Metagraph Aggregated Heterogeneous Graph Neural Network for Illicit Traded Product Identification in Underground Market

Yujie Fan[1], Yanfang Ye[1,*], Qian Peng[1], Jianfei Zhang[1], Yiming Zhang[1],
Xusheng Xiao[1], Chuan Shi[2], Qi Xiong[3], Fudong Shao[3], Liang Zhao[4]
[1]*Department of Computer and Data Sciences, Case Western Reserve University, OH, USA*
[2]*School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China*
[3]*Tencent Security Lab, Tencent, Guangdong, China*
[4]*Department of Computer Science, Emory University, GA, USA*
{*yxf370,yanfang.ye,qxp36,jxz1123,yxz2092,xxx175*}*@case.edu,*
shichuan@bupt.edu.cn, {*keonxiong,joeyshao*}*@tencent.com, liang.zhao@emory.edu*

***Abstract*—The emerging underground markets (e.g., Hack Forums) have been widely used by cybercriminals to trade in illicit products or services, which have played a vital role in the cybercriminal ecosystem. In order to combat the evolving cybercrimes, in this paper, we propose and develop an intelligent framework (named *PIdentifier*) to automate the analysis of Hack Forums for the identification of illicit product traded in a private contract at the first attempt (to evade the law enforcement, a private contract is made between a vendor and a buyer where the traded product and its detail are invisible). In *PIdentifier*, based on the large-scale extracted user profiles, user posts and different types of relations within the complex ecosystem in Hack Forums, we first introduce an attributed heterogeneous information network (AHIN) to model the rich semantics and complex relations among multi-typed entities (i.e., vendors, buyers, products, comments and topics). Then, we design different metagraphs to formulate the relatedness between buyers and products based on which a metagraph aggregated heterogeneous graph neural network (denoted as *mHGNN*) is proposed to learn node representations for illicit traded product identification by attentively propagating and aggregating the neighborhood information defined by the designed metagraphs. Comprehensive experiments are conducted on the real-world dataset collected from Hack Forums. Promising results demonstrate the performance of our proposed *PIdentifier* framework in illicit traded product identification by comparison with the state-of-the-art baselines.**

***Keywords*-Attributed Heterogeneous Information Network; Graph Neural Network; Underground Market; Illicit Traded Product Identification.**

I. Introduction

As the Internet has become one of the most important drivers in the global economy, it not only provides an open and shared platform for legitimate users to realize their innovations but also for cybercriminals to gain profits from illegal Internet-based activities. Underground markets emerging in the form of online underground forums, such as Hack Forums [1], Nulled [2], and BlackHatWorld [3], have been widely used by cybercriminals to advertise and trade in illicit products (e.g., stolen credit cards, malware) or services (e.g., bogus Amazon reviews, hacking services) for considerable profits, e.g., the estimated annual revenue for a campaign of stolen credit cards is $300 millions [4]. Since the underground markets have played a vital role in the cybercriminal ecosystem, to combat the evolving cybercrimes, there's imminent need to gain a deeper understanding about the dynamics and operations of the illicit activities and thus enable the law enforcement for proactive interventions. To this end, in this paper, we use Hack Forums, one of the most prevalent underground forums consisting of 4,746,471 registered users with 6,085,621 posted threads and 59,873,494 comments, as a showcase to investigate the profit model and transaction process.

To regularize the trading activities (e.g., anti-scam), more and more underground markets, including Hack Forums, have enforced escrow contracts between vendors and buyers. As illustrated in Figure 1, the vendor "T***y" (we here anonymize his user name), who sells a variety of different illicit products and services (to simplify the description, we will use product to denote product/service throughout the paper), has had 593 contracts completed in Hack Forums. We use one of his public contracts made with the buyer "J***X" to illustrate the transaction process: ① the vendor first posts a thread in a primary marketplace to advertise a lifetime *Premium Facebook Hacker* with $14.99 promotion price; ② the buyer may ask questions about the product by commenting on the advertised thread before ③ initializing a contract; after ④ the vendor accepts the contract, ⑤ the buyer pays in cryptocurrency while ⑥ the vendor delivers the product; after the transaction, ⑦ the contract will be marked as "complete" and ⑧ the buyer may further comment on the purchased product about purchase experience. Such public contract, which includes the traded product and its details (e.g., product description, price, obligations and terms, etc.), provides the defenders and law enforcement valuable information about the illicit trading activities in underground markets. Unfortunately, in Hack Forums, the majority of the contracts are made private

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00022

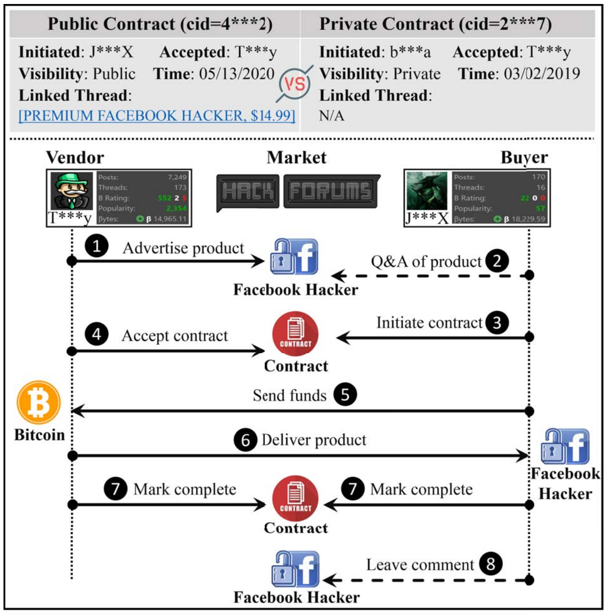


Figure 1. Profit model and transaction process in a underground market.

in which the traded products and their details are invisible. To put this into perspective, for all the 34,046 contracts (during June 10, 2018 to Nov. 18, 2019) we crawled from a primary marketplace (i.e., *Premium Tools and Programs* under *Premium Sellers Section*) in Hack Forums, 28,432 of them (83.5%) are private and only 5,614 (16.5%) are made public. This calls for novel techniques to automate the data analysis for the identification of illicit traded products in private contracts in underground markets.

Through our analysis of the underground market (i.e., Hack Forums in this work), we observe that although the traded product in a private contract is invisible, the information of the vendor and buyer associated with this contract is accessible. To this end, we propose to leverage user profiles (e.g., popularity, bussiness rating, etc.) that may reveal users' popularity and reputations in Hack Forums as well as their posts (i.e., threads and comments) to build the connection between the vendor, product and buyer in a private contract. For example, as shown in Figure 1, the traded product in the private contract between vendor "T***y" and buyer "b***a" is invisible; however, after further investigation, we find that "b***a" makes several comments on the thread where "T***y" advertises a *Premium Instagram Hacker* to express his purchase and use experiences: "*I made the purchase. Please send via Skype*"; "*Thanks. Everything is fine. The product is working as described*". We also identify that there's only one contract made between "T***y" and "b***a". These investigative leads enable us to conclude that the traded product in this private contract is the *Premium Instagram Hacker* whose price and details can be accessed in the posted thread.

Based on the above observation, in this work, we formulate the traded product identification in a given private contract as a prediction problem: given a private contract associated with buyer b and vendor s, the prediction is to yield the probability of $\hat{y}_{bp}=f(b,p^{(s)})$, where $p^{(s)}$ is a product posted by s. To solve this problem, we propose and develop an integrated framework, named *PIdentifier*. In *PIdentifier*, we first present an attributed heterogeneous information network (AHIN) to model the rich semantics and complex relations among different types of entities (i.e., vendors, buyers, products, comments and topics) extracted from Hack Forums. Then, different metagraphs are built upon the constructed AHIN to formulate the relatedness between buyers and products. To learn node presentations in AHIN, we further propose a metagraph aggregated heterogeneous graph neural network (denoted as *mHGNN*), which consists of the following three major steps: metagraph-guided neighbor search, attentive propagation and aggregation, and multi-view fusion. Finally, given a private contract, we retrieve the node representations of its related *buyer-product* (i.e., b, $p^{(s)}$) pairs to predict which product (i.e., $p^{(s)}$) is associated with this private contract. Based on the real-world data collection from Hack Forums, extensive experiments are conducted to evaluate the efficacy of the proposed *PIdentifier*. Promising results demonstrate the performance of *PIdentifier* in traded product identification by comparison with the state-of-the-art baselines. The major contributions of this work are:

- We present a novel heterogeneous graph architecture for abstract representation of the extracted user profiles, user posts and different types of relations among them.
- We propose an innovative metagraph aggregated heterogeneous graph neural network (i.e., *mHGNN*) to learn node representations by attentively propagating and aggregating the neighborhood information of nodes guided by different designed metagraphs.
- To the best of our knowledge, this is *the first work* of identifying illicit traded products hidden in private contracts in underground market. The developed system will facilitate defenders and law enforcement to better understand the dynamics of illicit activities in underground market and thus devise effective interventions to combat the evolving cybercrimes.

The rest of this paper is organized as follows. Section II introduces the related concepts and formulates the problem. Section III presents our proposed method in detail. In Section IV, we comprehensively evaluate the performance of our developed framework. Section V discusses the related work. Finally, Section VI concludes the paper.

II. Problem Formulation

In this section, we introduce the preliminary concepts applied in our framework, and formally define the illicit traded product identification problem.

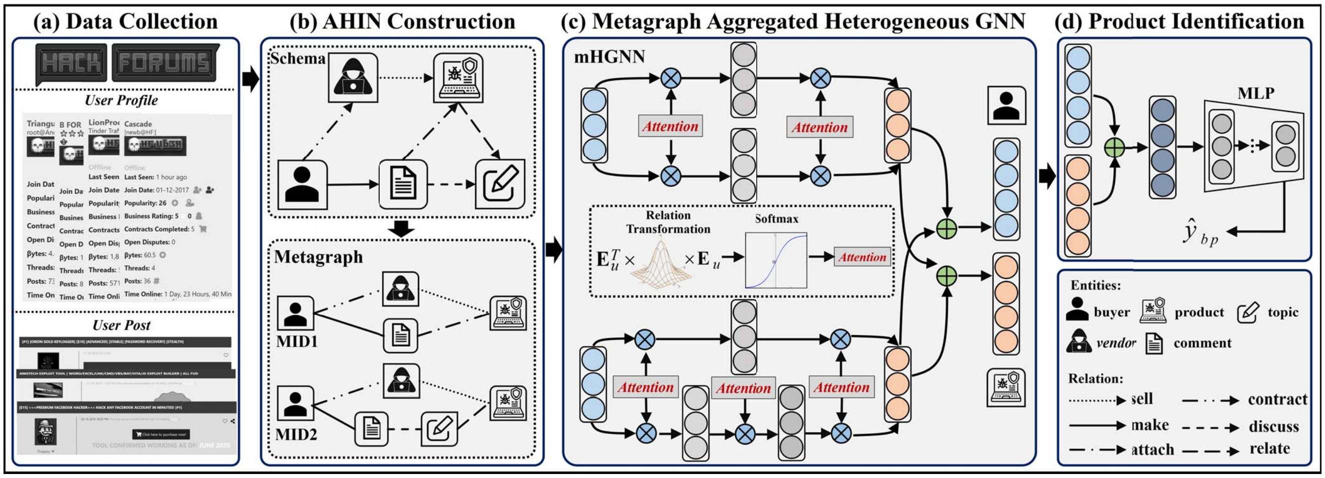


Figure 2. The overview of our proposed *PIdentifier* framework for illicit traded product identification in underground market.

Definition 1. *Attributed Heterogeneous Information Network (AHIN)* [5]. Let $\mathcal{T} = \{T_1, ..., T_m\}$ be a set of m entity types, $\mathcal{X}_i$ be the set of entities of type T_i and A_i be the set of attributes defined for entities of type T_i. An AHIN is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ with an entity type mapping ϕ: $\mathcal{V} \to \mathcal{T}$ and a relation type mapping ψ: $\mathcal{E} \to \mathcal{R}$, where $\mathcal{V} = \bigcup_{i=1}^{m} \mathcal{X}_i$ denotes the entity set and $\mathcal{E}$ is the relation set, $\mathcal{T}$ denotes the entity type set and $\mathcal{R}$ is the relation type set, $\mathcal{A} = \bigcup_{i=1}^{m} A_i$, and $|\mathcal{T}| + |\mathcal{R}| > 2$. *Network Schema* [5]. The network schema of an AHIN $\mathcal{G}$ is a meta-template for $\mathcal{G}$, denoted as a directed graph $\mathcal{T}_\mathcal{G} = (\mathcal{T}, \mathcal{R})$ with nodes as entity types from $\mathcal{T}$ and edges as relation types from $\mathcal{R}$.

Definition 2. *Metagraph* [6]. A metagraph $M = (\mathcal{T}_M, \mathcal{R}_M)$ is a sub-graph of network schema $\mathcal{T}_\mathcal{G} = (\mathcal{T}, \mathcal{R})$, where $\mathcal{T}_M \subseteq \mathcal{T}$ and $\mathcal{R}_M \subseteq \mathcal{R}$. A *reverse metagraph* M^- is defined on $M = (\mathcal{T}_M, \mathcal{R}_M)$, with same entity types but inverse relation types. Formally, $M^- = (\mathcal{T}_{M^-}, \mathcal{R}_{M^-})$, where $\mathcal{T}_{M^-} = \mathcal{T}_M$ and $\mathcal{R}_{M^-} = \mathcal{R}_M^{-1}$. We also define a function φ that transforms metagraph to its corresponding reverse metagraph, i.e., $\varphi : M \to M^-$.

Problem 1. *Illicit Traded Product Identification.* Based on the constructed AHIN $\mathcal{G}$ and the designed (reverse) metagraphs $\mathcal{M} = \{M_i^{(-)}\}_{i=1}^{K}$, given a private contract associated with buyer b and vendor s, the prediction is to yield the probability of $\hat{y}_{bp} = f(b, p^{(s)} | \mathcal{G}, \mathcal{M}; \Theta)$, where $p^{(s)}$ is any product posted by s, $(b, s), (s, p^{(s)}) \in \mathcal{E}$. The $p^{(s)}$ in the pair of $(b, p^{(s)})$ with the largest $\hat{y}_{bp}$ will be predicted as the traded product in a given private contract. Model parameter Θ will be learned in the following section.

III. Proposed Method

In this section, we introduce our proposed *PIdentifier* framework for illicit traded product identification in underground market.

A. Overview

The overview of *PIdentifier* is shown in Figure 2. In *PIdentifier*, (a) we first collect and extract the user profiles and user posts from Hack Forums using our developed web crawling tools; then (b) we construct an AHIN to model the multi-typed entities and their rich relations and devise different metagraphs to measure the relatedness between buyers and products. Based on the constructed AHIN, (c) metagraph aggregated heterogeneous graph neural network (i.e., *mHGNN*) is proposed to learn the buyer and product representations by attentively propagating and aggregating information based on different metagraphs; (d) the learned representations will then be used for illicit traded product identification. We will introduce the proposed method for each component in detail below.

B. AHIN Construction

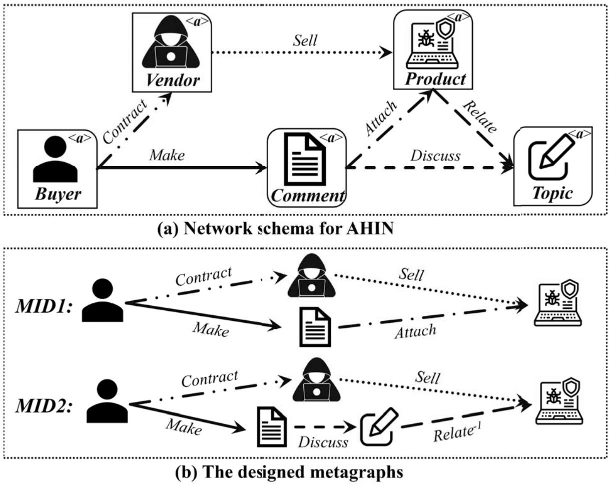
In our work, for illicit traded product identification in Hack Forums, we consider five types of entities (i.e., buyer, vendor, product, comment and post topic) and six types of relations among them.

Entity Feature Extraction. To describe the user (i.e., buyer and vendor), we extract features from user profile for representation, which include: i) *popularity*: it indicates how popular a user is in Hack Forums. A user can give different users different points to promote their popularity based on his/her rank in Hack Forums. For example, a user with the rank of *Ub3r* can give another user -3 to 3 points. ii) *business rating*: this is user review on a purchased product in a completed contract. There are three types of reviews in Hack Forums - *Booyah* (i.e., positive), *Bleh* (i.e., neutral) and *Boo* (i.e., negative). iii) *Bytes*: it implies how active a user is in Hack Forum. For example, a user can gain 3 Bytes if he/she posts a thread and 1 Byte for a reply; a user can also obtain Bytes from others' donations.

We also extract the numbers of completed contracts, open disputes, posted threads and comments for each user. Thus, a given user will be represented by a attribute feature vector including the extracted information above. For each product (i.e., thread) and comment, we apply *doc2vec* [7] to convert the text content of variant size into a fixed-length feature vector (200-dimension in our case). We also perform Latent Dirichlet Allocation [8] to extract topics from the posts (i.e., products or comments). Each topic will then be described by a unique one-hot representation.

Relation Feature Extraction. Six types of relations are further extracted to depict the rich relations among different types of entities: (*R1*) the *buyer-contract-vendor* relation denotes a contract made between a buyer and a vendor; (*R2*) the *vendor-sell-product* relation indicates a vendor sells or advertises a product; (*R3*) the *buyer-make-comment* relation depicts a buyer makes a comment; (*R4*) the *comment-attach-product* relation describes a comment is made on a product; (*R5*) the *comment-discuss-topic* relation denotes a comment discusses a specific topic; and (*R6*) *product-relate-topic* relation indicates a product is related to a specific topic.

Based on the definition in Section II, the network schema for AHIN in our application is shown in Figure 3.(a), where each entity is attached with an attribute feature vector.



(a) Network schema for AHIN

(b) The designed metagraphs

Figure 3. Network schema and metagraphs for AHIN.

C. Metagraph Representation

After further investigation of the underground market, we have the following observations: (1) A buyer who has a contract with a vendor in terms of a traded product is likely to comment on this product (either asking problems or providing reviews). For example, the buyer “T***7” and “b***4”, who sign contracts with the vendor “A***r” to purchase *ALPHA KEYLOGGER*, both comment on the product discussing their purchase experiences. (2) Before initiating a contract, a buyer may express his/her purchase intent in the market (e.g., in the sections of *Buyers Bay* and *Marketplace Discussions* in Hack Forums). Towards the above findings, we apply the concept of metagraph, which enables us to capture a more complex relationship between entities in AHIN than metapath, to formulate the relatedness between a buyer and the traded product sold by a vendor in a contract. As shown in Figure 3.(b), we design two metagraphs *MID1* and *MID2* and their corresponding reverse metagraphs *MID1*$^-$ and *MID2*$^-$. For example, *MID1* denotes that a buyer and a product is connected if the buyer has a contract with a vendor to purchase the product and also makes a comment on this product.

D. Metagraph Aggregated Heterogeneous GNN

To solve the illicit traded product identification problem, we exploit graph neural network (GNN) based models. There have been many studies on GNN models for (A)HIN or knowledge graph [9]–[12]. Although the results of these existing works are promising, they either fail to adapt specific metagraph for information aggregation or are incapable of distinguishing semantics of different types of relation during the propagation and aggregation process. To address these issues, we propose a novel heterogeneous graph neural network, named *mHGNN*, to attentively propagate and aggregate the neighborhood information across different metagraphs to learn node representations in AHIN for the identification problem. The proposed *mHGNN* is a three-step learning model: metagraph-guided neighbor search, attentive propagation and aggregation, and multi-view fusion.

Metagraph-guided Neighbor Search. Given an AHIN $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ with entity sets: buyer $\mathcal{B}$, vendor $\mathcal{S}$, product $\mathcal{P}$, comment $\mathcal{C}$, topic $\mathcal{K}$, and a metagraph scheme $M \in \mathcal{M}$, we design a neighbor search mechanism to retrieve the node's neighbors guided by the given metagraph. This is a two-step backward lookup procedure. Without loss of generality, we use M=*MID1* as an example to illustrate the proposed search algorithm (note that it can be readily applicable for *MID2*). For each product $p \in \mathcal{P}$, the 1st-step backward search aims to find its *comment-vendor* neighbors:

$$\mathcal{Z} = \{(c, s) | \forall c \in \mathcal{C}, s \in \mathcal{S} : \Pr(c|p) \cdot \Pr(s|p) = 1\}, \quad (1)$$

where a *comment-vendor* pair (c, s) is constrained by the joint probability indicating $(c, p) \in \mathcal{E}$ and $(s, p) \in \mathcal{E}$. For example, as shown in Figure 4, given *Product-1*, through 1st-step search, we obtain its *comment-vendor* neighbors $\mathcal{Z}$ $=\{$(*Comment-1,Vendor-1*), (*Comment-2,Vendor-1*), (*Comment-3,Vendor-1*)$\}$. And then, the 2nd-step lookup finds a *buyer-comment-vendor* neighbors:

$$\mathcal{Z}' = \{(b, \mathcal{Z}) | \Pr(b|c) \cdot \Pr(b|s) = 1\}. \quad (2)$$

In our application, we restrict each *comment-buyer* pair as a one-to-one relation; therefore, we have $\Pr(b|c) = \Pr(c)$ and define a function $\pi : \mathcal{C} \to \mathcal{B}$ to map the comment to its

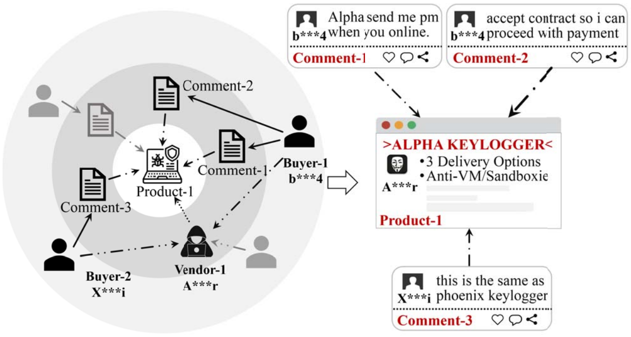


Figure 4. Example of metagraph-guided neighbors.

corresponding buyer. Hence, Eq. (2) can be rewritten as:

$$\mathcal{Z}' = \{(\pi(c), \mathcal{Z}) | \Pr(c) \cdot \Pr(\pi(c)|s) = 1\}. \tag{3}$$

Using the same example as shown in Figure 4, given *Product-1*, through the 2nd-step lookup, we can have $\mathcal{Z}'$ ={(*Buyer-1,Comment-1,Vendor-1*), (*Buyer-1,Comment-2,Vendor-1*), (*Buyer-2,Comment-3,Vendor-1*)}. Based on $\mathcal{Z}'$, the 1st-order neighbors of c, s and p in terms of M are:

$$\begin{aligned}\mathcal{N}(c) &= \big\{\pi(c)|\forall c : \big(\pi(c), c\big) \in \mathcal{Z}'\big\},\\ \mathcal{N}(s) &= \big\{\pi(c)|\forall c : \big(\pi(c), s\big) \in \mathcal{Z}'\big\},\\ \mathcal{N}(p) &= \big\{c, s|\forall(c, s) \in \mathcal{Z}'\big\}.\end{aligned} \tag{4}$$

Note that, for metagraph in the form of *MID2*, we can obtain $\mathcal{Z}$ using $\{\mathcal{C}, \mathcal{K}\}$ in place of $\mathcal{C}$, while rguaranteeing $p(c|p) = 0$ because the topic relates to either product or comment $p(b|k) \equiv 0$ for $\mathcal{Z}'$.

Attentive Propagation and Aggregation. Before performing information propagation and aggregation, an embedding layer $l : \mathbb{R}^{d_a} \rightarrow \mathbb{R}^d$ is first applied to map each entity's attributed feature vector of d_a dimension to a d-dimensional distributional vector, where l is a parameterized function (e.g., multilayer perceptron layers (MLP)). Intuitively, in our application, the neighbor who contributes more to the illicit product trading should gain more attention during aggregation. For example, as illustrated in Figure 4, for *Product-1*'s 1st-order neighbors (i.e., *Comment-1, 2, 3*), compared with *Comment-1* and *Comment-3*, *Comment-2* should gain more attention as it implies that *Buyer-1* has initiated a contract with *Vendor-1* to purchase *Product-1*. To allocate varying importance to neighbors, we first represent each relation type $r \in \mathcal{R}$ in AHIN by a relation-specific transformation $\mathbf{A} \in \mathbb{R}^{d \times d}$. Such that the attentive weights $\boldsymbol{\beta}$ of entity u (the neighbor of v) refer to the trading relevance of these two entities measured in the relation space $\boldsymbol{\alpha} \in \mathbb{R}^{d \times d \times |\mathcal{R}|}$ tensorized from $\mathbf{A}$, that is,

$$\boldsymbol{\beta}(v, u) = \mathbf{E}_v^T \boldsymbol{\alpha} \mathbf{E}_u, \quad \boldsymbol{\alpha} = \biguplus_{|\mathcal{R}|} \mathbf{A}. \tag{5}$$

Inspired by [13], to ease the computational cost during inference, we can factorize $\mathbf{A} = \mathbf{L} \times \mathbf{R}$ with $\mathbf{L} \in \mathbb{R}^{d \times d'}$ and $\mathbf{R} \in \mathbb{R}^{d' \times d}$ ($d' < d$). Then, $\boldsymbol{\beta}_i(v, u)$ can be reformulated as an inner product $\langle \mathbf{L}^T \mathbf{E}_v, \mathbf{R} \mathbf{E}_u \rangle$, where $\mathbf{L}^T \mathbf{E}_v \in \mathbb{R}^{d'}$ can be regarded as the projection of v from entity space to relation space; while $\mathbf{E}_u \mathbf{R} \in \mathbb{R}^{d'}$ as the projection of u. We then normalize the weight across all the 1st-order neighbors of v by applying softmax function:

$$\widetilde{\boldsymbol{\beta}}(v, u) = \frac{\exp(\boldsymbol{\beta}(v, u))}{\sum_{u' \in \mathcal{N}(v)} \exp(\boldsymbol{\beta}(v, u'))}. \tag{6}$$

To characterize the topological structure of entity v, we compute the linear combination of v's neighbors:

$$\mathbf{E}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} \widetilde{\boldsymbol{\beta}}_i(v, u) \mathbf{E}_u, \tag{7}$$

where the weight $\widetilde{\boldsymbol{\beta}}_i(v, u)$ determines how much information propagated from u to v in terms of i-th relation space. Finally, to aggregate v's representation $\mathbf{E}_v$ and its neighbors' representations $\mathbf{E}_{\mathcal{N}(v)}$, without loss of generality, we implement the sum aggregator $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, which sums these two representations up, followed by a nonlinear transformation:

$$\mathbf{E}_v = \sigma\big(\mathbf{W}(\mathbf{E}_v + \mathbf{E}_{\mathcal{N}(v)}) + \mathbf{b}\big), \tag{8}$$

where σ is the activation function (e.g., LeakyReLU [14]).

Similar to metagraph-guided neighbor search, the attentive propagation and aggregation is also a two-step process: in the first step, based on $\mathcal{N}(c)$ and $\mathcal{N}(s)$ defined in Eq. (4), it generates the embeddings $\mathbf{E}_c$ and $\mathbf{E}_s$ for comment c and vendor s by Eq. (7) and Eq. (8) respectively; in the second step, it learns the embedding $\mathbf{E}_p$ of product p based on $\mathcal{N}(p)$ as well as the generated $\mathbf{E}_c$ and $\mathbf{E}_s$.

Multi-view Fusion. Given a metagraph M, by applying the above proposed attentive propagation and aggregation method, we are able to learn the product embeddings. The buyer embeddings can be yielded in the similar way by using M's corresponding reverse metagraph M^-. Thus, a single layer of *mHGNN* consists of applying propagation and aggregation on M and M^- separately. The *mHGNN* can also be extended to multi-layer setting by stacking more propagation layers, which assembles the information from higher-order metagraph based neighbors. Formally, in h-th layer of *mHGNN*, entity v's embedding is calculated as:

$$\begin{aligned}\mathbf{E}_v^h &= \sigma\big(\mathbf{W}(\mathbf{E}_v^{h-1} + \mathbf{E}_{\mathcal{N}(v)}^{h-1}) + \mathbf{b}\big),\\ \mathbf{E}_{\mathcal{N}(v)}^{h-1} &= \sum_{u \in \mathcal{N}(v)} \widetilde{\boldsymbol{\beta}}_i^{h-1}(v, u) \mathbf{E}_u^{h-1}.\end{aligned} \tag{9}$$

For the k metagraphs $\{M_i\}_{i=1}^k$, as different metapaths depict the relatedness over entities in different views, we obtain a fused product embedding $\mathbb{E}_p$ by concatenating each embedding generated based on a specific metagraph; in the same manner, a buyer embedding $\mathbb{E}_b$ is fused based on different reverse metagraphs $\{M_i^-\}_{i=1}^k$:

$$\mathbb{E}_p = \bigoplus_{i=1}^k \mathbf{E}_p^{M_i}, \quad \mathbb{E}_b = \bigoplus_{i=1}^k \mathbf{E}_b^{M_i^-}. \tag{10}$$

Algorithm 1: *mHGNN*

```
Input: AHIN $\mathcal{G}$, metagraphs $\mathcal{M}$.
Output: Prediction function $f(b, p|\mathcal{G}, \mathcal{M}; \Theta)$.
/* Iterate over training samples */
foreach $(b, p) \in \mathcal{Y}^+ \cup \mathcal{Y}^-$ do
    /* Multiple metagraphs */
    foreach $M \in \mathcal{M}$ do
        $M^- \leftarrow \varphi(M)$;
        /* Multi-layer setting */
        for $h = 1, ..., H$ do
            $\mathbf{E}_p^h \leftarrow$Attn-Layer $(p, M)$;
            $\mathbf{E}_b^h \leftarrow$Attn-Layer $(b, M^-)$;
        end
        $\mathbf{E}_p^M \leftarrow \mathbf{E}_p^H$, $\mathbf{E}_b^{M^-} \leftarrow \mathbf{E}_b^H$;
    end
    $\mathbb{E}_p, \mathbb{E}_b \leftarrow$Concatenate embeddings by Eq. (10);
    $\hat{y}_{bp} \leftarrow$ Calculate the predict score by Eq. (11);
    Update parameters by Eq. (12);
end
/* Single layer aggregation */
Function Attn-Layer($v$, $M$):
    $\mathcal{N}(v) \leftarrow$ Collect neighbors via Eq. (1)-Eq. (4);
    $\mathbf{E}_{\mathcal{N}(v)}^{h-1} \leftarrow$ Combine neighbor info by Eq. (7);
    $\mathbf{E}_v^h \leftarrow$ Aggregate with itself via Eq. (8);
    return $\mathbf{E}_v^h$;
End Function
```

E. Learning Algorithm

Through *mHGNN*, we obtain the fused embeddings of buyer $\mathbb{E}_b$ and product $\mathbb{E}_p$. Then, we concatenate $\mathbb{E}_b$ and $\mathbb{E}_p$, and feed it into MLP layers $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, followed by a sigmoid layer to get the prediction score:

$$\hat{y}_{bp} = sigmoid\big(f(\mathbb{E}_b \oplus \mathbb{E}_p)\big), \tag{11}$$

where $\hat{y}_{bp}$ is the probability that buyer b purchases product p in a given contract. The loss function is designed as follows:

$$\mathcal{L} = \sum_{(b,p) \in \mathcal{Y}^+ \cup \mathcal{Y}^-} J(y_{bp}, \hat{y}_{bp}) + \gamma ||\Theta||_2^2, \tag{12}$$

where J measures the cross-entropy loss between ground truth y_{bp} and the predicted score $\hat{y}_{bp}$, $||\Theta||_2^2$ is the L2-regularizer to prevent over-fitting. The learning algorithm of *mHGNN* is given in Algorithm 1. For prediction, given a private contract associated with buyer b and vendor s, we calculate the probabilities $\hat{y}_{bp}$ of all (b, p) pairs (p posted by s) and the one with the highest probability is identified as the traded product in the given contract. The time complexity of a single attentive aggregation layer on a single metagraph scheme is $O(|\mathcal{E}|d^2)$, where $|\mathcal{E}|$ is the number of edges in AHIN. For H layers and k metagraphs, the overall complexity of *mHGNN* is $O(kH|\mathcal{E}|d^2)$.

IV. Experimental Results and Analysis

In this section, we conduct four sets of experimental studies to fully evaluate the performance of our developed *PIdentifier* framework: (1) the first set of experiments is to evaluate the proposed *mHGNN* in comparison with the state-of-the-art network embedding and GNN-based models; (2) in the second set of experiments, we examine how different components affect *mHGNN*; (3) the third set of experiment is to compare *PIdentifier* with other alternative machine learning methods for illicit traded product identification; (4) in the final set of experiments, we investigate how different choices of hyper-parameters affect the model performance.

A. Experimental Setup

Data Collection and Preparation. In this work, we mainly focus on Hack Forums, which is one of the largest and most prevalent underground markets. We first develop a set of crawling tools to collect the trading contracts in a primary marketplace (i.e., *Premium Tools and Programs* under *Premium Sellers Section*) in Hack Forums during Jun. 10, 2018 to Nov. 18, 2019. Subsequently, 34,046 trading contracts are collected, including 28,432 private contracts and 5,614 public contracts. Note that, in this work, we won't consider the case that a vendor has multiple contracts with the same buyer and vice verse (i.e., 245 contracts are disregarded). In the remaining 5,369 public contracts, the traded products in 2,688 public contracts are inaccessible as the links of the product threads are not provided (i.e., these can be treated as private contracts). After the preprocession, we finally have 2,681 public contracts where traded products are visible as ground truth. In the experiments, we extract the *buyer-product* pairs in these 2,681 public contracts as positive samples while randomly match each buyer to a non-contracted product sold by the same vendor to compose the negative samples.

Baseline Methods. We compare our proposed *mHGNN* with following state-of-the-art baselines, including network embedding methods and GNN-based models.

- **DeepWalk** [15] performs random walk and skip-gram to learn node embeddings in homogeneous network.
- **metapath2vec** [16] learns HIN representations by applying metapath guided random walk and skip-gram model.
- **metagraph2vec** [17], similar to metapath2vec, is also a HIN embedding model but learns node embeddings by leveraging metagraph guided random walk.
- **GCN** [18] is a semi-supervised graph convolutional network that averages neighbors' embeddings with linear projection.
- **GAT** [19] is a graph attention network that aggregates information of neighbors via self-attention mechanism.
- **RGCN** [20] is designed for heterogeneous graph and considers different relations between nodes for information aggregation.

- **HAN** [21] is a heterogeneous graph attention network model aggregating the information from neighbors through node-level attention and semantic-level attention.
- **MEIRec** [10] is a heterogeneous graph neural network model that aggregates information of metapath-guided neighbors in HIN via different aggregation functions.

For network embedding methods (i.e., DeepWalk, metapath2vec, metagraph2vec), since they are incapable of dealing with the attributes attached on the nodes, we concatenate the attributed feature vector with the learned node embeddings and then feed them into a three-layer MLP for training and prediction. The parameters of these methods are set as follows: the number of walks per node is 50, the walk length is 100, the window size is 5 and the number of negative instances is 5. For GNN-based models that are designed for homogeneous network (i.e., GCN, GAT), we first construct two corresponding metapaths (i.e., *buyer-comment-product* and *buyer-comment-topic-product*) according to the designed metagraphs, and then transform AHIN to the corresponding homogeneous graph based on each metapath, later apply GCN and GAT on each homogeneous graph. Here we test all the metapaths for GCN and GAT, and report the best performances. HAN and MEIRec employ the constructed metapaths to learn node embeddings. The embedding dimension is fixed to 100 for all baselines.

Parameter Settings for *mHGNN*. In *mHGNN*, we set feature embedding layer l as two-layer MLP, prediction function f as three-layer MLP, embedding dimension $d = 100$, model depth $H = 1$. The model parameters are initialized using Xavier initializer [22], and then optimized by performing Adaptive Moment Estimation (Adam) with learning rate of 0.002 and L2 regularization $\gamma = 10^{-4}$. For other parameters, we set dropout ratio to 0.5, epochs to 500, and using LeakyReLU as the activation function.

Evaluation Metrics. We use precision, recall, accuracy (ACC) and F1 to validate the effectiveness of our proposed model. We randomly split the dataset into training set, validation set and test set with a ratio of 0.6:0.2:0.2. The validation set is used to tune the hyper-parameters. For each model, we report the average performance in terms of precision, recall, ACC and F1 on 5 repeated processes.

B. Comparison with Baselines

In this section, we compare the performance of *mHGNN* with the above baselines for illicit traded product identification in Hack Forums. Based on the experimental results shown in Table I, we can see that:

- Our proposed *mHGNN* consistently outperforms all baselines. The reasons behind this are that (1) the designed metagraphs used in *mHGNN* are more expressive than metapaths in characterizing complex and comprehensive relations between nodes; (2) *mHGNN* explores metagraphs to retrieve node's neighbors, and further considers different semantics of different relation types to weight the neighborhood information during propagation and aggregation, and thus is able to achieve better performance.

Table I
COMPARISON OF *mHGNN* AND BASELINES IN ILLICIT TRADED PRODUCT IDENTIFICATION.

Method	Precision	Recall	ACC	F1
DeepWalk	0.7916	0.7877	0.7901	0.7896
metapath2vec	0.8206	0.8190	0.8200	0.8198
metagraph2vec	0.8402	0.8340	0.8377	0.8371
GCN	0.8341	0.8425	0.8375	0.8383
GAT	0.8447	0.8463	0.8453	0.8455
RGCN	0.8723	0.8638	0.8687	0.8680
HAN	0.8890	0.9026	0.8950	0.8957
MEIRec	0.8950	0.8813	0.8890	0.8881
mHGNN	**0.9170**	**0.9239**	**0.9201**	**0.9204**

- For GNN-based models, (1) RGCN, HAN and MEIRec, which are designed for heterogeneous graph, could preserve richer semantics and thus obtain better results than GCN and GAT; (2) HAN and MEIRec which incorporate metapath scheme for information aggregation perform better than RGCN; (3) HAN and GAT utilizing attention to weight the information propagated from neighbors yield better performances than GCN and RGCN.
- Generally, GNN-based models (i.e., GCN, GAT, RGCN, HAN and MEIRec) which combine the node attributes and structural information in a more comprehensive manner achieve better performance than traditional netwrok embedding methods (i.e., DeepWalk, metapath2vec and metagraph2vec) in illicit traded product identification.
- For traditional network embedding methods, metagraph2vec achieves a better performance than metapath2vec and DeepWalk. This is because that metagraph2vec leverages the same metagraphs designed for *mHGNN* to guide the random walk for path generation and subsequent learn higher quality node embeddings than metapath2vec and DeepWalk.

C. Ablation Study of mHGNN

In this set of experiments, we conduct ablation study to examine how different components (i.e., different metagraph schemes, attentive weighting, and aggregator selection) affect the performance of *mHGNN*. We prepare different variants of *mHGNN* as follows.

- **Metagraph variants**: We design two metagraphs to measure the relatedness of buyers and products sold by vendors in given contracts. To further explore the impact of each metagraph, we consider two variants of *mHGNN*, denoted as $mHGNN_{MID1}$ and $mHGNN_{MID2}$, which either takes *MID1* or *MID2* into consideration during propagation and aggregation process.
- **Weighting variant**: In this setting, we disable the attentive weighting mechanism and simply average the

information gathered from metagraph based neighbors. This variant is denoted as *mHGNN*$_{att-}$.

- **Aggregator variants**: Besides the sum aggregator implemented in *mHGNN*, we also prepare another two aggregator candidates for comparison: concatenation aggregator (i.e., concatenating itself and its neighbors for aggregation) and neighbor aggregator (i.e., merely considering neighborhood information for aggregation without taking the input from the node itself), denoted as *mHGNN*$_{cat}$ and *mHGNN*$_{nbr}$.

Table II
Comparison of *mHGNN* and its variants.

Variant	Precision	Recall	ACC	F1
mHGNN$_{MID1}$	0.9093	0.9060	0.9078	0.9076
mHGNN$_{MID2}$	0.8767	0.8862	0.8808	0.8814
mHGNN$_{att-}$	0.8648	0.8616	0.8634	0.8632
mHGNN$_{cat}$	0.9194	0.9190	0.9192	0.9192
mHGNN$_{nbr}$	0.8870	0.8936	0.8899	0.8903
mHGNN	**0.9170**	**0.9239**	**0.9201**	**0.9204**

The comparison of *mHGNN* and its variants are shown in Table II, from which we can see that:

- For metagraph variants, *mHGNN* integrating both two metagraph schemes (i.e., MID1 and MID2) for prediction outperforms *mHGNN*$_{MID1}$ and *mHGNN*$_{MID2}$ which only considers either MID1 or MID2. Moreover, *mHGNN*$_{MID1}$ performs better than *mHGNN*$_{MID2}$, which may because $MID1$ possesses more powerful capability of characterizing trading patterns compared with $MID2$.
- *mHGNN* using attentive weighting mechanism for information propagation obtains better results than *mHGNN*$_{att-}$ that simply averages the neighbors' information. This demonstrates the importance of weighting different neighbors based on different relation types during information propagation and aggregation.
- For the comparison of three aggregators, *mHGNN*$_{cat}$ achieves comparable performance with *mHGNN* that applies sum aggregator. However, both *mHGNN* and *mHGNN*$_{cat}$ outperform *mHGNN*$_{nbr}$. One possible reason is that discarding the entity's own information during aggregation may diminish the embedding quality.

D. Comparison with Alternative Approaches

In this section, we compare our developed *PIdentifier* framework with alternative machine learning approaches. Here, we construct two types of features: (1) content-based features (*f-1*): for buyer, we concatenate his extracted attributes with the post embedding learned via *doc2vec*; for product, we directly utilize its content embedding learned via *doc2vec* as the feature vector. (2) augmented features (*f-2*): for buyer, we augment *f-1* with relation-based features of *R1* and *R3*; for product, we augment *f-1* with relation-based features of *R2* and *R4*. Fianlly, we sum up the feature vectors of buyer and product, and then feed it into two typical classification models, SVM (i.e., we use LibSVM and the penalty is empirically set to 10 while other parameters are set by default.) and a three-layer MLP (using same parameters in *mHGNN*) for training and prediction. From the experimental results shown in Table III, we can see that:

- Augment features (*f-2*) performs better than content-based features (*f-1*), which indicates relation-based features added by *f-2* help the performance of machine learning as the rich semantics encoded in relations can bring more information;
- *mHGNN* explores AHIN representation to model content-based and relation-based features in a more expressive and comprehensive manner, and thus significantly improves the performance in illicit traded product identification.

Table III
Comparison with other approaches.

Method	SVM		MLP		*mHGNN*
Index	*f-1*	*f-2*	*f-1*	*f-2*	-
Precision	0.7173	0.7315	0.7500	0.7613	**0.9170**
Recall	0.7269	0.7306	0.7530	0.7605	**0.9239**
ACC	0.7202	0.7312	0.7510	0.7610	**0.9201**
F1	0.7221	0.7311	0.7515	0.7609	**0.9204**

E. Hyper-parameter Sensitivity

In this section, we conduct hyper-parameter sensitivity analysis of how different choices of dimension d and the model depth H affect the performance of *PIdentifier*.

From the results shown in Figure. 5, we can observe that: (1) When d increases from 50 to 250, the performance improves since more information is preserved for a large d, thus better representations can be learned. And, the performance inclines to be stable when d reaches to 200. (2) Increasing the model depth H ($H = 2, 3$) slightly is capable of boosting the performance; however, the performance decreases when considering a large $H = 4, 5$ as more noises (e.g., unrelated neighbors) could be brought into the model.

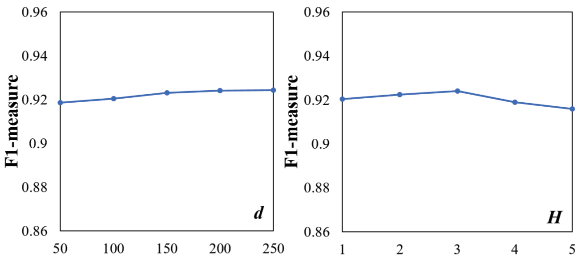


Figure 5. Parameter sensitivity evaluation.

F. Case Study

In this section, we apply *PIdentifier* for illicit traded product identification in the wild. We randomly select 600 private contracts from our data collection and use *PIdentifier* to identify the possible traded products in these contracts for further analysis. Among the identified products, we choose two of the most popular types of products for discussion, i.e., bot-related products with 126 contracts and social media hacking tools with 118 contracts. Figure 6 illustrates their contract distributions from June 2018 to Nov. 2019 in the primary marketplace (i.e., *Premium Tools and Programs* under *Premium Sellers Section*) in Hack Forums. Successfully identifying the traded products in private contracts would facilitate the market trend prediction and market scale estimation. For example, the advertised price of a bot is from \$15 to \$30 per month, thus the revenue of 34 identified bot-related products in Mar. 2019 was between \$510 and \$1,020 in this marketplace. Our developed system *PIdentifier*, which can be readily applicable to identify illicit traded products in various underground markets, will facilitate defenders and law enforcement to better understand the dynamics of illicit activities in underground markets and thus devise effective interventions to combat evolving cybercrimes.

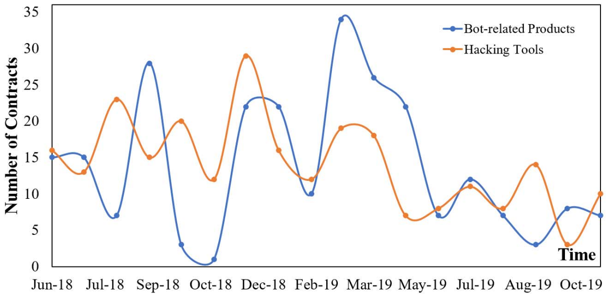


Figure 6. The contract distribution of two popular identified products.

V. Related Work

Underground Market Analysis. To combat the cybercrimes that have become increasingly dependent on the underground markets, various research efforts have focused on underground market analysis [23]–[27]. For example, in underground market user analysis domain, Zhang et al. [23] leverages writing and photography styles for drug trafficker identification; Abbasi et al. [27] characterizes users with content features and structural features and perform k-means clustering algorithm to identify expert hackers in Hacker Forums; other advanced techniques including deep learning are developed to profile sellers from their advertisements. Different from the existing works, in this paper, we propose and develop an intelligent framework (named *PIdentifier*) to automate the analysis of underground market (i.e., Hack Forums) for the identification of illicit traded products in private contracts at the first attempt.

Heterogeneous Information Network. HIN has been intensively studied and applied to various applications [17], [28]–[31]. Typically, HIN is used to model different types of entities and relations. Several studies have already investigated to measure the relevance over HIN entities, including path-based methods (e.g., metapath [28]) and structure-based methods (e.g., meta-structure [32] and metagraph [6]). However, HIN has limited capability of modeling additional attributes of entities, to tackle this challenge, attributed HIN (AHIN) [5] is then proposed to enrich the HIN by attaching individual feature vector to each entity.

Graph Neural Network. In recent years, there have been ample works on GNN-based models [9]–[12], [21]. The basic idea of GNN is to aggregate information from node' neighbors via neural networks. For example, GCN [18] averages the neighbors' embeddings while GAT [19] exploits self-attention mechanism to aggregate the neighbors' information. To deal with the heterogeneous property of HIN, several heterogeneous GNN models are proposed which models the heterogeneity by using metapath (e.g., MEIRec [10] and HAN [21]) or metagraph (Meta-GNN [12]). However, when applying these works to our application, they either fail to adapt specific meta-structure for information aggregation or are incapable of distinguishing semantics of different types of relation during the propagation and aggregation process.

VI. Conclusion

To gain deep insights into the dynamics of trading activities in underground markets, in this paper, we design and develop an intelligent framework named *PIdentifier* for identification of illicit traded products in private contracts. In *PIdentifier*, based on the large-scale extracted user profiles, user posts and different types of relations within the complex ecosystem, we introduce an AHIN to model rich senmantics and complex relations among vendors, buyers, products, comments and topics; and then design different metagraphs to formulate the relatedness between buyers and products. Based on the constructed AHIN, we further propose a heterogeneous GNN model (*mHGNN*) to attentively propagate and aggregate the information guided by our designed metagraphs to learn node representations for illicit traded product identification. Comprehensive experimental studies are conducted on the real-world data collected from Hack Forums. Promising results demonstrate that *PIdentifier* outperforms the state-of-the-art baselines in illicit traded product identification in underground market.

Acknowledgment

Y. Fan, Y. Ye, Q. Peng, J. Zhang and Y. Zhang's work is partially supported by the NSF under grants IIS-2027127, IIS-2040144, IIS-1951504, CNS-2034470, CNS-1940859, CNS-1946327, CNS-1814825, OAC-1940855 and ECCS-2026612, the DoJ/NIJ under grant NIJ 2018-75-CX-0032.

References

[1] Hackforums, https://hackforums.net/.

[2] Nulled, https://www.nulled.to/.

[3] Blackhatworld, https://www.blackhatworld.com/.

[4] A. Haslebacher, J. Onaolapo, and G. Stringhini, "All your cards are belong to us: Understanding online carding forums," in *APWG Symposium on Electronic Crime Research*. IEEE, 2017, pp. 41–51.

[5] X. Li, Y. Wu, M. Ester, B. Kao, X. Wang, and Y. Zheng, "Semi-supervised clustering in attributed heterogeneous information networks," in *WWW*, 2017, pp. 1621–1629.

[6] H. Jiang, Y. Song, C. Wang, M. Zhang, and Y. Sun, "Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks," in *IJCAI*, 2017, pp. 1944–1950.

[7] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014, pp. 1188–1196.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[9] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *CIKM*, 2018, pp. 417–426.

[10] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, "Metapath-guided heterogeneous graph neural network for intent recommendation," in *KDD*, 2019, pp. 2478–2486.

[11] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *KDD*, 2019, pp. 950–958.

[12] A. Sankar, X. Zhang, and K. C.-C. Chang, "Meta-gnn: metagraph neural network for semi-supervised learning in attributed heterogeneous information networks," in *ASONAM*, 2019, pp. 137–144.

[13] S. Abu-El-Haija, B. Perozzi, and R. Al-Rfou, "Learning edge representations via low-rank asymmetric projections," in *CIKM*, 2017, pp. 1787–1796.

[14] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013, p. 3.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.

[16] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017, pp. 135–144.

[17] Y. Fan, S. Hou, Y. Zhang, Y. Ye, and M. Abdulhayoglu, "Gotcha-sly malware! scorpion a metagraph2vec based malware detection system," in *KDD*, 2018, pp. 253–262.

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

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[20] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*. Springer, 2018, pp. 593–607.

[21] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *(AISTATS*, 2010, pp. 249–256.

[23] Y. Zhang, Y. Fan, W. Song, S. Hou, Y. Ye, X. Li, L. Zhao, C. Shi, J. Wang, and Q. Xiong, "Your style your identity: Leveraging writing and photography styles for drug trafficker identification in darknet markets over attributed heterogeneous information network," in *WWW*, 2019, pp. 3448–3454.

[24] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, "Key player identification in underground forums over attributed heterogeneous information network embedding framework," in *CIKM*, 2019, pp. 549–558.

[25] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, "Crimebb: Enabling cybercrime research on underground forums at scale," in *WWW*, 2018, pp. 1845–1854.

[26] R. S. Portnoff, S. Afroz, G. Durrett, J. K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson, "Tools for automated analysis of cybercriminal markets," in *WWW*, 2017, pp. 657–666.

[27] A. Abbasi, W. Li, V. Benjamin, S. Hu, and H. Chen, "Descriptive analytics: Examining expert hackers in web forums," in *ISI*. IEEE, 2014, pp. 56–63.

[28] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[29] 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 *KDD*, 2018, pp. 1531–1540.

[30] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, "Hindroid: An intelligent android malware detection system based on structured heterogeneous information network," in *KDD*, 2017, pp. 1507–1515.

[31] Y. Fan, Y. Zhang, S. Hou, L. Chen, Y. Ye, C. Shi, L. Zhao, and S. Xu, "idev: Enhancing social coding security by cross-platform user identification between github and stack overflow." in *IJCAI*, vol. 19, 2019, pp. 2272–2278.

[32] Z. Huang, Y. Zheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, "Meta structure: Computing relevance in large heterogeneous information networks," in *KDD*, 2016, pp. 1595–1604.