Information Sciences 653 (2024) 119770

Contents lists available at ScienceDirect

Information Sciences

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

Edge propagation for link prediction in requirement-cyber threat intelligence knowledge graph

Yang Zhang [a], Jiarui Chen [b], Zhe Cheng [a], Xiong Shen [a], Jiancheng Qin [b], Yingzheng Han [b,*], Yiqin Lu [b,1]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China
[b] School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China

ARTICLE INFO

Keywords:
Critical information infrastructure
Cyber threat intelligence
Graph neural network
Link prediction
Knowledge graph

ABSTRACT

Critical information infrastructure (CII) is a critical component of national socioeconomic systems and one of the primary targets of cyberattacks. Unfortunately, CII's security administration struggles to keep up with the rapidly evolving and complex cyber threats. In this research, we combine cybersecurity threat intelligence (CTI) with management security requirements (SR) data to construct a knowledge graph (KG) named RCTI and predict new knowledge on the heterogeneous graph. In addition, we propose EGNN, a novel GNN-based model that defines the representation of edges and develop an algorithm for propagating edge information. Experiments on three public datasets and the RCTI graph show that the EGNN achieves state-of-the-art performance. Finally, we use the EGNN model to predict new links on the RCTI graph, which by manual analysis achieves a 97% connectivity rate between the CTI and SR entities. Therefore, the EGNN can effectively detect management vulnerabilities and enhance CII's cybersecurity capability in the event of cybersecurity incidents.

1. Introduction

Critical information infrastructure (CII) is the central nervous system of state economic and social operations, and the national security, economy and livelihood of people will be seriously endangered once it is destroyed, disabled or suffers from data leakage. Due to the critical role of CII in national security, it has become the primary target of contemporary network attacks such as advanced persistent threat (APT) [2] and denial of service (DOS) [17]. The majority of nations and regions have adopted stringent CII management and supervision criteria. These security requirements (SRs) comprise quantitative performance indicators and standards for organizational management, such as authentication, authorization, accountability, and access control. Due to the specific and complicated characteristics of the SRs, it is challenging for managers to verify the compliance of organizations with security requirements and to enhance their capacity to prevent cyber threats.

* Corresponding author at: School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China.
E-mail addresses: cszhangyang@mail.scut.edu.cn (Y. Zhang), ee_jrchen@mail.scut.edu.cn (J. Chen), cszcheng@mail.scut.edu.cn (Z. Cheng), 201710105867@mail.scut.edu.cn (X. Shen), jcqin@scut.edu.cn (J. Qin), hanyz@scut.edu.cn (Y. Han), eeyqlu@scut.edu.cn (Y. Lu).
[1] IEEE Member.

https://doi.org/10.1016/j.ins.2023.119770
Received 30 March 2023; Received in revised form 6 October 2023; Accepted 8 October 2023

Available online 12 October 2023

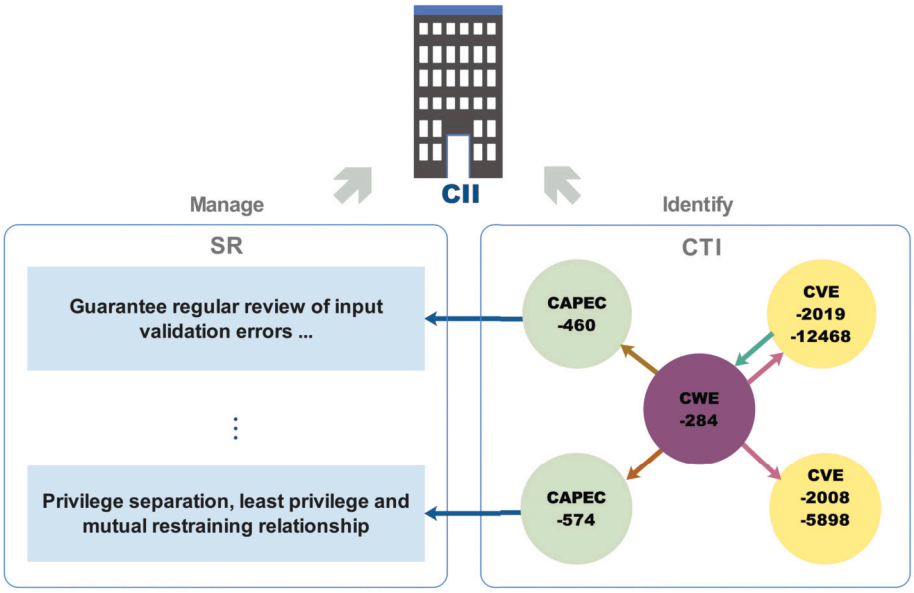

Fig. 1. An example of how CTI and SR collectively protect the CII. CTI is threat intelligence information released by cybersecurity practitioners to identify attacks on CII, whereas SR is the management requirement for CII. The security capability of CII can be effectively improved by associating SR with CII data.

Cyber threat intelligence (CTI) is the open exchange of cyber threat information for proactive, preventive, threat detection and protection against cyber attacks. CTI can improve organizational security (OrgSec) capabilities at multiple levels, including the strategic, tactical and operational levels.

To some extent, CTI and SR serve a similar purpose in that they can both be used to enhance CII's OrgSec capability. CTI identifies vulnerabilities and attacks from a technical, knowledge, and operational perspectives, whereas SR manages CII from management, system, and practitioner points of view. Fig. 1 illustrates how CTI and SR collaborate to safeguard the CII.

As shown in Fig. 1, the subject of protection is CII, which can be software, networks, services, physical or IT devices such as operating systems, IP addresses, firewalls, etc. SR data are the organization's security management requirement, with which each security attribute of CII must comply. Fig. 1 shows two security requirements: *authentication errors* and *separation of privileges*, which are linked to *CAPEC-460* and *CAPEC-574* in the CTI data respectively, CAPEC [34] (Common Attack Pattern Enumeration and Classification) is an attack pattern enumeration, and *CAPEC-460* is *HTTP Parameter Pollution* which indicates that an attacker overrides or adds HTTP GET/POST parameters by injecting query string delimiters.

The connection from *CAPEC-460* to the first requirement in SR means that when an *HTTP parameter pollution* event is detected in CII, the corresponding management item that *guarantees regular review of input validation errors* is not fully implemented, so we connect the two nodes via a *CAPEC_attribute_to_requirement* relation. Similarly, the edge from *CAPEC-574* to the last requirement in SR indicates that if *Privilege separation, least privilege and mutual restraining relationship* are fully implemented, the attack *Services Footprinting* (*CAPEC-574*) will fail. In addition, *CWE-284* is linked to *CAPEC-574* because *improper access control* may result in *services footprinting* attacks. *CVE-2019-12468* and *CVE-2008-5898* are entities in CVE [33] (Common Vulnerabilities & Exposures) that are also contextually relevant to *CWE-284*.

The preceding example demonstrates that by analyzing CII, CTI, and SR together, it is possible to develop correlations between CTI and SR data, thereby enhancing the cyber security capability of CII. We combine the CTI and SR data to construct the RCTI, a heterogeneous cybersecurity requirement knowledge graph.

Specifically, we collect 23 management documents of the People's Republic of China's national network security standards, which are then manually merged and de-duplicated to provide over 3,020 pieces of SR data. Meanwhile, we collect 13,800 CTI data points from cybersecurity intelligence websites [24,22,23], along with descriptions and hyperlinks, and manually connect links between SR data and CAPEC entities to build the RCTI knowledge graph. The architecture of the RCTI is shown in Fig. 2, and the construction method of the RCTI graph is discussed in Section 3.

However, the RCTI graph is far from complete due to the large number of edges that are difficult to add manually. Moreover, different from other cybersecurity domain KGs [44], the nodes in RCTI are not logically or temporally ordered, making the meta-path approach inappropriate. Consequently, we propose the EGNN, a novel GNN-based model that develops edge propagation algorithms capable of reasoning new knowledge on the RCTI.

The contributions of this paper are summarized as follows:

• To the best of our knowledge, this is the first work to combine SR and CTI data to construct a knowledge graph that can be used to identify management vulnerabilities and enhance the security capabilities of CII.

• We propose a novel GNN-based model called EGNN, which builds independent representations for known edges, and develops an algorithm for edge propagation, which enhances the representation capability of the graph.

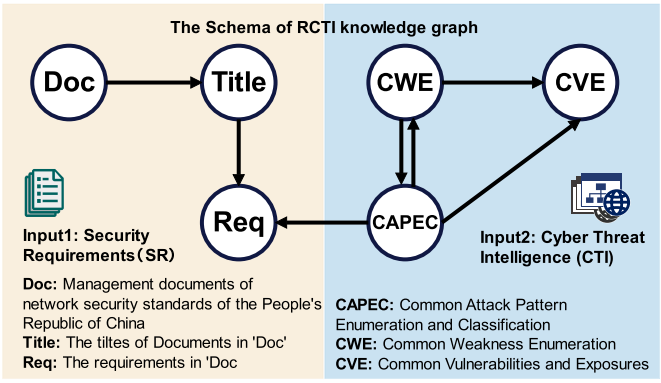

Fig. 2. The architecture of the RCTI knowledge graph. Two inputs are needed for the graph: (1) CTI information from the CVE, CWE and CAPEC websites that report cyber attacks and cyber network vulnerabilities. (2) SR data comprise 23 management documents of the People's Republic of China's national network security standards. CTI and SR data are manually connected by the links from the CAPEC entity to the Req entity.

• We evaluate the EGNN on three publicly available datasets to demonstrate its effectiveness and scalability. Additionally, we use the EGNN to predict links for the RCTI, and the results indicate that the EGNN is able to discover new knowledge between cybersecurity requirement data and cyber threat intelligence data. According to a manual validation, 97% of the cyber attack patterns can effectively identify one or more management requirements.

The rest of this paper is structured as follows. Section 2 reviews related works. In Section 3, we briefly introduce the RCTI knowledge graph's components. In Section 4, the recommended methodology is illustrated. The effectiveness of EGNN on the RCTI knowledge graph and three common datasets is evaluated in Section 5. Finally, the conclusion is presented in Section 6.

2. Related work

Several studies on knowledge inference in cybersecurity have been proposed recently. In this section, we focus on previous works on modeling cyber threat intelligence to construct knowledge graphs and then analyze inference models based on heterogeneous graphs.

2.1. Modeling of the CTI

As cyber attacks become more complex and organized, numerous well-written CTI exchange mechanisms, such as STIX [1], CVE [33], CWE [35] and CAPEC [34], have been created to share dynamic information about cyber threats in a prompt and comprehensive manner. CTI consists of unstructured or semistructured natural language with hyperlinks between elements, making automated modeling challenging. KG is commonly utilized in complex knowledge storing and reasoning due to its excellent knowledge representation and reasoning capabilities [44,16,45].

Designing an ontology is the foundation for constructing a knowledge graph, which provides a common understanding of the cybersecurity domain knowledge and unifies the most widely adopted cybersecurity standards. Some ontologies attempt to establish a comprehensive and generalized knowledge graph of cyber security knowledge [16]. In addition, there are several field-specific studies on cybersecurity knowledge reasoning, for instance, attack knowledge graphs created from machine-digestible evidence for hostile domain detection [31] or attack identification [31,45]. In this study, we construct the RCTI knowledge graph by combining unstructured SR data and semistructured CTI data to improve the security capabilities of CII.

2.2. Link prediction for heterogeneous knowledge graph

In recent decades, deep learning has demonstrated tremendous success in link prediction for KGs. Thus, we review these link prediction models and classify them into three categories: translation-based models, BERT-based models, and graph-based models.

2.2.1. Translation-based model

In KG, a triple is defined as $tr = (h, r, t)$, consisting of a head entity h, relation r and a tail entity t. The translation-based model defines r as a translation between the embeddings of h and t. The structural information is obtained by mapping triple elements into the same latent space such that entities with similar structures obtain similar representations. TransE [3], for instance, translates entities and relations to a low-dimensional real vector space, whereas RotateE [32] and ComplEx [37] map embeddings to complex vector space to model asymmetric relations. Similarly, TorusE [13] and QIQE-KGC [19] map to Lie group and quaternion space, respectively.

Despite the success of the translation-based model, it is challenging to predict links between entities that were never directly connected in the training data. As a result, translation-based models typically achieve high hit@K values but low MR (mean rank).

Table 1
Examples of the security requirement (SR) knowledge graph.

Doc (Document)	Title (Title)	Req (Security requirement)
Information Security Technology Information Security Risk Assessment Specification	Background ProcessUsage	Systematic monitoring of the controller's effectiveness and validation of compliance with the access policy model.
		The system should be monitored to detect deviations from the access rules and to record supervisory events to provide evidence in the event of a security issue.
	...	...
	Connectivity to the categorization system's security management center	When a suspected security incident is discovered, the organization's emergency center must be notified within the time frame given by the cloud service provider.
		In accordance with the incident response plan, track and report security incidents.

2.2.2. Bert-based model

Owing to the rapid advancement of natural language processing (NLP) technology, approaches that employ NLP technology for graph tasks achieves exceptional performance. KG-Bert [21] is the first to model the plausibility of triples with a pretrained language model that simply treats triples as textual sequences such as $(h||r||t)$ and sends them into BERT [12] for binary classification, where h is the token of the head entity and || is the concatenation operator. The approach is extremely successful but may lead to combinatorial explosion.

To overcome this issue, Pertrain-KGE [48] encodes entity and the relation embeddings. MLMLM [10] compares the mean likelihood of entity embeddings. StAR [39] divides triples into two asymmetric portions and uses a Siamese-style textual encoder to obtain contextualized representations. By reusing embeddings, these methods reduce the computational effort significantly.

2.2.3. Graph-based model

Graph-based models tend to capture the multihop information of knowledge graphs by means of graph neural networks (GNN). R-GCN [29] is the first model to introduce the GCN [18] framework into the link prediction task; it applies a convolution operation to the neighborhood with equal weights. By incorporating an attention mechanism, different weights can be assigned to neighboring nodes, leading to enhanced performance [25,43,42]. KBGAT [25] executes a two-layer transformation on the combination of entity and relation vectors. ReInceptionE [42] develops KBGAT [25] by taking (h, r) as a query and then employs a relation-aware attention mechanism to collect both local and global information. BiGAT [43] combines hierarchical multihop information with both in- and out-directional information using an attention mechanism. Graph-based models can also be combined with other algorithms [20,42,49] or multiple modalities [7,15,50] to achieve outstanding performance.

Research gaps remain in graph-based models. Current studies are based on node and relation features but ignore edge information. Therefore, we propose EGNN, which defines edge representations and develops an algorithm for edge propagation.

3. Requirement-CTI knowledge graph

In this section, we discuss the architecture of the Requirement-CTI(RCTI) knowledge graph, the data sources and the data preprocessing technique, then describe the data organization and annotation. Finally, we conduct a statistical analysis and define the task of link prediction on the RCTI graph.

3.1. Overview of the RCTI knowledge graph

The main techniques for increasing CII's OrgSec capacity include analyzing static indicators and assessing the effect of cyber-attack occurrence. Nevertheless, simply learning about attack tactics does not increase CII security. We must retrace the security events, identify the management vulnerabilities associated with the assault, and transform the weaknesses into operational procedures so that CII can avoid the next comparable attack and thereby increase the organization's security competence.

To accelerate the traceability of security incidents, we must break the barrier between CTI and SR knowledge. CTI is semistructured data generated using a bottom-up strategy, whereas SR knowledge is typically unstructured data constructed using a top-down approach as shown in Table 1, making association difficult. We create the *CAPEC_attribute_to_requirement* relation and manually insert 3,603 edges from CTI into the SR data to construct the RCTI graph, which implements the association of CTI and SR data. To the best of our knowledge, this is the first work to integrate CTI data with management knowledge for security research. Organizations may enhance their security capabilities both operationally and managerially when management vulnerabilities are discovered.

The architecture of the RCTI graph is shown in Fig. 2. Two inputs are needed for the graph: (1) CTI information from the CVE, CWE, and CAPEC websites that report cyberattacks and cyber network vulnerabilities; (2) SR data comprising 23 management documents of the People's Republic of China's national network security standards.

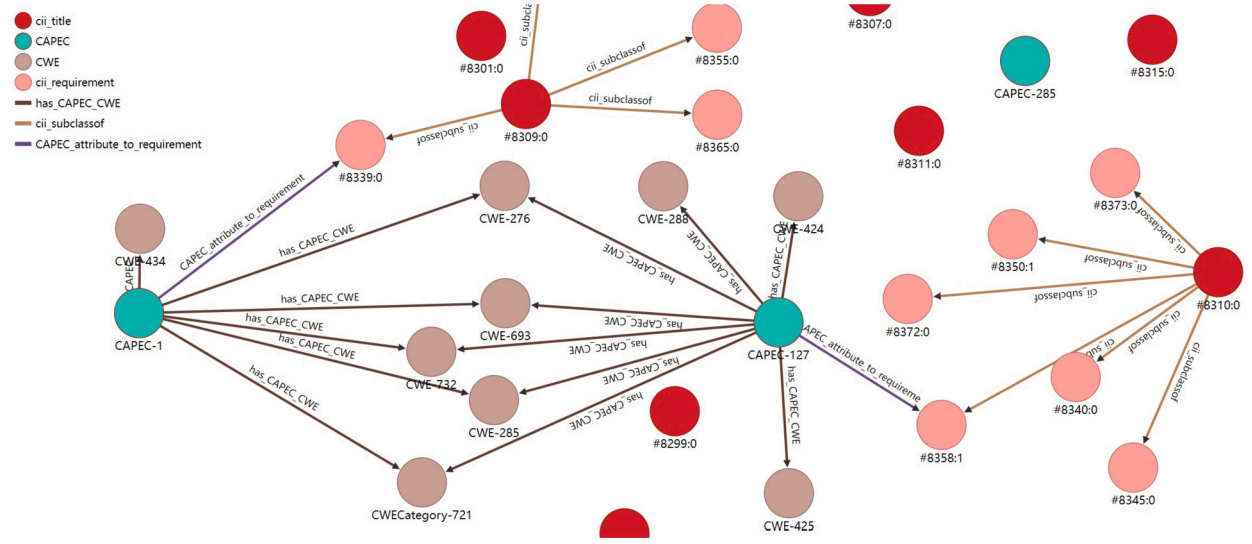

Fig. 3. Part of the proposed RCTI knowledge graph.

Our research team comprises a professional organization, two PhD candidates, six graduate students and four undergraduates, all of whom have more than three years of expertise in cyber attack and defense, or more than one year of experience in cyber security education and research.

We create a meta-graph consisting of six types of nodes and eight relations and then import 13,600 pieces of CTI data and 3,820 pieces of SR data to construct the RCTI. Each management requirement is analyzed and manually linked to attack patterns, and a PhD candidate verifies the accuracy of the annotated results. After filtering, 3,603 relations are established to combine CTI and SR data. However, the manually inserted relations are far from complete, thus we propose the EGNN model to complement the RCTI graph; the details of EGNN are presented in Section 4.

3.2. Data process

There are two data sources: security requirement data collected from 23 national cybersecurity management standards of the People's Republic of China and CTI data from open-source knowledge bases.

3.2.1. Security requirement data

Twenty-three management documents of the national network security standards of the People's Republic of China are collected and stored in the knowledge graph, and the structure of the SR subgraph is a hierarchical structure similar to a tree, as shown in Table 1. The hierarchy structure is derived naturally from the titles at all levels of the document, with the requirements of standards serving as leaf nodes. We construct three types of entities to represent different levels of nodes in the security requirement subgraph: Document, Title and Requirement. Then, we establish *document_has_title* and *title_has_requirement* relations depending on the document hierarchy. Repetitive titles and requirements are combined, superfluous nodes are eliminated, and 3,603 *CAPEC_attribute_to_requirement* edges are established from the CAPEC nodes to Req nodes via manual annotation.

3.2.2. The CTI data

While modeling the CTI graph, we do not target a specific attack pattern or type of vulnerability but instead consider all conceivable attack patterns and vulnerabilities. Furthermore, many attributes are not available due to the confidential characteristic of CII, such as IP address, domain name, and user information. We select CAPEC, CVE and CWE data and capture the hyperlinks between them to build the CTI subgraph. Five relationships are defined between the CTI entities: *CAPEC_to_CWE, CWE_to_CAPEC, CAPEC_to_CVE, CWE_to_CVE* and *CVE_to_CWE*.

3.3. The RCTI knowledge graph

The CTI subgraph is connected to the SR subgraph by the manually added *CAPEC_attribute_to_requirement* relation. The merged graph, RCTI, is a heterogeneous directed graph containing 16,820 nodes and 23,400 edges. Orientdb [28] is used to store and display the RCTI graph, as shown in Fig. 3.

The statistical information is presented in Table 2. $\#Entity$ and $\#Relation$ denote the numbers of entities and relations, $\#Train$, $\#Dev$ and $\#Test$ indicate the sizes of the training set, validation set and test set, respectively. We create the T/E value to represent $\#Train$ divided by $\#Entity$, which measures the sparseness of a graph. For example, the T/E value of FB15K-237 is 18.714, which

Table 2
Summary statistics for baseline datasets and the RCTI graph.

Dataset	#Entity	#Relation	#Train	#Dev	#Test	T/E
WN18RR	40,943	11	86,835	3,034	3,134	2.121
WN18	40,943	18	141,442	5,000	5,000	3.454
FB15K-237	14,541	237	272,115	17,535	20,466	18.714
RCTI	16,820	8	18,000	2,700	2,700	1.070

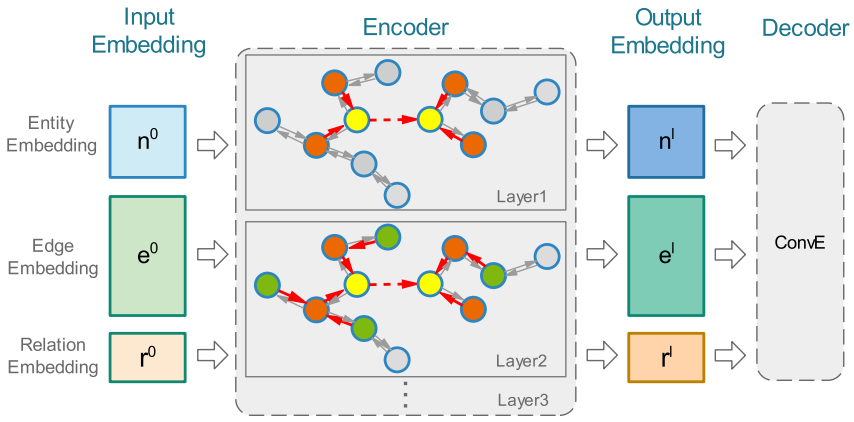

Fig. 4. The architecture of the EGNN. The EGNN is an encoder-decoder model. The encoder is a GNN-based model in which entity, edge, and relation embeddings are propagated and aggregated. In contrast to standard GNNs, the encoder builds edge representations e and proposes a novel edge propagation algorithm. ConvE [11] is used as a decoder that simulates the nonlinear interactions between representations.

means that, on average, a node in the training set is linked to 18 edges, whereas in the RCTI graph, the T/E ratio is 1.07, indicating that each node is linked to an average of one edge. Thus, the RCTI is 18 times sparser than FB15K-237.

When the relations between the requirement data and the CTI data are defined by the RCTI, the reasoning issue is transformed into a link prediction task on a heterogeneous directed graph.

4. The EGNN model

In this part, we first describe the RCTI link prediction problem and then build the EGNN model. Fig. 4 shows the architecture of the proposed model.

4.1. Problem formulation

A heterogeneous knowledge graph is denoted as $G = (N, R, E)$, where N, R and E represent the entities, relations, and edges in the training set, respectively, and $|R| > 1$. A single triple $(h, r, e, t) \in N \times R \times E \times N$ is a ground truth instance in the graph, h and t are the entities in N, r is the relation in R and e is the edge in E. The link prediction task on the graph is to predict the missing entity with the given $(h, r, e, ?)$ or $(?, r, e, t)$ using the known triples.

GCN [18] is a fundamental graph structure learning approach that learns graph information by aggregating the representations of neighbor nodes (and relations) iteratively as $H^{L+1} = f(H^L, R^L, W^e, W^r)$. Where W^e and W^r denote the specific parameters of the entities and relations in the model, respectively, and the training parameters are the embeddings at the 0-th layer for entities, as $H^0 = N^0$ in most existing graph convolution models.

To increase the representation capacity of the graph and enable edge propagation, we extend the network by adding inverted edges according to CompGCN [38],

$$\mathcal{E}' = \mathcal{E} \cup \left\{(h, r, t) \mid \left(t, r^{-1}, h\right) \in \mathcal{E}\right\} \tag{1}$$

Moreover, we develop a novel GNN-based edge propagation method:

$$H^{L+1} = f\left(H^L, R^L, E^L, W^h, W^r, W^e\right) \tag{2}$$

where E is the representation of all known edges, the number of which equals $\mathcal{E}'$. Hence, the link prediction task is transformed into a conditional query problem as: $Q(h, r, e) \longrightarrow t$, which considerably improves the efficiency of prediction.

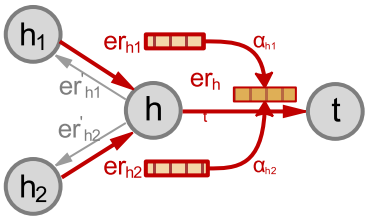

Fig. 5. Edge propagation. In synchronization with node propagation, edge representations are propagated from the head toward the tail entities with different attentions, and the edges sharing the head node get the same edge representation at the same layer.

4.2. EGNN model details

As previously stated, the GCN model gathers multihop information by aggregating neighbor nodes and relations. However, the magnitude of relations is far less than that of entities. Using the WN18RR dataset as an example, there are 40943 entities but only 11 types of relations, making the number of entities 3722 times greater than the number of relations.

The sparsity of relations restricts the representation capability of knowledge graphs; thus, we propose an edge representation approach to add a representation for each edge in the training set and construct an edge propagation model, as shown in Fig. 4. The EGNN is an encoder-decoder model. The representations of nodes, relations, and edges are propagated and aggregated by the encoder, and the nonlinear interactions between the three types of representations are then learned by the decoder. In the test phase, novel links are predicted using the decoder and the optimized representations of all layers.

4.2.1. Entity propagation

Conventional GNN models employ only the representation of the final layer for downstream tasks, but LightGCN [14] demonstrates that each layer of a GNN contains essential information, therefore, we concatenate the embedding of each layer to serve as the basis for prediction.

$$\begin{cases} N^{\ell} = \phi\left(\left\{n^{\ell} \mid \ell = 0 \cdots K\right\}\right), \\ E^{\ell} = \phi\left(\left\{er^{\ell} \mid \ell = 0 \cdots K\right\}\right), \\ R^{\ell} = \phi\left(\left\{r^{\ell} \mid \ell = 0 \cdots K\right\}\right) \end{cases} \tag{3}$$

where ϕ is the concatenation operation, $N^{\ell} \in \mathbb{R}^{|\mathcal{E}| \times d_f}$, $E^{\ell} \in \mathbb{R}^{|\mathcal{T}| \times d_f}$ and $R^{\ell} \in \mathbb{R}^{|\mathcal{R}| \times d_f}$ represent the final embeddings of entities, edges, and relations, respectively, d_f denotes the dimensionality of the final embeddings, $\mathcal{T}$ is the number of training sets, and K is a hyperparameter that specifies the number of GNN layers. In addition, n^{ℓ} is the embedding of the $\ell - th$ layer computed as follows:

$$n_i^{\ell+1} = \partial\left(\left(\mathrm{W}_n^{\ell} + \mathrm{I}\right) \cdot \sum_{t_j \in \mathcal{N}_i} \alpha_{ij}^{\ell}\left(h_i^{\ell} \times er_{ij}^{\ell}\right)\right) \tag{4}$$

where Tanh is used as the activation function ∂, $\times$ is an aggregation operator that updates the information by element-wise multiplication of the head and edge embeddings of every neighbor triple. h_i and er_{ij} represent the embeddings of head entity i and edge ij, respectively. The identity matrix $I \in \mathbb{R}^{(d_l \times d_l)}$ added to the transformation weight $W_n \in \mathbb{R}^{(d_l \times d_l)}$ in propagation is proven to be simple and efficient [9] since it can optimize W_n for a minor norm to achieve the global optimal solution, and d_l is the dimensionality of the entity at layer l. The attention value α_{ij} is computed as:

$$\alpha_{ij} = \frac{\exp\left(\left(h_i \circ er_{ij}\right)^T t_j\right)}{\sum_{t_m \in \mathcal{N}_i} \exp\left(\left(h_i \circ er_{im}\right)^T t_m\right)} \tag{5}$$

where $\circ$ is the concatenation operation in Eq. (10).

4.2.2. Edge propagation

Consistent with the entity representations, the EGNN creates an independent representation for each known edge e_{ij}, which represents the edge between nodes i and j. Meanwhile, we retain the relation representation r_{ij} from the traditional GNN algorithm and use er_{ij} to combine edge and relation information as follows:

$$er_{ij} = 0.5\left(e_{ij} + r_{ij}\right) \tag{6}$$

In edge propagation, edge information is passed from one layer to the next in a head-to-tail direction, as shown in Fig. 5, where edges with the same head have the same representation at the same layer. This approach enables edges to acquire information about edges farther away and eventually obtain independent edge representations.

$$er_{ij}^{\ell+1} = \partial\left(\left(\mathrm{W}_{er}^{\ell} + \mathrm{I}\right) \cdot \sum_{h_j \in \mathcal{N}_i} \alpha_{ij}^{\ell} er_{ij}^{\ell}\right) \tag{7}$$

In the prediction stage, given that the edges in the test set do not currently have representations, we use a single edge propagation step to create the target edge representations in the test set:

$$er_{ij}^{\text{predict}} = \partial\left(\left(\mathrm{W}_{er}^{\ell} + \mathrm{I}\right) \cdot \sum_{h_m \in \mathcal{N}_i} \alpha_{mi}^{\ell} er_{mi}^{\ell}\right) \tag{8}$$

where er_{mi} denotes the edge representation with i as the tail entity. We use the DGL [41] framework to implement the above propagation algorithm, which can effectively reduce the programming complexity and increase the training efficiency.

4.2.3. Relation propagation

Despite the small number of relations, this approach provides an effective classification of relation categories, so we use simple MLPs for the dimensional transformation of relations:

$$r^{\ell+1} = \left(W_r^{\ell} + I\right) r^{\ell} \tag{9}$$

4.2.4. Decoder

ConvE [11], which uses a 2D convolution neural network to match the (h, r) with t, is selected as the decoder. We extend the query Q as follows:

$$\mathrm{Q} = \mathrm{ConvE}(\mathrm{h} \| (r + er)) \tag{10}$$

where || indicates the concatenation operation and $Q \in \mathbb{R}^{|\mathcal{T}| \times d_f}$ is the query embedding. InfoNCE [40] loss with additive margin is used during training:

$$\mathcal{L} = -\log \frac{e^{(f(Q,t)-\gamma)/\tau}}{e^{(f(Q,t)-\gamma)/\tau} + \sum_{i=1}^{|N|} e^{f(Q,t')/\tau}} \tag{11}$$

The additive margin $\gamma > 0$ encourages the model to increase the score of the positive triple (h, r, e, t), τ is the temperature that can be used to adjust the relative importance of negative triples, and $\mathrm{N} = |\mathcal{E}|$ is the number of candidate entities. $f(Q, t)$ is the score function defined as follows:

$$f(Q,t) = \frac{1}{1 + e^{-Q^T \cdot t}} \tag{12}$$

where $f(Q,t) \in [-1, 1]$. Algorithm 1 shows the whole process of the EGNN. The complexity of EGNN is $O(m + n)$, where m and n represent the numbers of edges and nodes, respectively.

Algorithm 1 Edge propagation for knowledge graph (EGNN).

Input: Original graph G, initial entity embedding N^0, initial edge embedding E^0, initial relation embedding R^0, GNN layer L
Output: pred

for Each layer l of GNN **do**
  Recalculation $E^0 = 0.5(E^0 + R^0)$ using Eq. (6)
  Calculate attention value for triples to get α_{ij} using Eq. (5)
  Propagate and combine node embedding N^l and edge embedding E^l with attention using Eq. (4) and Eq. (7)
  Calculate attention value for triples to get α^{l+1} using Eq. (5)
end for
Add the edge vector with α^{l+1} to the tail node to get E^{l+1} using Eq. (8)
Relation Propagation using Eq. (9)
Using ConvE to calculate the score between each $(h^{l+1}, e^{l+1}, r^{l+1})$ and t^{l+1}, using Eq. (10)
Calculate the loss using Eq. (11)
repeat Iterative training to obtain optimal predictions
until Early stop until the MMR value has not increased during 50 epochs

5. Experiments

In this section, the effectiveness of the EGNN is evaluated by comparing it to various benchmark methods. We then apply the EGNN model to the RCTI knowledge graph to infer new connections between management knowledge and CTI knowledge. Finally, we estimate the accuracy of these new links.

5.1. Experimental setup

For the link prediction task, we conduct experiments on the RCTI graph and three popular public datasets FB15K-237 [36], WN18 and WN18RR [11]. The statistics of these datasets are provided in Table 2. $\#T/E$ is the ratio of the number of training sets to the number of entities, indicating that although all knowledge graphs are sparse, WN18RR, WN18 and RCTI are sparser than FB15K-237.

The EGNN is compared to various state-of-the-art approaches, including TransE [3], RotatE [32] and Rot-Pro [30] as representations of translation-based models, DistMult [46] for bilinear models, MuRP [5], HyperGEL [47] and ConE [4] for hyperbolic models. R-GCN [29] and CompGCN [38] as graph neural network models, ConvE [11] and ConvKB [26] as convolution models, and KB-GAT [25], KBGAN [8], and A2N [6] as attention-based model. To ensure a fair comparison, we employ the authors' available source code or our own implementations and evaluate each method in a filtered setting.

In FB15k-237, the EGNN achieves the optimal performance at: $d = 200$, learning rate = 0.0035, dropout = 0.7, batch size = 100 and $l = 2$ for hidden layers, and $c = 300$ for the channels in the decoder. For WN18RR, WN18 and RCTI, the optimal parameters are: $d = 300$, dropout = 0.6, batch size = 1,000, $l = 1$ and $c = 500$. Temperature τ is fixed to 0.05, margin γ is set to 0.5 and the kernel size is 8 for all datasets. Models are trained on a V100 GPU. The early stopping approach (with the hyperparameter "patience" set to 50 iterations) is used to select the final model with a maximum number of epochs of 1,000, and the checkpoint with the highest validation performance is chosen to report the test performance.

The following evaluation measures are frequently employed in link prediction tasks:

1. $Hits@N$ represents the fraction of correct entities in the top N ($N = 1$, 3 or 10):

$$\text{Hit @ N} = \frac{|\{q \in Q : q \leq N\}|}{|Q|}, \tag{13}$$

where Q is the set of ranks derived from test predictions. $Hits@N$ is always between 0 and 1, and the greater the value is, the better the result.

2. The mean rank (MR) represents the mean rank of correct entities:

$$\text{MR} = \frac{1}{|Q|} \sum_{q \in Q} q. \tag{14}$$

This value is always between 1 and $|\mathcal{E}|$, and the lower the value is, the better the result.

3. The mean reciprocal rank (MRR) is the mean reciprocal of the rank:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{q}, \tag{15}$$

5.2. EGNN performance evaluation

Table 3 and Table 4 compare the performance of the EGNN and the baseline methods. In general, convolutional models [11,26] outperform TransE [3] and DistMult [46] because the convolutional operation can produce better synthesis by combining multilayers and multientity features.

Due to the consideration of the weights of various neighbors, the attention model [8,25,6] can anticipate candidate nodes, thereby reducing the MR value. KBGAT [25] and CompGCN [38] introduce the representation of two-hop relations, symmetric relations and self-reflexive relations, improve the expression ability of the relations, and enhance the model performance, but they significantly increase the computational complexity.

Based on the GNN architecture, the EGNN introduces an attention and edge propagation mechanism that permits simultaneous transmission of edges, relations, and entities. As shown in Table 3, the training set for FB15k-237 has 272,115 edges, which considerably improves the representation capability and enables EGNN to obtain the best results for FB15k-237 measurements. The training set of WN18RR, in contrast, contains only 86,835 edges, which restricts the representational capacity. Nonetheless, the performance on WN18RR is surpassed only by the hyperbolic models [5,47,4] on WN18RR. On WN18, the EGNN also achieves the best performance on four of five metrics, showing the importance of the edge representation.

However, expanding the edge representation significantly increases memory and computational resource requirements. Our research is conducted on a single V100, and we believe that additional resources could further enhance the performance.

In addition, we use these baseline models to predict the RCTI graph, using the authors' available source code and employing the same settings. Due to the coarse and sparse topology of the RCTI knowledge graph, the results indicate that all approaches are less effective. In spite of this, the EGNN achieves the best performance regarding the majority of measures, as shown in Table 4.

5.3. Ablation study

Ablation studies are conducted to demonstrate the validity of the EGNN's individual components. On WN18RR, we replace the ConvE component in the EGNN with the score function of the TransE and HolE [27]. Since the edge representations in the model are independent, we modify the objective function of TransE to be $h + e + r$ and that of HolE [27] to be $h \circ (e + r)$, where $\circ$ indicates the Hadamard product. The findings are presented in Table 5, which indicates that the EGNN is still capable of making accurate predictions, even when the ConvE module is removed, and the addition of the ConvE module does indeed enhance the EGNN's performance.

Table 3
Link prediction performance of the EGNN and baseline models on the FB15K-237 and WN18RR datasets. [1] means the results reported by [39], [2] means the results re-evaluated by [20], and others are taken directly from the previous papers. – indicates missing values.

Dataset	FB15K-237					WN18RR				
	hit@10	hit@3	hit@1	MR	MRR	hit@10	hit@3	hit@1	MR	MRR
Translation-based Model										
TransE [3][1]	0.441	0.376	0.198	323	0.279	0.532	0.441	0.043	2300	0.243
RotatE [32]	0.480	0.328	0.205	-	0.297	-	-	-	-	-
DistMult [46]	0.446	0.301	0.199	512	0.281	0.504	0.470	0.412	7000	0.444
ConvKB [26]	0.421	-	-	-	0.243	0.525	-	-	-	0.248
ConvE [11]	0.501	0.356	0.237	244	0.325	0.520	0.440	0.400	4187	0.430
MuRP [5]	0.518	0.367	0.243	-	0.335	0.566	0.495	0.440	-	0.481
HyperGEL [47]	0.525	0.378	0.253	-	0.345	0.577	0.501	0.445	-	0.488
Rot-Pro [30]	0.540	0.383	0.246	201	0.344	0.577	0.482	0.397	2815	0.457
ConE [4]	0.540	0.381	0.247	201	0.345	0.579	0.515	0.453	-	0.494
GNN-based Model										
R-GCN [29]	0.417	-	0.151	-	0.248	-	-	-	-	-
KBGAT [25][2]	0.331	-	-	270	0.157	0.554	-	-	**1921**	0.412
KBGAN [8]	0.481	-	-	-	0.215	0.458	-	-	-	0.278
A2N [6]	0.486	0.348	0.232	-	0.317	0.510	0.460	0.420	-	0.450
EGNN	0.532	**0.383**	**0.258**	**168**	**0.350**	0.548	0.489	0.435	2828	0.474

Table 4
Link prediction performance of the EGNN and baseline models on the RCTI and WN18.

Dataset	RCTI					WN18				
	hit@10	hit@3	hit@1	MR	MRR	hit@10	hit@3	hit@1	MR	MRR
TransE [3]	0.181	0.072	0.011	4873	0.002	0.754	-	-	263	-
DistMult [46]	0.228	0.181	0.121	5426	0.160	-	-	-	-	0.911
ConvE [11]	0.304	0.209	0.137	3620	0.192	0.956	0.946	0.935	374	0.943
R-GCN [29]	0.111	0.088	0.006	7204	0.159	-	-	-	-	-
KBGAT [25]	0.264	0.217	**0.174**	**1525**	0.206	-	0.914	0.770	**211**	0.845
CompGCN [38]	0.319	0.227	0.156	3350	0.209	0.958	0.949	0.931	284	0.942
EGNN	**0.324**	**0.230**	0.160	3822	**0.214**	**0.958**	**0.953**	**0.944**	250	**0.950**

Table 5
Ablation study of the EGNN on WN18RR, where rp indicates replacing the ConvE operation with the corresponding score function in the EGNN.

Methods	hit@10	hit@3	hit@1	MR	MRR
EGNN	**0.548**	**0.489**	**0.435**	2828	**0.474**
rp $h+e+r$	0.518	0.450	0.356	3661	0.415
rp $h \circ (e+r)$	0.544	0.477	0.418	**2750**	0.461

Furthermore, we introduce CompGCN [38] to conduct a comparative analysis and evaluate the performance at 64, 100 and 200 dimensions, as depicted in Fig. 6. The results indicate that the EGNN performs more effectively on both datasets, while CompGCN is more consistent. The EGNN achieves the best results at dimensions 100 and 200, whereas CompGCN is more accurate at dimension 64, proving the scalability and robustness of EGNN.

5.4. Reasoning based on cybersecurity intelligence and management knowledge

To quantify the newly acquired information between cybersecurity intelligence and management knowledge, we randomly choose 100 CAPEC nodes from the RCTI graph and predict 5 candidates for each node using the *CAPEC_attribute_to_requirement* relation. After filtering, 439 new knowledge instances remain (the other 61 are duplicates with the training set or validation set). Manual evaluation of these 439 new knowledge instances by the research team indicates that 97 of these 100 CAPEC node candidate entities are properly connected to more than one management requirement, which means that the EGNN is able to find more than one management vulnerability for 97% of the common cyber attacks.

Furthermore, we analyze the composition of these 439 new links, and the statistical data are displayed in Fig. 7. Of the 439 new links, 55 are connected to *Document* entities, all of which are considered to be True, as all CAPEC entities should be connected to most management documents. There are 148 *Title* links out of 313 and 41 *Requirement* links out of 71 that are considered True. The average precision is 55%.

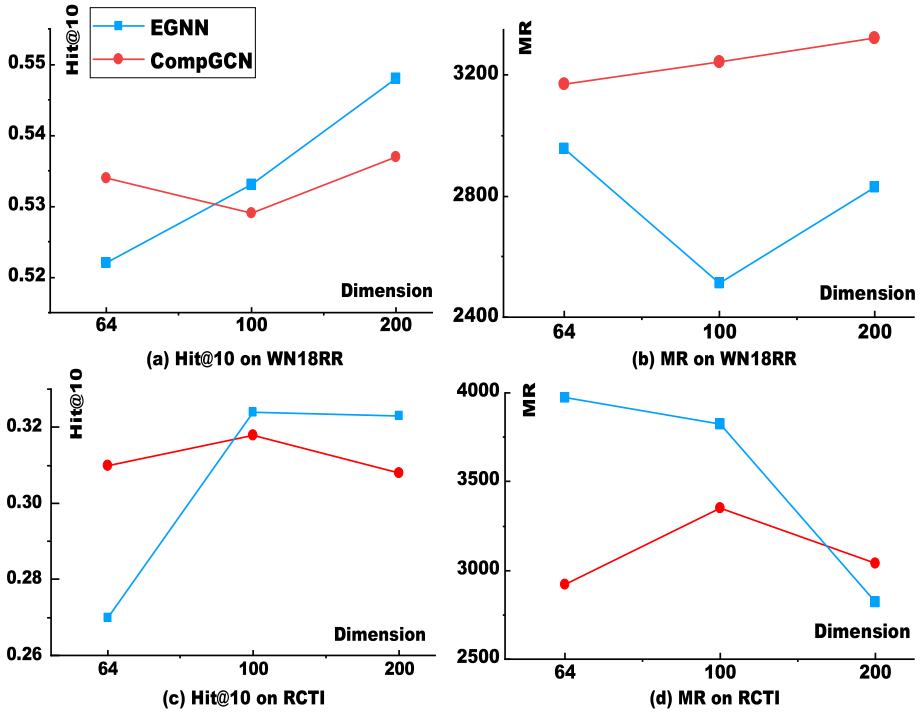

Fig. 6. The performance of the EGNN and CompGCN on the WN18RR and RCTI in various dimensions. The EGNN achieves the best results at dimensions 100 and 200, whereas the CompGCN is more accurate at dimension 64.

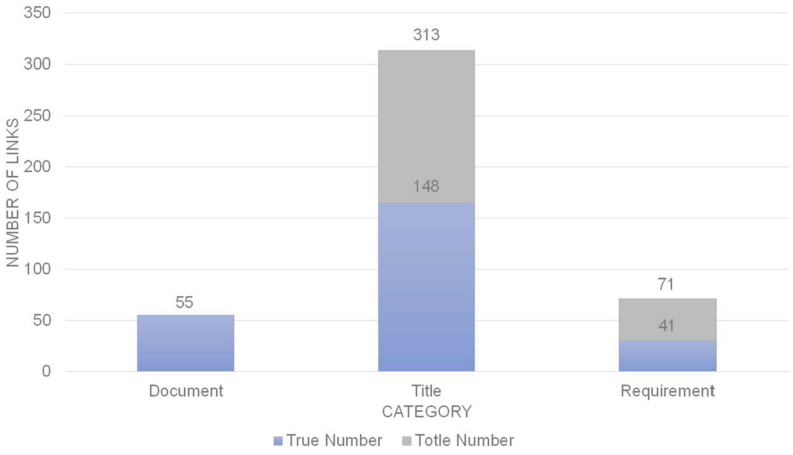

Fig. 7. The distribution of newly predicted links. SR knowledge comprises three categories of entities: Document, Title, and Requirement. The graph illustrates the ratio of new links to those regarded as correct.

Table 6 displays partial candidates and manual analysis findings, where *182* is the CAPEC id for the *Flash Injection* attack technique. In this attack, an attacker deceives a victim into executing malicious flash content that performs instructions or makes flash calls that the attacker specifies. Cross-site flashing is an example of this type of attack in which an attacker-controlled argument to a reference call loads material supplied by the attacker. To perform this attack, the target must be able to run Flash apps, and in some situations, the victim must click on a link provided by the attacker.

EGNN provides four *CAPEC-182* related items, where *#1162:53* is the ID corresponding to the management requirement in the RCTI graph, which *signifies periodic examination for input validation issues*. We believe that the *CAPEC-182* attack is detectable via the *#1162:53* operation. *Access control (AC)* is the content of *Title-887*. *#1192:63* is a management solution to the *buffer overflow* problem that is unrelated to *CAPEC-182*, as *False indicates*. Last, *#1165:53* indicates that the input to the system must adhere to the organization-predefined format and content. After performing a thorough investigation, we have assigned the *True* label to three of the four candidates since we believe they are closely related to *CAPEC-182*. The remaining candidate has been designated False.

Table 6
Parts of new knowledge between cybersecurity intelligence and management requirements.

CAPEC		Candidate Requirements		Value
ID	content	ID	content	
182	Flash Injection	#1162:53	Ensure input validation errors are audited periodically.	True
		Title_887	CS Security Management Basic Controls/Access Control (AC)	True
		#1192:63	Buffer overflow: Industrial control system software may suffer from buffer overflow.	False
		#1165:53	Limit system input according to the organization's predefined content and format.	True
275	DNS rebinding	#1215:4	Network topology monitoring must achieve the standardization of audit data, and its content must include the date, time, subject identification, object identification, type, result, IP address, and port.	True
		#294:0	Essential security management criteria for industrial control systems.	True
53	Postfix, Null Terminate and Backslash	#1164:53	Responses to invalid inputs should not alter the operation order.	True
		#1162:53	Ensure input validation errors are audited periodically.	True
		#1224:53	The correctness of the input data should be verified.	True
		title_1501	Approve the information security strategy, delegate security roles, and coordinate security implementation across the enterprise.	True
3	Using Leading Ghost Character Sequences to Bypass InputFilters	#1205:0	The software fault tolerance of the security management center's console should include a mechanism for validating data.	True
		#1165:53	Limit system input according to the organization's predefined content and format.	True
		#1186:37	Error confirmation protocol execution.	True
		Title_412	Risk element identification	True
487	ICMP Flood	Title_1923	System development and supply chain security/unsupported system components	False
		#1211:9	GB/T209842007 classifies threats into eleven categories: hardware and software failure, physical environment impact, inactivity or operational error, improper management, malicious code, unauthorized or abused, network assaults, physical attacks, leakage, manipulation, and repudiation.	True
55	Rainbow Table Password Cracking	#1166:3	Offer corresponding correlation analysis rules for common attacks and access violations, including host scanning, port scanning, DDoS attacks, worms, password guessing, springboard attacks, etc.	True
		#1166:64	Inadequate password protection: Without suitable password management mechanisms, unauthorized users may obtain unlawful access to confidential information.	True
		Title_692	The security administrator should handle the system's keys consistently, including key production, distribution, update, storage, backup, and destruction.	True
		#1174:64	The evaluator examines on-site if the password is saved in plaintext in the local system or portable device, whether there have been occurrences of password leakage in the past, and evaluates the password's dependability using brute-force cracking techniques in a simulated scenario.	True
		#1189:61	Unauthorized attacks (misuse, sniffer, spoofing, and social activity) to access user credentials stored in the industrial control system server components.	True
		Title_67	Treatment of alterations	False
		Title_345	Configuration Management (CM)/Configuration Minimum Functionalization	False
140	Bypassing of Intermediate Forms in Multiple-Form Sets	Title_61	Communications and Operations Management/ Media Disposal and Security/Information Disposal Protocols	True
		#1212:35	Prohibit the insecure storing of individual passwords.	False
		#1189:61	Unauthorized attacks (misuse, sniffer, spoofing, and social activity) to access user credentials stored in the industrial control system server components.	False
		#1205:14	A cyber attack incident is an information security event that exploits the configuration defects, and program flaws in an information system or uses a violent attack on an information system via network or other technical means to cause an abnormality in the information system or a potential risk to the current operation of the system.	True

Notably, the links from *CAPEC* to *Title* and *Document* are not defined in the meta-graph, as *Title* and *Document* are highly abstract phrases for which it is difficult to manually define the connections. However, with the assistance of the EGNN, we can infer the connection probability between any meta-nodes.

CAPEC-275 is the corresponding attack pattern for *DNS rebinding*. An adversary provides material whose IP address is resolved by a DNS server that the adversary controls in this attack. Following first contact by a web browser (or comparable client), the

adversary modifies the IP address to which its name resolves to an address within the target company that is not publicly accessible. This permits the web browser to inspect this internal address on behalf of an attacker.

#1215:4 and *#294:0* specify that audit data should be standardized to include date, time, subject identity, object identification, type, result, IP address, and port information. These are all regarded as successful *CAPEC-275* attack defense strategies.

This study illustrates that EGNN can accurately predict new knowledge on the RCTI and identify management vulnerabilities to enhance the security capability of CII.

6. Conclusion

This paper begins by introducing management knowledge and cyber threat intelligence as the foundation for enhancing the cybersecurity capabilities of critical information infrastructures. Then we develop the RCTI knowledge graph to integrate cyber threat intelligence data with cybersecurity management requirement data to identify management vulnerabilities in the case of cybersecurity incidents.

The EGNN is proposed for the link prediction task on the RCTI graph, which provides representations for existing edges for the first time and effectively enhances the representation capacity of the graph. To demonstrate the efficacy of the model, we compared the EGNN to several baselines, and the EGNN achieves the best performance on majority metrics.

Finally, we employ the EGNN to predict links on the RCTI graph and evaluate the accuracy of the newly acquired information. The findings indicate that newly discovered information contributes to the improvement of CII's security capability by identifying management flaws.

Future directions include the incorporation of natural language processing techniques into link prediction algorithms and the implementation of the joint analysis of semantic and graph structures to produce more robust implicit representations. Multimodal data can also be considered for the expansion of the security knowledge graph.

CRediT authorship contribution statement

Yang Zhang: Conceptualization, Methodology, Software, Writing – original draft. **Jiarui Chen:** Data curation. **Zhe Cheng:** Data curation, Visualization. **Xiong Shen:** Data curation, Investigation. **Jiancheng Qin:** Project administration. **Yingzheng Han:** Supervision, Writing – review & editing. **Yiqin Lu:** Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the Key Area Research and Development Program of Guangdong Province, China [grant numbers 2019B010137001, 2020B0101120002, 2018B010113001]; and the National Key Research and Development Program of China [grant number 2020YFB1805300].

References

[1] Introduction to stix. Website, https://oasis-open.github.io/cti-documentation/stix/intro.html, 2022.
[2] A. Alshamrani, S. Myneni, A. Chowdhary, D. Huang, A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities, IEEE Commun. Surv. Tutor. 21 (2019) 1851–1877, https://doi.org/10.1109/comst.2019.2891891.
[3] Antoine Bordes, A.G.D. Nicolas Usunier, Translating embeddings for modeling multi-relational data, in: NIPS2013, 2013.
[4] Y. Bai, Z. Ying, H. Ren, J. Leskovec, Modeling heterogeneous hierarchies with relation-specific hyperbolic cones, Adv. Neural Inf. Process. Syst. 34 (2021) 12316–12327.
[5] I. Balazevic, C. Allen, T. Hospedales, Multi-relational Poincaré graph embeddings, Adv. Neural Inf. Process. Syst. 32 (2019).
[6] T. Bansal, D.C. Juan, S. Ravi, A. McCallum, A2n: attending to neighbors for knowledge graph inference, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4387–4392.
[7] P. Bloem, X. Wilcke, L. van Berkel, V. de Boer, Kgbench: a collection of knowledge graph datasets for evaluating relational and multimodal machine learning, in: The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18, Springer, 2021, pp. 614–630.
[8] L. Cai, W.Y. Wang, Kbgan: adversarial learning for knowledge graph embeddings, arXiv preprint, arXiv:1711.04071, 2017.
[9] M. Chen, Z. Wei, Z. Huang, B. Ding, Y. Li, Simple and deep graph convolutional networks, in: Proceedings of the 37th International Conference on Machine Learning, 2020, JMLR.org.
[10] L. Clouâtre, P. Trempe, A. Zouaq, A.P.S. Chandar, Mlmlm: link prediction with mean likelihood masked language model, in: FINDINGS, 2020.
[11] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Thirty-Second Aaai Conference on Artificial Intelligence/Thirtieth Innovative Applications of Artificial Intelligence Conference/Eighth Aaai Symposium on Educational Advances in Artificial Intelligence, 2018, pp. 1811–1818.

[12] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR, arXiv:1810.04805, 2018.
[13] T. Ebisu, R. Ichise, Toruse: knowledge graph embedding on a Lie group, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI Press, 2018, pp. 1819–1826.
[14] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: simplifying and powering graph convolution network for recommendation, in: SIGIR '20: The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020.
[15] N. Huang, Y.R. Deshpande, Y. Liu, H. Alberts, K. Cho, C. Vania, I. Calixto, Endowing language models with multimodal knowledge graph representations, arXiv preprint, arXiv:2206.13163, 2022.
[16] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, J. Goodall, Developing an ontology for cyber security knowledge graphs, in: The 10th Annual Cyber and Information Security Research Conference, 2015.
[17] D. Javaheri, S. Gorgin, J.A. Lee, M. Masdari, Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: classification, overview, and future perspectives, Inf. Sci. 626 (2023) 315–338, https://doi.org/10.1016/j.ins.2023.01.067.
[18] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR 2017, 2017.
[19] L. Li, X. Zhang, Z. Jin, C. Gao, R. Zhu, Y. Liang, Y. Ma, Knowledge graph completion method based on quantum embedding and quaternion interaction enhancement, Inf. Sci. 648 (2023) 119548.
[20] R. Li, Y. Cao, Q. Zhu, G. Bi, F. Fang, Y. Liu, Q. Li, How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view, in: AAAI'22, 2021.
[21] L. Yao, C. Mao, Y. Luo, KG-BERT: BERT for knowledge graph completion, in: arXiv preprint, arXiv:1909.03193, 2019.
[22] MITRE, a. Common vulnerabilities and exposures (cve), https://www.cve.org/, February 23, 2023.
[23] MITRE, b. Common weakness enumeration (cwe), https://cwe.mitre.org/, February 23, 2023.
[24] MITRE, T.M.C. Common attack pattern enumeration and classification (capec), https://capec.mitre.org/, February 23, 2023.
[25] D. Nathani, J. Chauhan, C. Sharma, M. Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in: ACL 2019, 2019.
[26] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen, D. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 327–333, https://aclanthology.org/N18-2053.
[27] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016.
[28] D. Ritter, L. Dell'Aquila, A. Lomakin, E. Tagliaferri, Orientdb: a nosql, open source MMDMS, in: Proceedings of the British International Conference on Databases 2021, London, United Kingdom, March 28, 2022, 2021, pp. 10–19, CEUR-WS.org.
[29] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, in: Proceedings, vol. 15, Springer International Publishing, 2018, pp. 593–607.
[30] T. Song, J. Luo, L. Huang, Rot-pro: modeling transitivity by projection in knowledge graph embedding, Adv. Neural Inf. Process. Syst. 34 (2021) 24695–24706.
[31] X.Q. Sun, Z.L. Wang, J.H. Yang, X.R. Liu, Deepdom: malicious domain detection with scalable and heterogeneous graph convolutional networks, Comput. Secur. 99 (2020) 102057, https://doi.org/10.1016/j.cose.2020.102067, <Go to ISI>://WOS:000591706000005.
[32] Z. Sun, Z.H. Deng, J.Y. Nie, J. Tang, Rotate: knowledge graph embedding by relational rotation in complex space, in: 2019 ICLR, 2019.
[33] C. Team, Cve (version 20061101) and candidates as of 20220629, Available from MITRE, https://cve.mitre.org/data/downloads/allitems.html, 2020.
[34] C. Team, Schema documentation - schema version 3.5, Website, https://capec.mitre.org/documents/schema/index.html, 2022.
[35] C. Team, Schema documentation - schema version 6.8, Available from MITRE, https://cwe.mitre.org/data/index.html, 2022.
[36] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, 2015, pp. 57–66.
[37] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning, vol. 48, 2016, pp. 2071–2080, JMLR.org.
[38] S. Vashishth, S. Sanyal, V. Nitin, P.P. Talukdar, Composition-based multi-relational graph convolutional networks, arXiv:1911.03082 [abs], 2020.
[39] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference 2021, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1737–1748.
[40] L. Wang, W. Zhao, Z. Wei, J. Liu, SimKGC: simple contrastive knowledge graph completion with pre-trained language models, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 4281–4294, https://aclanthology.org/2022.acl-long.295.
[41] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, Deep graph library: a graph-centric, highly-performant package for graph neural networks, https://doi.org/10.48550/arXiv.1909.01315, 2019.
[42] Z. Xie, G. Zhou, J. Liu, X. Huang, Reinceptione: relation-aware inception network with joint local-global structural information for knowledge graph embedding, in: ACL, 2020.
[43] Z. Xie, R. Zhu, J. Liu, G. Zhou, J.X. Huang, Hierarchical neighbor propagation with bidirectional graph attention network for relation prediction, IEEE/ACM Trans. Audio Speech Lang. Process. 29 (2021) 1762–1773.
[44] Y. Xu, Y. Fang, C. Huang, Z. Liu, Hghan: hacker group identification based on heterogeneous graph attention network, Inf. Sci. 612 (2022) 848–863.
[45] L. Yan, C. Luo, R. Shao, Discrete log anomaly detection: a novel time-aware graph-based link prediction approach, Inf. Sci. 647 (2023) 119576, https://doi.org/10.1016/j.ins.2023.119576.
[46] B. Yang, S.W.t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proceedings of the International Conference on Learning Representations (ICLR) 2015, 2015, https://www.microsoft.com/en-us/research/publication/embedding-entities-and-relations-for-learning-and-inference-in-knowledge-bases/.
[47] A. Zeb, A.U. Haq, J. Chen, Z. Lei, D. Zhang, Learning hyperbolic attention-based embeddings for link prediction in knowledge graphs, Knowl.-Based Syst. 229 (2021) 107369.
[48] Z. Zhang, X. Liu, Y. Zhang, Q. Su, X. Sun, B. He, Pretrain-kge: learning knowledge representation from pretrained language models, in: FINDINGS, 2020.
[49] Z. Zhou, J. Shi, S. Zhang, Z. Huang, Q. Li, Effective stabilized self-training on few-labeled graph data, Inf. Sci. 631 (2023) 369–384.
[50] J. Zhu, C. Huang, P. De Meo, Dfmke: a dual fusion multi-modal knowledge graph embedding framework for entity alignment, Inf. Fusion 90 (2023) 111–119.