI*. On the whole, the proposed approach *HinCTI* consistently outperforms all other typical methods and achieves approximately 6%–8% improvements in $Macro\text{-}F_1$ and 6%–7% in $Micro\text{-}F_1$.

5.6 Discussions and Limitations

A large amount of structured CTI can be first collected, and then the proposed approach can be leveraged to extract diverse semantic information. This is significant not only for threat type identification of threat infrastructure nodes but also for the mining of CTI, as demonstrated by our measurement study. Actually, our current design is still preliminary, and we discuss its limitations here. In this research, considering the limitations of data acquisition, only four types of infrastructure nodes and five types of relations are considered explicitly. However, our model is extensible, in which more types of nodes and relations can be introduced to produce higher-level semantics, such as organizations, domain owners, techniques and tools utilized to achieve the attack, occurrence time and locations of the attacks incidents, and relations among them. Moreover, we have not considered the dynamic nature of infrastructure nodes' threat type, that is, we only process the latest threat type of infrastructure nodes in this research. However, ignoring infrastructure nodes' history threat types affects the performance of identification.

6 Conclusions and Future Work

In this paper, we propose a CTI modeling and threat type identification system based on HIN, called *HinCTI*. We design meta-schema and a set of meta-paths and meta-graphs to model CTI on HIN, which can extract and incorporate higher-level semantics of cyber-threat infrastructure nodes involved in CTI. Through the proposed *MIIS* measure-based heterogeneous GCN-based threat type identification approach, we overcome the challenge of limited labels of cyber-threat infrastructure nodes. Through the hierarchical regularization, our identification approach can also alleviate the problem of overfitting. Experiments based on real-world dataset demonstrate that our

developed system *HinCTI* that integrates our proposed approach can significantly improve the performance of threat type identification compared with the existing state-of-the-art baseline methods.

For future work, we plan to explore other information to enrich the node features and relations of the cyber threat intelligence HIN for further improving the performance of our approach. Another interesting direction for future work is the extraction of fine-grained structured data (including node and their relationships) from intelligence reports recorded in natural language, leveraging topic modeling and natural language processing techniques. Doing so will greatly enrich the heterogeneous information network and enhance the performance of threat identification.

ACKNOWLEDGMENTS

We are especially grateful to the anonymous reviewers and editors for their valuable comments and suggestions that help improve the quality of this manuscript. This work was supported in part by the NSFC-General Technology Fundamental Research Joint Fund under grant U1836215 and BUPT Excellent Ph.D. Students Foundation under grant CX2018216. Philip S. Yu was supported by US National Science Foundation under grants III-1763325, III-1909323, and CNS-1930941. Hao Peng was supported by the National Key R&D Program of China under grant 2018YFC0830804.

REFERENCES

[1] S. Samtani, M. Abate, V. Benjamin, and W. Li, *Cybersecurity as an Industry: A Cyber Threat Intelligence Perspective*, pp. 1–20. Cham: Springer International Publishing, 2019.

[2] R. McMillan, "Definition: threat intelligence." https://www.gartner.com/doc/2487216/definition-threat-intelligence, 2013. Retrieved January, 2019.

[3] D. Bianco, "The Pyramid of Pain." http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html, 2013.

[4] A. Modi, Z. Sun, A. Panwar, T. Khairnar, Z. Zhao, A. Doupé, G.-J. Ahn, and P. Black, "Towards automated threat intelligence fusion," in *IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pp. 408–416, IEEE, 2016.

[5] A. Boukhtouta, D. Mouheb, M. Debbabi, O. Alfandi, F. Iqbal, and M. El Barachi, "Graph-theoretic characterization of cyber-threat infrastructures," *Digital Investigation*, vol. 14, pp. S3–S15, 2015.

[6] C. Sillaber, C. Sauerwein, A. Mussmann, and R. Breu, "Data quality challenges and future research directions in threat intelligence sharing practice," in *Workshop on Information Sharing and Collaborative Security*, pp. 65–70, ACM, 2016.

[7] S. Lee, H. Cho, N. Kim, B. Kim, and J. Park, "Managing cyber threat intelligence in a graph database: Methods of analyzing intrusion sets, threat actors, and campaigns," in *International Conference on Platform Technology and Service (PlatCon)*, pp. 1–6, IEEE, 2018.

[8] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 755–766, ACM, 2016.

[9] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 103–115, ACM, 2017.

[10] F. Böhm, F. Menges, and G. Pernul, "Graph-based visual analytics for cyber threat intelligence," *Cybersecurity*, vol. 1, no. 1, p. 16, 2018.

[11] U. Noor, Z. Anwar, A. W. Malik, S. Khan, and S. Saleem, "A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories," *Future Generation Computer Systems*, 2019.

[12] 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.

[13] P. K. Manadhata, S. Yadav, P. Rao, and W. Horne, "Detecting malicious domains via graph inference," in *European Symposium on Research in Computer Security*, pp. 1–18, Springer, 2014.

[14] X. Kong, B. Cao, and P. S. Yu, "Multi-label classification by mining label and instance correlations from heterogeneous information networks," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622, ACM, 2013.

[15] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1298–1306, ACM, 2011.

[16] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & Security*, 2017.

[17] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information eXpression (STIX^{TM})," *MITRE Corporation*, vol. 11, pp. 1–22, 2012.

[18] R. Danyliw, J. Meijer, and Y. Demchenko, "The incident object description exchange format," tech. rep., 2007.

[19] Mandiant, "Sophisticated indicators for the modern threat landscape: An introduction to OpenIOC." Technical report, Mandiant Whitepaper, 2013.

[20] T. Yadav and A. M. Rao, "Technical aspects of cyber kill chain," in *International Symposium on Security in Computing and Communication*, pp. 438–452, Springer, 2015.

[21] H. Gascon, B. Grobauer, T. Schreck, L. Rist, D. Arp, and K. Rieck, "Mining attributed graphs for threat intelligence," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 15–22, ACM, 2017.

[22] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "Fraudar: Bounding graph fraud in the face of camouflage," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 895–904, ACM, 2016.

[23] Y. Shi, G. Chen, and J. Li, "Malicious domain name detection based on extreme machine learning," *Neural Processing Letters*, vol. 48, no. 3, pp. 1347–1357, 2018.

[24] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall, "Developing an ontology for cyber security knowledge graphs," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, pp. 1–4, 2015.

[25] S. Noel, E. Harley, K. Tam, M. Limiero, and M. Share, "CyGraph: graph-based analytics and visualization for cybersecurity," in *Handbook of Statistics*, vol. 35, pp. 117–167, Elsevier, 2016.

[26] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering*, vol. 4, no. 1, pp. 53–60, 2018.

[27] Y. Gao, X. Li, J. Li, Y. Gao, and N. Guo, "Graph mining-based trust evaluation mechanism with multidimensional features for large-scale heterogeneous threat intelligence," in *IEEE International Conference on Big Data*, pp. 1–6, IEEE, 2018.

[28] T.-H. Chen, S. W. Thomas, and A. E. Hassan, "A survey on the use of topic models when mining software repositories," *Empirical Software Engineering*, vol. 21, no. 5, pp. 1843–1919, 2016.

[29] H. Azarbonyad, M. Dehghani, T. Kenter, M. Marx, J. Kamps, and M. De Rijke, "HiTR: Hierarchical topic model re-estimation for measuring topical diversity of documents," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[30] S. Samtani, R. Chinn, and H. Chen, "Exploring hacker assets in underground forums," in *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 31–36, IEEE, 2015.

[31] S. Samtani, K. Chinn, C. Larson, and H. Chen, "AZSecure hacker assets portal: cyber threat intelligence and malware analysis," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pp. 19–24, IEEE, 2016.

[32] S. Samtani, R. Chinn, H. Chen, and J. F. Nunamaker Jr, "Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence," *Journal of Management Information Systems*, vol. 34, no. 4, pp. 1023–1053, 2017.

[33] 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, pp. 1–16, 2011.

[34] 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 *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pp. 491–506, 2012.

[35] K. Pei, Z. Gu, B. Saltaformaggio, S. Ma, F. Wang, Z. Zhang, L. Si, X. Zhang, and D. Xu, "HERCULE: Attack story reconstruction via community discovery on correlated log graph," in *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ACSAC '16, (New York, NY, USA), pp. 583–595, ACM, 2016.

[36] 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*, pp. 701–710, ACM, 2014.

[37] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, ACM, 2016.

[38] 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*, pp. 135–144, ACM, 2017.

[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*, pp. 1797–1806, ACM, 2017.

[40] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 729–734, IEEE, 2005.

[41] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[42] X. Wang, H. Ji, C. Shi, B. Wang, C. Peng, P. S. Yu, and Y. Ye, "Heterogeneous graph attention network," in *Proceedings of the 26th international conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2019.

[43] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1531–1540, ACM, 2018.

[44] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, vol. 19, pp. 3238–3245, 2019.

[45] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, "HinDroid: An intelligent android malware detection system based on structured heterogeneous information network," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1507–1515, ACM, 2017.

[46] Y. Fan, S. Hou, Y. Zhang, Y. Ye, and M. Abdulhayoglu, "Gotcha-sly malware!: Scorpion a metagraph2vec based malware detection system," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 253–262, ACM, 2018.

[47] 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.

[48] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, pp. 1744–1772, Secondquarter 2019.

[49] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1245–1254, ACM, 2009.

[50] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1663–1677, 2012.

[51] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 635–644, ACM, 2017.

[52] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[54] S. Gopal and Y. Yang, "Recursive regularization for large-scale classification with hierarchical and graphical dependencies," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 257–265, ACM, 2013.

[55] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-CNN," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 1063–1072, International World Wide Web Conferences Steering Committee, 2018.

[56] D. B. Wagner, "Dynamic programming," *The Mathematica Journal*, vol. 5, no. 4, pp. 42–51, 1995.

[57] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

Yali Gao is currently pursuing the Ph.D. degree with the Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and Telecommunications, China. She has published some papers in journals and conference proceedings. Her current research interests include social network, representation learning, and distributed computing and trusted services.

Xiaoyong Li received the Ph.D. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2009. He is currently a professor of computer science at Beijing University of Posts and Telecommunications (BUPT). In 2009, he was awarded an Outstanding Doctoral Graduate in Shaanxi Province, China. In 2012, he was awarded a New Century Excellent Talent in University, China. In 2015, he won the IET Information Security Premium Award. His current research interests include cloud computing, network security and trusted systems. As the first author, he has published more than 60 papers in journals and conference proceedings. He has obtained five patents and five software copyrights in network security, cloud computing, and other fields.

Hao Peng is currently an Assistant Professor at the School of Cyber Science and Technology, and Beijing Advanced Innovation Center for Big Data and Brain Computing in Beihang University. His research interests include representation learning, urban computing and text mining.

Binxing Fang received his Ph.D. from Harbin Institute of Technology, China in 1989. He is a member of the Chinese Academy of Engineering and a professor in School of Cyberspace Security at Beijing University of Posts and Telecommunications. He is currently the chief scientist of State Key Development Program of Basic Research of China. His current interests include big data and information security.

Philip S. Yu received the PhD degree in electrical engineering from Stanford University. He is a distinguished professor in computer science with the University of Illinois, Chicago, and is the Wexler chair in Information Technology. His research interests include big data, data mining, data stream, database, and privacy. He was the editor-in-chief of the IEEE Transactions on Knowledge and Data Engineering and the ACM Transactions on Knowledge Discovery from Data. He received the ACM SIGKDD 2016 Innovation Award, a Research Contributions Award from the IEEE International Conference on Data Mining (2003), and a Technical Achievement Award from the IEEE Computer Society (2013). He is a fellow of the IEEE and ACM.