Cyber Threat Intelligence Modeling Based on Heterogeneous Graph Convolutional Network

Jun Zhao[1,2], Qiben Yan[3,*], Xudong Liu[1,2,*], Bo Li[1,2,*], Guangsheng Zuo[1,2]
[1] *School of Computer Science and Engineering, Beihang University, Beijing, China*
[2] *Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China*
[3] *Computer Science and Engineering, Michigan State University, East Lansing, Michigan, USA*

Abstract

Cyber Threat Intelligence (CTI), as a collection of threat information, has been widely used in industry to defend against prevalent cyber attacks. CTI is commonly represented as Indicator of Compromise (IOC) for formalizing threat actors. However, current CTI studies pose three major limitations: first, the accuracy of IOC extraction is low; second, isolated IOC hardly depicts the comprehensive landscape of threat events; third, the interdependent relationships among heterogeneous IOCs, which can be leveraged to mine deep security insights, are unexplored. In this paper, we propose a novel CTI framework, HINTI, to model the interdependent relationships among heterogeneous IOCs to quantify their relevance. Specifically, we first propose multi-granular attention based IOC recognition method to boost the accuracy of IOC extraction. We then model the interdependent relationships among IOCs using a newly constructed heterogeneous information network (HIN). To explore intricate security knowledge, we propose a threat intelligence computing framework based on graph convolutional networks for effective knowledge discovery. Experimental results demonstrate that our proposed IOC extraction approach outperforms existing state-of-the-art methods, and HINTI can model and quantify the underlying relationships among heterogeneous IOCs, shedding new light on the evolving threat landscape.

1 Introduction

Nowadays, we are witnessing a rapid growth of sophisticated cyber attacks (e.g., zero-day attack, advanced persistent threat) [34]. Such attacks can effortlessly bypass traditional defenses such as firewalls and intrusion detection systems (*IDS*), breach critical infrastructures, and cause devastating catastrophes [7, 20, 39]. To combat these emerging threats, security experts proposed Cyber Threat Intelligence (CTI) that consists of a collection of Indicators of Compromise (IOCs). Different from the well-known security databases (e.g., CVE[1], ExploitDB[2]), CTI can facilitate organizations to proactively release more comprehensive and valuable threat warnings (e.g., malicious IPs, malicious DNS, malware and attack patterns, etc.) when a system encounters suspicious outsider or insider threats [23].

In recent years, CTI has been increasingly adopted by security researchers and industries to share and capitalize on their discoveries, as well as by security firms to analyze the threat landscape using the deluge of data [5, 30]. The original CTI extraction and analysis require extensive manual inspection of the attack event descriptions, which becomes rather time-consuming given the enormous volume of threat description data. Recent studies have proposed automated methods to extract CTI in the form of Indicator of Compromise (IOC) from unstructured security-related texts [4, 22]. Most of existing IOC extraction methods, such as *CleanMX*[3], *PhishTank*[4], *IOC Finder*[5], and *Gartner peer insight*[6], follow the OpenIOC [10] standard and extract particular types of IOCs (e.g., malicious IP, malware, file Hash, etc) by leveraging a set of regular expressions. However, such extraction approaches face three major limitations. First, the accuracy of IOC extraction is low, which inevitably leads to the omission of critical threat objects [22]. Second, isolated IOC hardly depicts the comprehensive landscape of threat events, making it virtually impossible for CTI subscribers to gain a complete picture into the incoming threat. Third, there is a lack of an effective computing framework to efficiently measure the interactive relationships among heterogeneous IOCs.

To combat these limitations, HINTI, a cyber threat intelligence framework based on heterogeneous information network (HIN), is proposed to model and analyze CTIs. Specifically, HINTI proposes a multi-granular attention based IOC recognition approach to boost the accuracy of IOC extraction.

[1] http://cve.mitre.org/
[2] https://www.exploit-db.com/
[3] http://list.clean-mx.com
[4] https://www.phishtank.com
[5] https://www.fireeye.com/services/freeware/ioc-finder.html
[6] https://www.gartner.com/reviews/market/security-threat-intelligence-services

Then, HINTI leverages HIN to model the interdependent relationships among heterogeneous IOCs, which can depict a more comprehensive picture of threat events. Moreover, we propose a novel CTI computing framework to quantify the interdependent relationships among IOCs, which helps uncover novel security insights. In short, the main contributions of this paper are summarized as follows:

- **Multi-granular Attention based IOC Recognition**. We propose multi-granular attention based IOC recognition approach to automatically extract cyber threat objects from multi-source threat texts, which can learn the significance of features with different scales. Our model outperforms the state-of-the-art methods in terms of IOC recognition accuracy and recall. In total, we extract over 397,730 IOCs from the unstructured threat descriptions.
- **Heterogeneous Threat Intelligence Modeling**. We model different types of IOCs using heterogeneous information network (HIN), which introduces various meta-paths to capture the interdependent relationships among heterogeneous IOCs while depicting a more comprehensive landscape of cyber threat events.
- **Threat Intelligence Computing Framework**. *We are the first* to present the concept of *cyber threat intelligence computing*, and design a general computing framework, as shown in Figure 5. The framework first utilizes a weight-learning based node similarity measure to quantify the interdependent relationships between heterogeneous IOCs, and then leverages attention mechanism based heterogeneous graph convolutional networks to embed the IOCs and their interactive relations.
- **Threat Intelligence Prototype System**. To evaluate the effectiveness of HINTI, we implement a CTI prototype system. Our system has identified 1,262,258 relationships among 6 types of IOCs including attackers, vulnerabilities, malicious files, attack types, devices and platforms, based on which we further assess the real-world applicability of HINTI using three real-world applications: IOC significance ranking, attack preference modeling, and vulnerability similarity analysis.

2 Background

2.1 Cyber Threat Intelligence

Cyber Threat Intelligence (CTI) extracted from security-related data is structured information used to proactively resist cyber attacks. CTI consists of reasoning, context, mechanism, indicators, implications, and actionable advice about an existing or evolving cyber attack that can be used to create preventive measures in advance [30]. CTI allows subscribers to expand their visibility into the fast-growing threat landscape, and enable early identification and prevention of a cyber threat. Take WannaCry virus as an example, if security guards can timely capture the threat intelligence that indicates "Wannacry permeates port 445 to attack systems", the malicious intrusion can be easily blocked by locking down port 445, which is the most direct and effective way of combating WannaCry virus [7].

Meanwhile, social media (e.g., Blog, Twitter) has increasingly become an effective medium for exchanging and spreading cyber security information, on which security experts and organizations often post their discoveries to reach a wider audience promptly [32]. These posts usually include a magnitude of valuable security-related information [25, 26], such as the warnings regarding latest vulnerabilities, hacking tools, data breaches, and existing or upcoming software patches, providing one of the main raw materials for extracting CTIs.

Early CTI extraction requires extensive manual inspection of the threat descriptions, which becomes rather time-consuming given the enormous volume of such descriptions. To facilitate the automatic generation and sharing of CTI, a large volume of methods and frameworks are established, such as *IODEF* [13], *STIX* [4], *TAXII* [36], *OpenIOC* [10], and *CyBox* [28], CleanMX, PhishTank, IOC Finder and [2,22,31,46]. The majority of existing methods and frameworks leverage regular expressions to extract IOCs, which may suffer from a low accuracy due to their inability in predefining a comprehensive set of the IOC rules.

2.2 Motivation

The main goal of this research is to address the limitations of the existing CTI analytics frameworks by modeling the interdependent relationships among heterogeneous IOCs. As a motivating example, given a security-related post: *"Last week, Lotus exploited CVE-2017-0143 vulnerability to affect a larger number of Vista SP2 and Win7 SP devices in Iran. CVE-2017-0143 is a remote code execution vulnerability including a malicious file SMB.bat"*. Most of the existing CTI frameworks can extract specific IOCs but neglect the relationships among them, as shown in Figure 1. It is obvious that such IOCs could not draw a comprehensive picture of the threat landscape, let alone quantifying their interactive relationships for in-depth security investigation.

Different from the existing CTI frameworks, HINTI aims to implement a computational CTI framework, which can not only extract IOCs efficiently but also model and quantify the relationships between them. Here, we use the motivating example to illustrate how HINTI works step-by-step in practice as follows.

(i) First, the security-related post is annotated by the *B-I-O* sequence tagging method [43] as shown in Figure 2, where *B-X* indicates that the element of type *X* is located at the beginning of the fragment, *I-X* means that the element belonging to type *X* is located in the middle of the fragment, and *O* represents a non-essential element of other types. In this

```
<? Xml version=1.0 encoding=utf-8>
  <indicator id=1a0ee12, op=OR>
    <Description>
      <Actor>Lotus</Actor>
      <Vul>CVE-2017-0143</Vul>
      <Dev>Vista SP2</Dev>
      <Dev>Win7 SP1</Dev>
      <Type>Remote code execution</Type>
      <File>SMB.bat</File>
    </Description>
```

Figure 1: An example of extracted IOCs without any relations among them.

research, we annotated 30,000 such training samples from 5,000 threat description texts, which are the raw materials used to build our IOC extraction model.

Last$_{(O)}$ week$_{(O)}$, Lotus$_{(B\text{-}Attacker)}$ exploited$_{(O)}$ CVE-2017-0143$_{(B\text{-}Vul)}$ vulnerability$_{(O)}$ to$_{(O)}$ affect$_{(O)}$ a$_{(O)}$ large$_{(O)}$ number$_{(O)}$ of$_{(O)}$ Vista SP2$_{(B\text{-}Device)}$ and$_{(O)}$ Win7 SP1$_{(B\text{-}Device)}$ devices$_{(O)}$ in$_{(O)}$ Iran$_{(O)}$. CVE-2017-0143$_{(B\text{-}Vul)}$ is$_{(O)}$ a$_{(O)}$ remote$_{(B\text{-}Type)}$ code$_{(I\text{-}Type)}$ execution$_{(I\text{-}Type)}$ vulnerability$_{(O)}$ involving$_{(O)}$ a$_{(O)}$ malicious$_{(O)}$ file$_{(O)}$ SMB.bat$_{(B\text{-}File)}$.

Figure 2: An annotation example with the *B-I-O* tagging method.

(ii) The labeled training samples are then fed into the proposed neural network architecture as shown in Figure 6 to train our proposed IOC extraction model. As a result, HINTI has the ability to accurately identify and extract IOCs (e.g., *Lotus*, *SMB.bat*) using the proposed multi-granular attention based IOC extraction method (**see Section 4.1 for details**).

(iii) HINTI then utilizes the syntactic dependency parser [6] (e.g., subject-predicate-object, attributive clause, etc.) to extract associated relationships between IOCs, each of which is represented as a triple $(IOC_i, relation, IOC_j)$. In this motivating example, HINTI extracts the relationship triples involving $(Lotus, exploit, CVE-2017-0143)$, $(CVE-2017-0143, affect, VistaSP2)$, etc. Note that the extracted relational triples can be incrementally pooled into an HIN to model the interactions among IOCs for depicting a more comprehensive threat landscape. Figure 3 shows a miniature graphic representation describing interactive relations among IOCs extracted from the example. Compared with Figure 1, it is obvious that HINTI can depict a more intuitive and comprehensive threat landscape than the previous approaches. In this paper, we mainly consider 9 relationships (**R1**∼**R9**) among 6 different types of IOCs (**see Section 4.2 for details**).

(iv) Finally, HINTI integrates a CTI computing framework

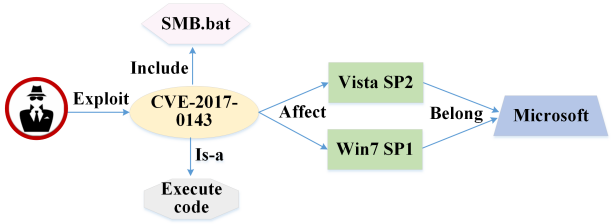


Figure 3: A miniature of a constructed CTI includes attacker, vulnerability, malicious file, attack type, device, and platform, which describes a particular threat: an attacker utilizes *CVE-2017-0143* vulnerability to invade *Vista SP2* and *Win7 SP1* devices. *CVE-2017-0143* is a *remote code execution* vulnerability that involves a malicious file *"SMB.bat"*.

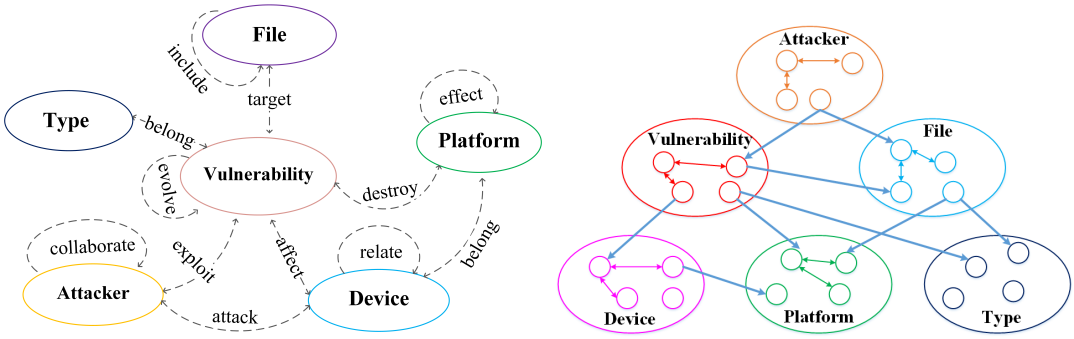
based on heterogeneous graph convolutional networks (**see Section 4.3**) to effectively quantify the relationships among IOCs for knowledge discovery. Particularly, our proposed CTI computing framework characterizes IOCs and their relationships in a low-dimensional embedding space, based on which CTI subscribers can use any classification (e.g., *SVM*, *Naive Bayes*) or clustering algorithms (*K-Means*, *DBSCAN*) to gain new threat insights, such as predicting which attackers are likely to intrude their systems, and identifying which vulnerabilities belong to the same category without the expert knowledge. In this work, we mainly explore three real-world applications to verify the effectiveness and efficiency of the CTI computing framework: IOC significance ranking (**see Section 6.1**), attack preference modeling (**see Section 6.2**), and vulnerability similarity analysis (**see Section 6.3**).

2.3 Preliminaries

In this paper, we use heterogeneous information network (HIN) to model the relationships among IOCs. Here, we first introduce the preliminary knowledge about HIN.

Definition 1 Heterogeneous Information Network of Threat Intelligence (HINTI) *is defined as a directed graph* $G = (V, E, T)$ *with an object type mapping function* $\varphi : V \rightarrow M$ *and a link type mapping function* $\Psi : E \rightarrow R$. *Each object* $v \in V$ *belongs to one particular object type in the object type set* M: $\varphi(v) \in M$, *and each link* $e \in E$ *belongs to a particular relation type in the relation type set* R: $\Psi(e) \in R$. T *denotes the types of nodes and relationships.*

In this paper, we focus on 6 common types of IOCs: *attacker (A), vulnerability (V), device (D), platform (P), malicious file (F), and attack type (T)*, and the links connecting different objects represent different semantic relationships. To better understand the object types and relationship types in HINTI, it is imperative to provide the meta-level (i.e., schema-level) description of the network. Consequently, we introduce



(a) Network schema. (b) Network instance.

Figure 4: Network schema and instance of HIN containing 6 types of IOCs. (a): The network schema of HIN, which depicts the relationship template among different types of IOCs, such as $Device \xrightarrow{belong} Platform$. (b): An instance of network schema, which describes the concrete relationships between IOCs by following a network schema, e.g., $Office\ 2012 \xrightarrow{belong} Windows$.

the network schema [37] for describing the meta-level relationships.

Definition 2 Network Schema. *The network schema of HINTI, denoted as $H_S = (A, R)$, is a meta template for $G = (V, E, T)$ with the object type mapping $\varphi : V \to M$ and the link type mapping $\Phi : E \to R$. It is a directed graph of object types M with edges representing relations from R.*

The schema of HINTI specifies type constraints on the sets of IOCs and their relationships. Figure 4 (a) shows the network schema of HINTI, which defines the relationship templates among IOCs to effectively guide the semantic exploration in HINTI. For example, for a relationship that describes: "*attackers invade devices*", the semantic schema can be written as: $attacker \xrightarrow{invade} device$. Figure 4 (b) presents a concrete instance of the network schema.

Definition 3 Meta-path. *A meta-path [37] P is a path sequence defined on a network schema $S = (N, R)$, and is represented in the form of $N_1 \xrightarrow{R_1} N_2 \xrightarrow{R_2} \cdots \xrightarrow{R_i} N_{i+1}$, which defines a composite relation $R = R_1 \diamond R_2 \diamond \cdots \diamond R_{i+1}$, where $\diamond$ denotes the composition operator on relations. A meta-path P is a symmetric path when the relation R defined by the path is symmetric (i.e, P is equal to P^{-1}).*

Table 1 displays the meta-paths considered in HINTI. For example, the relationship *"the attackers (A) exploit the same vulnerability (V)"* can be described by a length-2 meta-path $attacker \xrightarrow{exploit} vulnerability \xrightarrow{exploit^-} attacker$, denoted as AVA^T (P_4), which means that the two attackers exploit the same vulnerability. Similarly, $AVDPD^TV^TA^T$ (P_{17}) portrays a close relationship between IOCs that "*two attackers who leverage the same vulnerability invade the same type of device and ultimately destroy the same type of platform*".

Table 1: Meta-paths used in HINTI.

ID	Meta-path
P_1	Attacker-Attacker
P_2	Device-Device
P_3	Vulnerability-Vulnerability
P_4	Attacker-Vulnerability-Attacker
P_5	Attacker-Device-Attacker
P_6	Device-File-Device
P_7	Device-Platform-Device
P_8	Vulnerability-File-Vulnerability
P_9	Vulnerability-Type-Vulnerability
P_{10}	Vulnerability-Device-Vulnerability
P_{11}	Vulnerability-Platform-Vulnerability
P_{12}	Attacker-Device-Platform-Device-Attacker
P_{13}	Attacker-Vul-Device-Vul-Attacker
P_{14}	Attacker-Vul-Platform-Vul-Attacker
P_{15}	Attacker-Vul-Type-Vul-Attacker
P_{16}	Vul-Device-Platform-Device-Vul
P_{17}	Attacker-Vul-Device-Platform-Device-Vul-Attacker

3 Architecture Overview of HINTI

HINTI, as a cyber threat intelligence extraction and computing framework, is capable of effectively extracting IOCs from threat-related descriptions and formalizing the relationships among heterogeneous IOCs to demystify new threat insights. As shown in Figure 5, HINTI consists of four major components, including:

- ***Data Collection and IOC Recognition***. We first de-

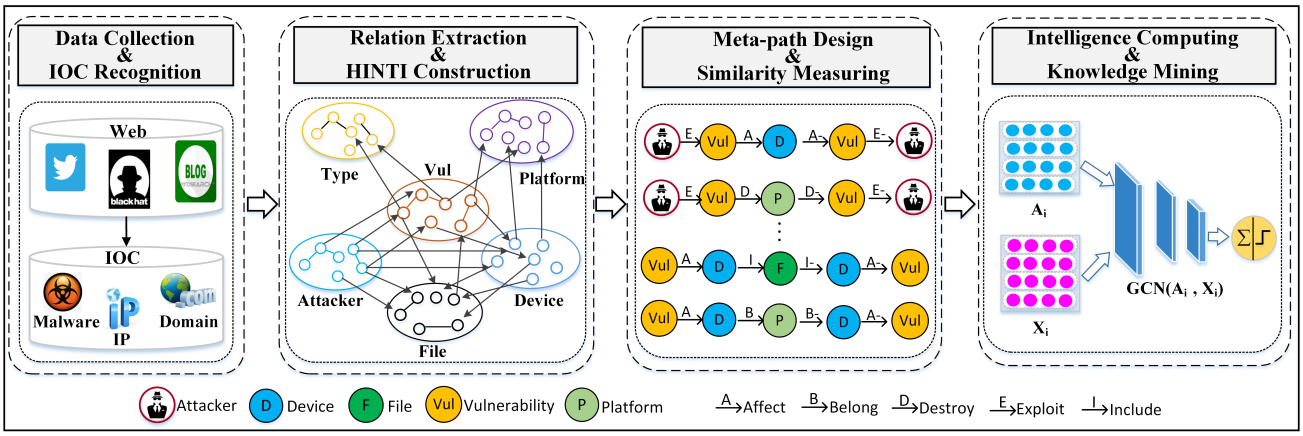


Figure 5: The overall architecture of HINTI. HINTI consists of four major components: (a) collecting security-related data and extracting threat objects (i.e., IOCs); (b) modeling interdependent relationships among IOCs into a heterogeneous information network; (c) embedding nodes into a low-dimensional vector space using weight-learning based similarity measure; and (d) computing threat intelligence based on graph convolutional networks and knowledge mining.

velop a data collection system to automatically capture security-related data from blogs, hacker forum posts, security news, and security bulletins. The system utilizes a breadth-first search to collect the HTML source code, and then leverages Xpath (XML Path language) to extract threat-related descriptions. After that, we utilize a multi-granular attention based IOC recognition method to extract IOC from the collected threat-related descriptions (**see Section 4.1 for details**).

- ***Relation Extraction and IOC modeling***. HINTI addresses the challenge of CTI modeling by leveraging heterogeneous information networks, which can naturally depict the interdependent relationships between heterogeneous IOCs. As an example, Figure 4 shows a model that capture the interactive relationships among attacker, vulnerability, malicious file, attack type, platform, and device (**see Section 4.2 for details**).

- ***Meta-path Design and Similarity Measure***. Meta-path is an effective tool to express the semantic relations among IOCs in constructed HIN. For instance, $attacker \xrightarrow{exploit} vulnerability \xrightarrow{exploit^{-}} attacker$, indicates that two attackers are related by exploiting the same vulnerability. We design 17 types of meta-paths (**See Table 1**) to describe the interdependent relationships between IOCs. With these meta-paths, we present a weight-learning based node similarity computing approach to quantify and embed the relationships as the premise for threat intelligence computing.

- ***Threat Computing and Knowledge Mining***. In this component, an effective threat intelligence computing framework is proposed, which can quantify and measure the relevance among IOCs by leveraging graph convolutional network (GCN). Our proposed threat intelligence computing framework could reveal richer security knowledge within a more comprehensive threat landscape.

4 Methodology

4.1 Multi-granular Attention Based IOC Extraction

Extracting IOCs from multi-source threat texts is one of the major tasks of threat intelligence analytics, and the quality of the extracted IOCs significantly influences the analysis results of cyber threats. Recently, Bidirectional Long Short-Term Memory+Conditional Random Fields (BiLSTM+CRF) model [15] has demonstrated excellent performance in text chunking and Named-entity Recognition (NER). However, directly applying this model to IOC extraction is unlikely to succeed, since threat texts usually contain a large number of threat objects with different grams and irregular structures. Consequently, we need an efficient method to learn the discriminative characteristics of IOCs with different sizes. In this paper, we propose a multi-granular attention based IOC extraction method, which can extract threat objects with different granularity. Particularly, Figure 6 presents the proposed IOC extraction framework, which leverages the multi-granular attention mechanism to characterize IOCs. Different from the traditional BiLSTM+CRF model, we introduce new word-embedding features with different granularities to capture the characteristics of IOCs with different sizes. Furthermore, we utilize a self-attention mechanism to learn the importance of the features to improve the accuracy of IOC extraction.

Our proposed method takes a threat description sentence $X = (x_1, x_2, \cdots, x_i)$ as input, where x_i represents *i-th* word

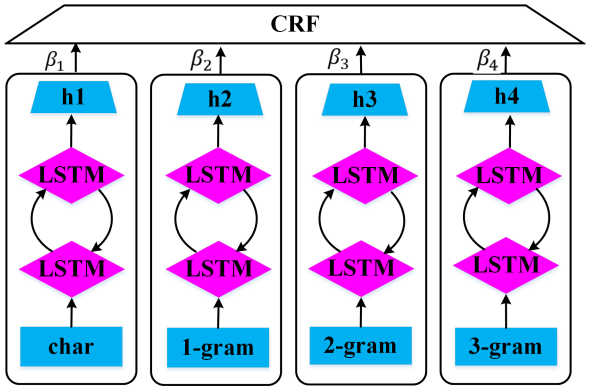


Figure 6: The framework of multi-granular IOC extraction.

in X. We first chunk the sentence into n-gram components including char-level, 1-gram, 2-gram, and 3-gram, which are the inputs of our trained model, written as follows:

$$e^j_{x_i} = V^j_{embedding}(x_i), \tag{1}$$

where $V^j_{embedding}$ transforms the chunk with granularity j into a vector space and x_i is the i-th word in a sentence X. Thus, the threat description sentence X_i can be vectorized as follows:

$$\begin{aligned} \overrightarrow{h^j_i} &= LSTM_{forward}([e^j_{x_0}, e^j_{x_1}, \cdots, e^j_{x_i}]) \\ \overleftarrow{h^j_i} &= LSTM_{backward}([e^j_{x_0}, e^j_{x_1}, \cdots, e^j_{x_i}]) \end{aligned} \tag{2}$$

where $\overrightarrow{h^j_i}$ and $\overleftarrow{h^j_i}$ are the embedded features learned by forward LSTM and backward LSTM, respectively. Let O be the output of Bi-LSTM, which is a weighted sum of embedded features with weights corresponding to the importance of different features:

$$O = H \cdot W^T \tag{3}$$

where $H = \sum^j \vec{\beta}_i \sigma(h^j_1, h^j_2, \cdots, h^j_i)$, $h^j_i = (\overrightarrow{h^j_i} + \overleftarrow{h^j_i})$, $\vec{\beta}_i$ is the weight vector to represent the importance of h^j_i, in which j and i are the segmentation granularity of sentences and the corresponding index of the chunk. W is the parameter matrix.

Given a security-related sentence $X = (x_1, x_2, \cdots, x_i)$, its corresponding threat object sequence $Y = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_i)$, and its output of Bi-LSTM O, we can compute the overall label score of X and Y as follows:

$$S(X,Y) = \sum_{i=0}^{n} (A_{\hat{y}_i, \hat{y}_{i+1}} + O_{i,\hat{y}_i}) \tag{4}$$

where $A_{\hat{y}_i, \hat{y}_{i+1}}$ is the state transition matrix in CRF model, and $O_{i,\hat{y}_i}$, as the output of Bi-LSTM hidden layer (calculated by Eq. (3)), represents the label score of i-th word corresponding to the type $\hat{y}_i$. Next, we utilize $softmax$ function to normalize the overall label score:

$$p(Y|X) = \frac{e^{S(X,Y)}}{\sum_{\tilde{y} \in Y_X} e^{S(X,Y)}} \tag{5}$$

We design an objective function to maximize the probability $p(Y|X)$ to achieve the highest label score for different IOCs, which can be written as follows:

$$\begin{aligned} argmax\ log(p(Y|X)) = argmax\ (S(X,Y) - \\ log(\sum_{\tilde{y} \in Y_X} e^{S(X,\tilde{y})})) \end{aligned} \tag{6}$$

By solving the objective function above, we assign correct labels to the n-gram components, according to which we can identify the IOCs with different lengths. Our multi-granular attention based IOC extraction method is capable of identifying different types of IOCs, and its evaluation is presented in Section 5.

4.2 Cyber Threat Intelligence Modeling

CTI modeling is an important step to explore the intricate relationship between heterogeneous IOCs. In our work, HIN is introduced to group different types of IOCs into a graph to explore their interactive relationships. In this section, we portray the main principle of threat intelligence modeling.

To model the intricate interdependent relationships among IOCs, we define the following 9 relationships among 6 types of IOCs as follows.

- ***R1***: To depict the relation of an attacker and the exploited vulnerability, we construct the ***attacker-exploit-vulnerability*** matrix A. For each element $A_{i,j} \in \{0,1\}$, $A_{i,j}$=1 indicates attacker i exploits vulnerability j.

- ***R2***: To depict the relation of an attacker and a device, we build the ***attacker-invade-device*** matrix D. For each element $D_{i,j} \in \{0,1\}$, $D_{i,j}$=1 indicates attacker i invades device j.

- ***R3***: Two attacker can cooperate to attack a target. To study the relationship of *attacker-attacker*, we construct the ***attacker-cooperate-attacker*** matrix C. For each element $C_{i,j} \in \{0,1\}$, $C_{i,j}$=1 indicates there exists a cooperative relationship between attacker i and j.

- ***R4***: To describe the relation of a vulnerability and the affected device, we build the ***vulnerability-affect-device*** matrix M. For each element $M_{i,j} \in \{0,1\}$, $M_{i,j}$=1 indicates vulnerability i affects device j.

- ***R5***: A vulnerability is often labeled as a specific attack type by Common Vulnerabilities and Exposures (*CVE*)

system[7]. To explore the relation of *vulnerability-attack type*, we build the ***vulnerability-belong-attack type*** matrix G, where each element $G_{i,j} \in \{0,1\}$ denotes if vulnerability i belongs to an attack type j.

- ***R6***: A vulnerability often involves one or more malicious files. To describe the relation of *vulnerability-file*, we build the ***vulnerability-include-file*** matrix F. For each element $F_{i,j} \in \{0,1\}$, $F_{i,j}$=1 denotes that vulnerability i includes malicious file j.

- ***R7***: A malicious file often targets a specific device. We establish the ***file-target-device*** matrix T to explore the relation of *file-device*. For each element $T_{i,j} \in \{0,1\}$, $T_{i,j}$=1 indicates malicious file i targets device j.

- ***R8***: Oftentimes, a vulnerability evolves from another. To study the relationship of *vulnerability-vulnerability*, we build the ***vulnerability-evolve-vulnerability*** matrix E, where each element $E_{i,j} \in \{0,1\}$ indicates if vulnerability i evolves from vulnerability j.

- ***R9***: To depict the relation *device-platform* that a device belongs to a platform, we build the ***device-belong-platform*** matrix P where each element $P_{i,j} \in \{0,1\}$ illustrates if device i belongs to platform j.

Based on the above 9 types of relationships, HINTI leverages the syntactic dependency parser [6] (e.g., subject-predicate-object, attributive clause, etc.) to automatically extract the 9 relationships among IOCs from threat descriptions, each of which is represented as a triple $(IOC_i, relation, IOC_j)$. For instance, given a security-related description: "*On May 12, 2017, WannaCry exploited the MS17-010 vulnerability to affect a larger number of Windows devices, which is a ransomware attack via encrypted disks*". Using the syntactic dependency parser, we can extract the following triples: (WannaCry, exploit, MS17-010), (MS17-010, affect, Windows device), (WannaCry, is, ransomware). Such triples extracted from various data sources can be incrementally assembled into HINTI to model the relationships among IOCs, which could offer a more comprehensive threat landscape that describes the threat context. Particularly, we further define 17 types of meta-paths shown in Table 1 to probe into the interdependent relationships over attackers, vulnerabilities, malicious files, attack types, devices, and platforms. HINTI is able to convey a richer context of threat events by scrutinizing 17 types of meta-paths and reveal the in-depth threat insights behind the heterogeneous IOCs **(see Section 6 for details)**.

4.3 Threat Intelligence Computing

In this section, we illustrate the concept of threat intelligence computing, and design a general threat intelligence computing framework based on heterogeneous graph convolutional networks, which quantifies and measures the relevance between IOCs by analyzing meta-path based semantic similarity. Here, we first provide a formal definition of threat intelligence computing based on heterogeneous graph convolutional networks.

Definition 4 Threat Intelligence Computing Based on Heterogeneous Graph Convolutional Networks. *Given the threat intelligence graph $G = (V, E)$, the meta-path set $M = \{P_1, P_2, \cdots, P_i\}$. The threat intelligence computing: i) computes the similarity between IOCs based on meta-path P_i to generate corresponding adjacency matrix A_i; ii) constructs the feature matrix of nodes X_i by embedding attribute information of IOCs into a latent vector space; iii) conducts graph convolution $GCN(A_i, X_i)$ to quantify the interdependent relationships between IOCs by following meta-path P_i, and embeds them into a low-dimensional space.*

The threat intelligence computing aims to model the semantic relationships between IOCs and measure their similarity based on meta-paths, which can be used for advanced security knowledge discovery, such as threat object classification, threat type matching, threat evolution analysis, etc. Intuitively, the objects connected by the most significant meta-paths tend to bear more similarity [37]. In this paper, we propose a weight-learning based threat intelligence similarity measure, which uses self-attention to improve the performance of similarity measurement between any two IOCs. This method can be formalized as below:

Definition 5 Weight-learning based Node Similarity Measure. *Given a set of symmetric meta-path set $P = [P_m]_{m=1}^{M'}$, the similarity $S(h_i, h_j)$ between any two IOCs h_i and h_j is defined as:*

$$S(h_i, h_j) = \sum_{m}^{M'} \overrightarrow{w} \frac{2 \cdot |\{h_{i \to j} \in P_m\}|}{|\{h_{i \to i} \in P_m\}| + |\{h_{j \to j} \in P_m\}|} \quad (7)$$

where $h_{i \to j} \in h_m$ is a path instance between IOC h_i and h_j following meta-path P_m, $h_{i \to i} \in P_m$ is that between IOC instance h_i and h_i, and $h_{j \to j} \in P_m$ is that between IOC instance h_j and h_j, where $|\{h_{i \to j} \in P_m\}| = \mathbf{C}_{P_m}(i,j)$, $|\{h_{i \to i} \in P_m\}| = \mathbf{C}_{P_m}(i,i)$, $|\{h_{j \to j} \in P_m\}| = \mathbf{C}_{P_m}(j,j)$, and $\mathbf{C}_{P_m}$ is the commuting matrix based on meta-path P_m defined below. $\overrightarrow{w} = [w_1, \ldots, w_m, \ldots, w_{M'}]$ denote the meta-path weights, where w_m is the weight of meta-paths P_m, and M' is the number of meta-paths.

$S(h_i, h_j)$ is defined in two parts: (1) the semantic overlap in the numerator, which describes the number of meta-path between IOC instance h_i and h_j; (2) and the semantic broadness in the denominator, which depicts the number of total meta-paths between themselves. The larger number of meta-path between IOC instance h_i and h_j, the more similar the two IOCs are, which is normalized by the semantic broadness

[7] http://cve.mitre.org/

of denominator. Moreover, different from existing similarity measures [37], we propose an attention mechanism based similarity measure method by introducing the weight vector $\vec{w} = [w_1, \ldots, w_m, \ldots, w_{M'}]$, which is a trainable coefficient vector to learn the importance of different meta-paths for characterizing IOCs.

Obviously, it is computationally expensive to measure the similarity among IOCs in the constructed heterogeneous graph as it usually requires to randomly walk a larger number of nodes in the graph. Fortunately, in our work, it is unnecessary to walk through the entire graph as we prescribe a limit by introducing predefined meta-paths, and we only focus on the symmetrical meta-paths presented in Table 1. To calculate the similarity between IOCs under different meta-path instances, we need to compute the corresponding commuting matrices [37] following the meta-paths.

Given a meta-path set $P = \sum_m^{M'} \{A_1, A_2, \cdots, A_{l+1}\}$, the meta-path based commuting matrix can be defined as $\mathbf{C}_P = U_{A_1A_2} \circ U_{A_2A_3} \cdots \circ U_{AA_{l+1}}$, where $\mathbf{C}_P(i, j)$ represents the probability of object $i \in A_1$ reaching object $j \in A_{l+1}$ under the path P, and $\circ$ is a connection operation. These symmetric meta-paths not only efficiently reduce the complexity of walking, but also ensures that the commuting matrix can be easily decomposed, which greatly reduces the computational costs. In addition, the symmetric meta-paths in the graph G allow us to leverage the pairwise random-walk [37] to further accelerate the computation.

With Eq. (7) and pairwise random-walk, we can obtain the similarity embedding between any two IOCs h_i and h_j under a meta-path set P. Based on the low-dimensional similarity embedding, we derive a weighted adjacent matrix between IOCs, denoted as $A_i \in R^{N \times N}$, where N is the number of a specific type of IOC in G. Meanwhile, to utilize the attributed information of nodes, we train a word2vec model [24] to embed the attribute information of nodes into a feature matrix $X_i \in R^{N \times d}$, where N is the number of IOCs in A_i, and d is the dimension of node feature. With the learned adjacency matrix A_i and its feature matrix X_i, we can leverage the classical GCN [18] to characterize the relationship between IOC h_i and h_j. Particularly, the layer-wise propagation rule of GCN can be defined as below:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{8}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of IOCs with self-connections, I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $W^{(l)}$ is a l-th layer trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as $relu$. $H^{(l)} \in R^{N \times d}$ is the matrix of activation in the l-th layer. We perform graph convolution [18] on A_i and X_i to generate the embedding Z between IOCs belonging to type i:

$$Z = f(X_i, A_i) = \sigma(\hat{A}_i \cdot relu(\hat{A}_i X_i W_i^{(0)}) W_i^{(1)}) \tag{9}$$

where $W_i^{(0)} \in R^{d \times H}$ is an input-to-hidden weight matrix for a hidden layer with H feature maps, $W_i^{(1)} \in R^{H \times F}$ is a hidden-to-output weight matrix with F feature maps in the output layer, $X_i \in R^{N \times d}$, N is the number of a specific type of IOCs, d is the dimension of their corresponding features, and σ is another activation function, such as $sigmoid$. $\hat{A}_i = \tilde{D}^{-\frac{1}{2}}\tilde{A}_i\tilde{D}^{-\frac{1}{2}}$ can be calculated offline. Here, we leverage the cross-entropy loss to optimize the performance of our proposed threat intelligence framework, written as follows:

$$Loss(Y_{lf}, Z_{lf}) = -\sum_{l \in Y_l} \sum_{f=1}^{F} Y_{lf} \cdot lnZ_{lf} \tag{10}$$

where Y_l is the set of node indices that have labels, Y_{lf} is the real label, and Z_{lf} is a corresponding label that our model predicts. Based on Eq. (10), we conduct stochastic gradient descent to continuously optimize the neural network weights $W_i^{(0)}$, $W_i^{(1)}$, and $\vec{w}$ to reduce the loss, and build a general threat intelligence computing framework. Using this framework, security organizations are able to mine richer security knowledge hidden in the interdependent relationships among IOCs.

5 Experimental Evaluation

5.1 Dataset and Settings

We develop a threat data collector to automatically collect cyber threat data from a set of sources, including 73 international security blogs (e.g., *fireeye, cloudflare*), hacker forum posts (e.g., *Blackhat, Hack5*), security bulletins (e.g., *Microsoft, Cisco*), CVE detail description, and ExploitDB. A complete list of data sources is presented in the Baidu cloud[8]. We set up a daemon to collect the newly generated security events every day. So far, more than 245,786 security-related data describing threat events have been collected. For training and evaluating our proposed IOC extraction method, 30,000 samples from 5,000 texts are annotated by utilizing the *B-I-O* sequence tagging method (**see Section 2.2 for the example**), and an annotation example is shown in Figure 2.

For 30,000 labeled samples, we randomly select 60% of samples as a training set, 20% of samples as a verification set, and the rest of the samples as our test set. Based on the data sets, we comprehensively evaluate the performance of HINTI for extracting IOCs and threat intelligence computing. We run all of the experiments on 16 cores Intel(R) Core(TM) i7-6700 CPU @3.40GHz with 64GB RAM and 4× NVIDIA Tesla K80 GPU. The software programs are executed on the TensorFlow-GPU framework on Ubuntu 16.0.4.

5.2 Evaluation of IOC Extraction

A set of experiments are conducted to evaluate the sensitivity of different parameters in the multi-granular based IOC

[8] https://pan.baidu.com/s/1J631WMYY_T_awa8aY5xy3A

extraction model. We mainly consider 8 hyper-parameters that seriously impact the performance of the model as shown in Table 2. More specifically, Embedding_dim is one of the most important factors that determine the generalization capability of the model. Here, we fix other parameters while fine-tuning the embedding size in the range of $(50, 100, 150, 200, 250, 300, 350, 400)$. Experimental results show that the accuracy of extracted IOC achieves the best when Embedding_dim=300. Learning_rate is another major factor for determining the stride of gradient descent in minimizing the loss function, which determines whether the model can find a global optimal solution. We fix other parameters to fine-tune the Learning_rate in the range of $(0.001, 0.005, 0.01, 0.05, 0.1, 0.5)$, and the performance reaches the best when the Learning_rate=0.001. Similarly, we fine-tune the other hyper-parameters with 5,000 epochs, and the hyper-parameters allowing our model to perform optimally are recorded in Table 2.

Table 2: Hyperparameters setting in the multi-granular based IOC extraction method.

Parameter	value	Parameter	Value
Embedding_dim	300	Hidden_dim	128
Sequence_length	500	Epoch_num	5,000
Learning_rate	0.001	Batch_size	64
Dropout_rate	0.5	Optimizer	Adam

Table 3: Performance of IOC extraction w.r.t. IOC types.

IOC Type	Precision	Recall	Micro-F1
IP	99.56	99.52	99.54
File	94.36	96.88	95.60
Type	99.86	99.81	99.83
Email	99.32	99.87	99.49
Device	93.26	92.78	93.02
Vender	93.07	94.45	94.24
Version	96.98	97.99	97.48
Domain	96.58	95.89	96.23
Software	88.25	89.31	88.78
Function	95.03	95.59	95.31
Platform	94.31	92.57	93.43
Malware	89.76	91.23	90.49
Vulnerability	99.25	98.73	98.99
Other	98.29	98.42	98.35

In this paper, we extract 13 major types of IOCs, and the

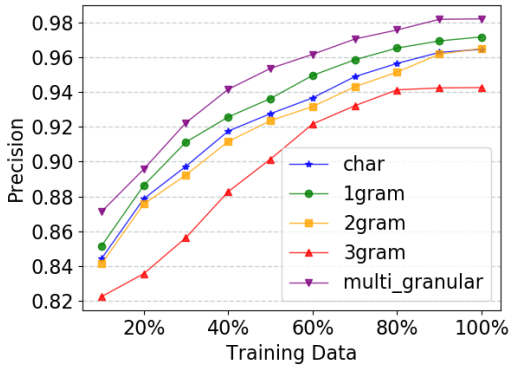


Figure 7: Performance of IOC extraction using embedding features with different granularity.

performance is presented in Table 3. Overall, our IOC extraction method demonstrates excellent performance in terms of precision, recall, and Micro-F1 (i.e., micro-averaged F1-score) for most types of IOCs, such as *function*, *malicious IP*, and *device*. However, we observe a performance degradation when recognizing software and malware. This can be attributed to the fact that most software and malware is named by random strings such as md5 hash. Moreover, we find that the number of training samples impacts the performance of the model. Specifically, the performance becomes unsatisfactory (e.g., *Software, Malware*) when the number of a certain type of training samples is insufficient (i.e., less than 5,000).

In order to verify the effectiveness of multi-granular embedding features, we assess the performance of IOC extraction with features of different granularity including *char-level*, *1-gram*, *2-gram*, *3-gram* and *multi-granular* features. The experimental results are demonstrated in Figure 7, from which we can observe that the proposed multi-granular embedding feature outperforms others since it leverages the attention mechanism to simultaneously learn multi-granular features to characterize different patterns of IOCs.

Next, to verify the effectiveness of the proposed IOC extraction method, we compare it with the state-of-the-art entity recognition approaches, including general NER tools *NLTK NER*[9], and *Stanford NER*[10], professional IOC extraction method *Stucco* [16] and *iACE* [22], and popular entity recognition approaches *CRF* [21], *BiLSTM* and *BiLSTM+CRF* [15]. The experimental results of different methods on real-world data are demonstrated in Table 4.

The results indicate that our proposed IOC extraction outperforms the state-of-the-art entity recognition methods and tools in terms of precision, recall, and Micro-F1, and its improvement can be attributed to the following factors. First, compared with *Standford NER* and *NLTK NER*, the NLP tools

[9] http://www.nltk.org/book/ch07.html
[10] https://stanfordnlp.github.io/CoreNLP/ner.html

Table 4: Performance of threat entity recognition using different methods.

Method	Accuracy	Precision	Micro-F1
NLTK NER	69.45	68.51	67.49
Stanford NER	68.35	66.74	68.58
iACE	92.14	91.26	92.25
Stucco	91.16	92.21	91.47
CRF	92.64	91.80	92.65
BiLSTM	94.78	95.21	94.35
BiLSTM+CRF	96.38	96.42	96.27
Multi-granular	**98.59**	**98.72**	**98.69**

trained with general news corpora, our method uses a security-related training corpus collected and labeled by ourselves as a data source for training our model. Second, different from the rule-based extraction approaches (e.g., *iACE* and *Stucco*), our proposed deep learning based method provides an end-to-end system with more advanced features to represent various IOCs. Third, comparing to RNN-based methods (e.g., *BiLSTM* and *BiLSTM+CRF*), our proposed method brings in multi-granular embedding sizes (*char-level, 1-gram, 2-gram,* and *3-gram*) to simultaneously learn the characteristics of various sizes and types of IOCs, which can identify more complex and irregular IOCs. Last but not the least, our method implements an attention mechanism to learn the weights of features with various scales to effectively characterize different types of IOCs, further enhancing the IOC recognition accuracy.

6 Application of Threat Intelligence Computing

Our proposed threat intelligence computing framework based on heterogeneous graph convolutional networks can be used to mine novel security knowledge behind heterogeneous IOCs. In this section, we evaluate its effectiveness and applicability using three real-world applications: profiling and ranking for CTIs, attack preference modeling, and vulnerability similarity analysis.

6.1 Threat Profiling and Significance Ranking of IOCs

Due to the disparity in the significance of threats, it is important to derive the threat profile and rank the significance of IOCs for demystifying the landscape of threats. However, most of the existing CTIs are incapable of modeling the associated relationships between heterogeneous IOCs.

Different from isolated CTIs, HINTI leverages HIN to model the interdependent relationships among IOCs with two characteristics: first, the isolated IOCs can be integrated into a graph-based HIN to clearly display the associated relationships among IOCs, which is capable of directly depicting the basic threat profile. For example, Figure 3 depicts a threat profiling sample: an attacker utilizes *CVE-2017-0143* vulnerability to invade *Vista SP2* and *Win7 SP1* devices belonging to the Microsoft platform, and CVE-2017-0143 is a *remote code execution* vulnerability that uses a *"SMB.bat"* malicious file. Second, the significance of IOCs in HINTI can be naturally ranked based on the proposed threat intelligence computing framework.

Table 5 shows the top 5 authoritative ranking score [35] of vulnerability, attacker, attack type, and platform, from which security experts can gain a clear insight into the impact of each IOC. Degree centrality [33], which describes the number of links incident upon a node, is widely used in evaluating the importance of a node in a graph. It can used to quantify the immediate risk of a node that connects with other nodes for delivering network flows, such as virus spreading. Here, degree centrality can be utilized in verifying the effectiveness of the proposed threat intelligence computing framework in ranking the importance of IOCs. It is worth noting that both our ranking method and degree centrality work regardless of the time of attacks. We compute the degree centrality ranking of IOCs based on the fact that the node with a higher degree centrality is more important than a node with a lower one. For instance, if the degree centrality of a vulnerability is higher, it indicates that this vulnerability is exploited by more attackers or it affects more devices. The ranking result of degree centrality shown in Table 5 is consistent with the ranking result based on the proposed threat intelligence computing framework, demonstrating the capability of the CTI computing framework in ranking the importance of different types of IOCs.

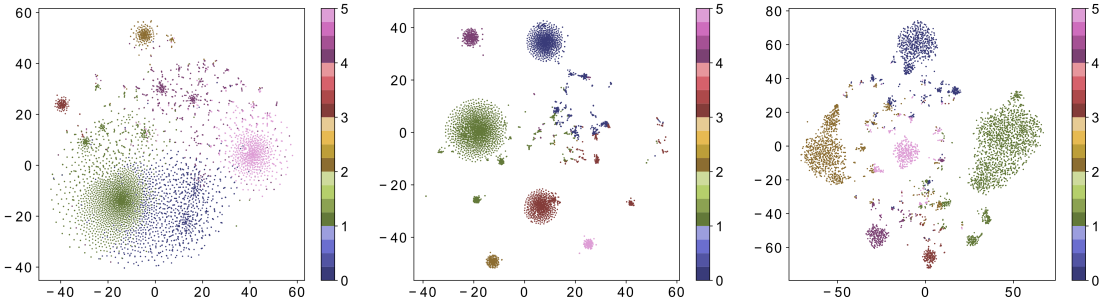
6.2 Attack Preference Modeling

Attack preference modeling is meaningful for security organizations to gain insight into the attack intention of attackers, build attack portraits, and develop personalized defense strategies. Here, we leverage HINTI to integrate different types of IOCs and their interdependent relationships to comprehensively depict the picture of attack events, which helps model the attack preferences. With the proposed threat intelligence computing framework, we model attack preferences by clustering the embedded attackers' vectors.

In this task, each malicious IP address is treated as an intruder, and its attack preferences are mainly reflected in three features including the platforms it destroys (including *Windows, Linux, Unix, ASP, Android, Apache, etc*), the industries it invades (e.g., *education, finance, government, Internet of Things, and Industrial control system, etc*), and the exploit types it employs (e.g., *DOS, Buffer overflow, Execute code,*

Table 5: The significance ranking of different types of IOCs. ($CVE1$: CVE-2017-0146, $CVE2$: CVE-2006-5911, $CVE3$: CVE-2008-6543, $CVE4$: CVE-2012-1199, $CVE4$: CVE-2006-4985; AR: Authoritative Ranking, DC: Degree Centrality value.)

Vulnerability			**Attacker**			**Platform**			**Attack Type**		
No.	AR	DC	Monicker	AR	DC	Category	AR	DC	Exploit_type	AR	DC
$CVE1$	0.2713	7,643	Meatsploit	0.2764	549	PHP	0.4562	17,865	Webapps	0.5494	11,648
$CVE2$	0.2431	7,124	GSR team	0.1391	327	windows	0.2242	13,793	DOS	0.1772	8,741
$CVE3$	0.2132	6,833	Ihsan	0.0698	279	Linux	0.0736	8,792	Overflow	0.1533	7,652
$CVE4$	0.1826	6,145	Techsa	0.0695	247	Linux86	0.0623	8,147	CSRF	0.0966	5,433
$CVE5$	0.1739	5,637	Aurimma	0.0622	204	ASP	0.0382	5,027	SQL	0.0251	2,171



(a) $AVDV^TA^T$ (P_{13}) (b) $AVDPD^TV^TA^T$ (P_{17}) (c) $AVPV^TA^T$ (P_{14})

Figure 8: The performance of attack preference modeling with different meta-paths, in which the preference of attacker i is reduced to a two-dimensional space (x_i, y_i) and each cluster represents a group with a specific attack preference.

Sql injection, XSS, Gain information, etc).

Specifically, we first utilize our proposed threat intelligence computing framework to embed each attacker into a low-dimensional vector space, and then perform *DBSCAN* algorithm on the embedded vector to cluster attackers with the same preferences into corresponding groups. Figure 8 shows the top 3 clustering results under different types of meta-paths, in which the meta-path $AVDPD^TV^TA^T$ (P_{17}) performs the best performance with compact and well-separated clusters, indicating that it contains richer semantic relationships for characterizing attack preferences than other meta-paths.

To verify the effectiveness of attack preference modeling, we identify 5,297 distinct attackers (each unique IP address is treated as an attacker) who have submitted at least 10 cyber attacks. For these attackers, five cybersecurity researchers consisting of three doctoral and two master students spent about fortnight to manually annotate their attack preferences from three perspectives: the platforms they destroyed, the industries they attacked, and the attack types they exploited. To ensure the accuracy of data labeling, we test the consistency of the tags for the 5,297 attackers and remove the samples with ambiguous tags. As a result, we obtain 3,000 samples with consistent tags. Based on the labeled samples, we further evaluate the performance of different meta-paths on modeling attack preferences. In the attack modeling scenario, we only focus on the meta-paths that both the start node and the end node are attackers. The experimental results are demonstrated in Table 6. Obviously, different meta-paths display different abilities in characterizing the attack preferences of cyber intruders. The performance when using P_{17} is more

Table 6: Performance of modeling attack preference with different meta-paths.

Metapath	**Accuracy**	**Precision**	**Micro-F1**
P_1	74.31	76.22	75.25
P_4	71.16	73.27	72.16
P_5	69.15	71.43	70.27
P_{12}	72.14	76.46	74.24
P_{13}	79.65	81.31	80.47
P_{14}	77.48	79.34	78.40
P_{15}	80.17	79.76	79.96
P_{17}	**81.39**	**81.72**	**81.55**

superior than the one with other meta-paths, which indicates that P_{17} holds more valuable information that characterizes the attack preferences of cybercriminals, since P_{17} includes the semantics information of P_1, P_4, P_5 *and* $P_{12} \sim P_{15}$.

In addition, we compare the capabilities of our proposed computing framework with those of other state-of-the-art embedding methods in terms of attack preference modeling. Our analysis result shows that the accuracy of attack preference modeling reaches 0.81, which outperforms the existing popular models *Node2vec* (with precision of 0.71) [1], *metapaht2vec* (with precision of 0.73) [11] and *HAN* (with precision of 0.76) [42]. The performance improvement can be attributed to the following characteristics. First, our computing framework utilizes weight-learning to learn the significance of different meta-paths for evaluating the similarity between attackers. Second, the proposed computing framework leverages GCN to learn the structural information between attackers to obtain more discriminative structural features that improves the performance of attack preference modeling.

6.3 Vulnerability Similarity Analysis

Vulnerability classification or clustering is crucial for conducting vulnerability trend analysis, the correlation analysis of incidents and exploits, and the evaluation of countermeasures. The traditional vulnerability analysis relies on the manual investigation of the source codes, which requires expert expertise and consumes considerable efforts. In this section, we propose an unsupervised vulnerability similarity analysis method based on the proposed threat intelligence computing framework, which can automatically group similar vulnerabilities into corresponding communities. Particularly, the vulnerability-related IOCs are first embedded into a low-dimensional vector space using CTI computing framework. Then, the *DBSCAN* algorithm is performed on the embedded vector space to cluster vulnerabilities into corresponding communities. The clustering results are presented in Figure 9.

Figure 9 (c) shows all vulnerabilities are clustered into 12 clusters using meta-path $VDPD^TV^T(P_{16})$, which is very close to the classification standard (i.e., 13) recommended by CVE details, an authoritative database that publishes vulnerability information. By manually analyzing the training samples, we find that *HTTP Response Splitting* vulnerability does not appear in our dataset. Therefore, our cluster number (i.e., 12) is consistent with *CVE Details*[11]. To further validate the effectiveness of threat intelligence computing framework for vulnerability clustering, we randomly select 100 vulnerabilities from each cluster for manual inspection to measure the consistency of the vulnerability types in each cluster, and the results are presented in Table 7. Obviously, the clustering performance of cluster 8 (i.e, File Inclusion) and cluster 10 (i.e., Directory Traversal) is remarkably worse than other clusters. To explain the reason, we examine our training data and the computing framework. We found that the proportion of these two types of vulnerabilities is too small (cluster 8 is 3.4% and cluster 10 is 4.2%), making our computing framework very likely to be under-fit with insufficient data. However, the proposed computing framework performs well on most types of vulnerabilities in an unsupervised manner, especially given sufficient samples (e.g, cluster 7 is 17.6% and cluster 5 is 15.7%).

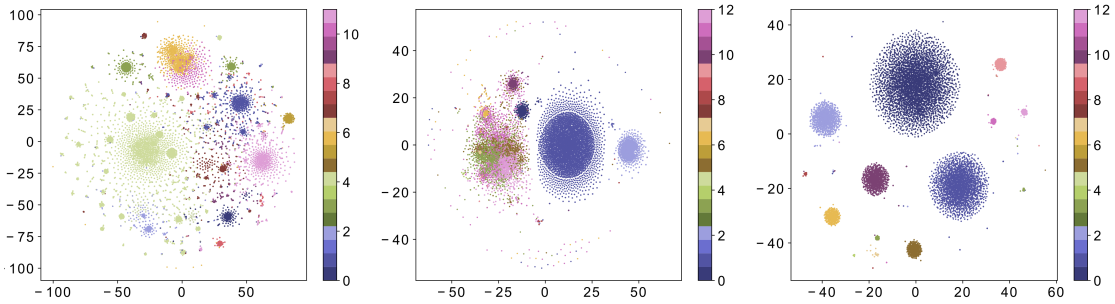
Table 7: Accuracy of vulnerability clustering.

Cluster ID	Vulnerability type	Accuracy
cluster1	Denial of Service	80.12
cluster2	XSS	83.53
cluster3	Execute Code	81.50
cluster4	Overflow	76.50
cluster5	Gain Privilege	91.56
cluster6	Bypass Something	71.74
cluster7	CSRF	93.27
cluster8	File Inclusion	61.72
cluster9	Gain Informa	70.42
cluster10	Directory Traversal	69.49
cluster11	Memory Corruption	81.56
cluster12	SQL Injection	80.67
average	#	**78.51**

In addition, by examining the clustering results, we have an observation that the vulnerabilities in the same cluster are likely to have evolutionary relationships. For instance, *CVE-2018-0802*, an office zero-day vulnerability, is evolved from the *CVE-2017-11882*. They both include *EQNEDT32.exe* file used to edit the formula in Office software, which allows remote attackers to execute arbitrary codes by constructing a malformed font name. The modeling and computation of interdependent relationships among IOCs in HINTI facilitate the discovery of such intricate connections between vulnerabilities.

In summary, HINTI is capable of depicting a more comprehensive threat landscape, and the proposed CTI computing framework has the ability to bring novel security insights toward different real-world security applications. However, there are still numerous opportunities for enhancing these security applications. Specifically, for attack preference modeling, although each individual IP address is treated as an attacker, we cannot determine whether it belongs to a real attacker or is disguised by a proxy. Fortunately, even if the real attack address cannot be captured, understanding the attack preferences of these IP proxies, which are widely used in cybercrime, is also meaningful for gaining insight into the cyber threats. For vulnerability similarity analysis, data imbal-

[11] https://www.cvedetails.com/



(a) $AVDV^TA^T$ (P_{13}) (b) VDV^T (P_{10}) (c) $VDPD^TV^T$ (P_{16})

Figure 9: Illustration of the vulnerability similarity analysis based on different meta-paths, in which vulnerability i can be reduced into a two-dimensional space (x_i, y_i) and each cluster indicates a particular type of vulnerability.

ance issue affects the performance of model, and inadequate training samples often result in model underfitting, as shown in the case of cluster 8 and cluster 10.

7 Related Work

Cyber Treat Intelligence. An increasing number of security vendors and researchers start exploring CTI for protecting system security and defending against new threat vectors [28]. Existing CTI extraction tools such as *IBM X-Force*[12], *Threat crowd*[13], *Opencti.io*[14], *AlienVault*[15], *CleanMX*[16] and *PhishTank*[17] use regular expression to synthesize IOC from the descriptive texts. However, these methods often produce high false positive rate by misjudging legitimate entities as IOCs [22].

Recently, Balzarotti et al. [2] develop a system to extract IOCs from web pages and identify malicious URLs from JavaScript codes. Sabottke et al. [31] propose to detect potential vulnerability exploits by extracting and analyzing the tweets that contain "CVE" keyword. Liao et al. [22] present a tool, *iACE*, for automatically extracting IOCs, which excels at processing technology articles. Nevertheless, *iACE* identifies IOCs from a single article, which does not consider the rich semantic information from multi-source texts. Zhao et al. [46] define different ontologies to describe the relationship between entities based on expert knowledge. Numerous popular CTI platforms including *IODEF* [9], *STIX* [3], *TAXII* [40], *OpenIOC* [13], and *CyBox* [19] focus on extracting and sharing CTI. Yet, none of the existing approaches could uncover the interdependent relations among CTIs extracted from multi-source texts, let alone quantifying CTIs' relevance and mining valuable threat intelligence hidden behind the isolated CTIs.

[12]https://exchange.xforce.ibmcloud.com/
[13]https://www.threatcrowd.org/
[14]https://demo.opencti.io/
[15] https://otx.alienvault.com/
[16]http://list.clean-mx.com
[17]https://www.phishtank.com

Heterogeneous Information Network. Real-world systems often contain a large number of interacting, multi-typed objects, which can naturally be expressed as a heterogeneous information network (HIN). HIN, as a conceptual graph representation, can effectively fuse information and exploit richer semantics in interacting objects and links [37]. HIN has been applied to network traffic analysis [38], public social media data analysis [45], and large-scale document analysis [41]. Recent applications of HIN include mobile malware detection [14] and opioid user identification [12]. In this paper, *for the first time*, we use HIN for CTI modeling.

Graph Convolutional Network. Graph convolutional networks (GCN) [17] has become an effective tool for addressing the task of machine learning on graphs, such as semi-supervised node classification [17], event classification [29], clustering [8], link prediction [27], and recommended system [44]. Given a graph, GCN can directly conduct the convolutional operation on the graph to learn the nonlinear embedding of nodes. In our work, to discern and reveal the interactive relationships between IOCs, we utilize GCN to learn more discriminative representation from attributes and graph structure simultaneously, which is the premise for threat intelligence computing.

8 Discussion

Data Availability. The proposed framework assumes that sufficient threat description data can be obtained for generating comprehensive and the latest CTIs. Fortunately, with the growing prosperity of social media, an increasing number of security-related data (e.g., blogs, posts, news and open security databases) can be collected effortlessly. To automatically collect security-related data, we develop a crawler system to collect threat description data from 73 international security sources (e.g., blogs, hacker forum posts, security bulletins, etc), providing sufficient raw materials for generating cyber threat intelligence.

Model Extensibility. In this paper, 6 types of IOCs and 9

types of relationships are modeled in HINTI. However, our proposed framework is extensible, in which more types of IOCs and relationships can be enrolled to represent richer and more comprehensive threat information, such as malicious domains, phishing Emails, attack tools, their interactions, etc.

High-level Semantic Relations. In view of the computational complexity of the model, our threat intelligence computing method focues on utilizing the meta-paths to quantify the similarity between IOCs while ignoring the influence of the meta-graph on it, which inevitably misses characterizing some high-level semantic information. Nevertheless, the proposed computing framework introduces an attention mechanism to learn the signification of different meta-paths to characterize IOCs and their interactive relationships, which effectively compensates for the performance degradation caused by ignoring the meta-graphs.

Security Knowledge Reasoning. Although our proposed framework exhibits promising results in CTI extraction and modeling computing, how to implement advanced security knowledge reasoning and prediction is still an open problem, e.g., it remains challenging to predict whether a vulnerability could potentially affect a particular type of devices in the future. We will investigate this problem in the future.

9 Conclusion

This work explores a new direction of threat intelligence computing, which aims to uncover new knowledge in the relationships among different threat vectors. We propose HINTI, a cyber threat intelligence framework, to model and quantify the interdependent relationships among different types of IOCs by leveraging heterogeneous graph convolutional networks. We develop a multi-granular attention mechanism to learn the importance of different features, and model the interdependent relationships among IOCs using HIN. We propose the concept of threat intelligence computing and present a general intelligence computing framework based on graph convolutional networks. Experimental results demonstrate that the proposed multi-granular attention based IOC extraction method outperforms the existing state-of-the-art methods. The proposed threat intelligence computing framework can effectively mine security knowledge hidden in the interdependent relationships among IOCs, which enables crucial threat intelligence applications such as threat profiling and ranking, attack preference modeling, and vulnerability similarity analysis. We would like to emphasize that the knowledge discovery among interdependent CTIs is a new field that calls for a collaborative effort from security experts and data scientists.

In future, we plan to develop a predicative and reasoning model based on HINTI and explore preventative countermeasures to protect cyber infrastructure from future threats. We also plan to add more types of IOCs and relations to depict a more comprehensive threat landscape. Moreover, we will leverage both meta-paths and meta-graphs to characterize the IOCs and their interactions to further improve the embedding performance, and to strike a balance between the accuracy and computational complexity of the model. We will also investigate the feasibility of security knowledge prediction based on HINTI to infer the potential future relationships between the vulnerabilities and devices.

Acknowledgement

We would like to thank our shepherd Tobias Fiebig, and the anonymous reviewers for providing valuable feedback on our work. We also thank Hao Peng and Lichao Sun for their feedback on the early version of this work. This work was supported in part by National Science Foundation grants CNS1950171, CNS-1949753. It was also supported by the NSFC for Innovative Research Group Science Fund Project (62141003), National Key R&D Program China (2018YFB0803 503), the 2018 joint Research Foundation of Ministry of Education, China Mobile (MCM20180507) and the Opening Project of Shanghai Trusted Industrial Control Platform (TICPSH202003020-ZC). Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of any funding agencies.

References

[1] Jure Leskovec Aditya Grover. node2vec: Scalable feature learning for networks. In *Acm Sigkdd International Conference on Knowledge Discovery Data Mining*, 2016.

[2] Marco Balduzzi, Marco Balduzzi, and Davide Balzarotti. Automatic extraction of indicators of compromise for web applications. In *WWW*, 2016.

[3] Sean Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, 11:1–22, 2012.

[4] Eric W Burger, Michael D Goodman, Panos Kampanakis, and Kevin A Zhu. Taxonomy model for cyber threat intelligence information exchange technologies. In *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, pages 51–60, 2014.

[5] Onur Catakoglu, Marco Balduzzi, and Davide Balzarotti. Automatic extraction of indicators of compromise for web applications. *The web conference*, pages 333–343, 2016.

[6] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. pages 740–750, 2014.

[7] Qian Chen and Robert A. Bridges. Automated behavioral analysis of malware a case study of wannacry ransomware. In *16th IEEE ICMLA*, pages 454–460, 2017.

[8] Weilin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Chojui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. *knowledge discovery and data mining*, pages 257–266, 2019.

[9] Roman Danyliw, Jan Meijer, and Yuri Demchenko. The incident object description exchange format. *International Journal of High Performance Computing Applications*, 5070:1–92, 2007.

[10] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *23rd ACM SIGKDD*, pages 135–144, 2017.

[11] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *23rd ACM SIGKDD*, pages 135–144. ACM, 2017.

[12] Yujie Fan, Yiming Zhang, Yanfang Ye, and Xin Li. Automatic opioid user detection from twitter: Transductive ensemble built on different meta-graph based similarities over heterogeneous information network. In *IJCAI*, pages 3357–3363, 2018.

[13] Fireeye. Openioc. https://www.fireeye.com/blog/threat-research/2013/10/openioc-basics.html. Accessed January 20, 2020.

[14] Shifu Hou, Yanfang Ye, Yangqiu Song, and Melih 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*, pages 1507–1515, 2017.

[15] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv:1508.01991*, 2015.

[16] Michael D Iannacone, Shawn Bohn, Grant Nakamura, John Gerth, Kelly MT Huffer, Robert A Bridges, Erik M Ferragut, and John R Goodall. Developing an ontology for cyber security knowledge graphs. *CISR*, 15:12, 2015.

[17] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv: Learning*, 2016.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Tero Kokkonen. Architecture for the cyber security situational awareness system. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pages 294–302. Springer, 2016.

[20] Mehmet Necip Kurt, Yasin Yılmaz, and Xiaodong Wang. Distributed quickest detection of cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security*, 13(8):2015–2030, 2018.

[21] John Lafferty, Andrew Mccallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, pages 282–289, 2001.

[22] Xiaojing Liao, Yuan Kan, Xiao Feng Wang, Li Zhou, and Raheem Beyah. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In *ACM Sigsac Conference on Computer Communications Security*, 2016.

[23] Rob Mcmillan. Open threat intelligence. http://www.gartner.com/doc/2487216/definition-threat-intelligence. Accessed January 20, 2020.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[25] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In *Proceedings of the 2016 IEEE Advances in Social Networks Analysis and Mining*, pages 860–867, 2016.

[26] Eric Nunes, Ahmad Diab, Andrew Gunn, Ericsson Marin, Vineet Mishra, Vivin Paliath, John Robertson, Jana Shakarian, Amanda Thart, and Paulo Shakarian. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *2016 IEEE ISI*, pages 7–12, 2016.

[27] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. pages 2609–2615, 2018.

[28] P Pawlinski, P Jaroszewski, P Kijewski, L Siewierski, P Jacewicz, P Zielony, and R Zuber. Actionable information for security incident response. *European Union Agency for Network and Information Security, Heraklion, Greece*, 2014.

[29] Hao Peng, Jianxin Li, Qiran Gong, Yangqiu Song, Yuanxing Ning, Kunfeng Lai, and Philip S Yu. Fine-grained event categorization with heterogeneous graph convolutional networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3238–3245. AAAI Press, 2019.

[30] Sara Qamar, Zahid Anwar, Mohammad Ashiqur Rahman, Ehab Al-Shaer, and Bei-Tseng Chu. Data-driven analytics for cyber-threat intelligence and information sharing. *Computers & Security*, 67:35–58, 2017.

[31] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In *USENIX Security*, 2015.

[32] Carl Sabottke, Octavian Suciu, and Tudor Dumitraș. Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In *24th USENIX Security*, pages 1041–1056, 2015.

[33] Deepak Sharma and Avadhesha Surolia. *Degree Centrality*. Springer New York, 2013.

[34] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park. A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions. *Journal of Supercomputing*, pages 1–32, 2016.

[35] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: Principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[36] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: a structural analysis approach. *ACM SIGKDD*, 14(2):20–28, 2013.

[37] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[38] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S Yu, and Xiao Yu. Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Transactions on TKDD*, 7(3):11, 2013.

[39] Wiem Tounsi and Helmi Rais. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security*, 72:212–233, 2018.

[40] Thomas D Wagner, Esther Palomar, Khaled Mahbub, and Ali E Abdallah. Towards an anonymity supported platform for shared cyber threat intelligence. *risks and security of internet and systems*, pages 175–183, 2017.

[41] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. Text classification with heterogeneous information network kernels. In *13th AAAI*, 2016.

[42] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, P. Yu, and Yanfang Ye. Heterogeneous graph attention network. 2019.

[43] Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. *arXiv: Computation and Language*, 2018.

[44] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983. ACM, 2018.

[45] Jiawei Zhang, Xiangnan Kong, and Philip S Yu. Transferring heterogeneous links across location-based social networks. In *The 7th ACM international conference on Web search and data mining*, pages 303–312, 2014.

[46] Yishuai Zhao, Bo Lang, and Ming Liu. Ontology-based unified model for heterogeneous threat intelligence integration and sharing. In *2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 11–15, 2017.