2024 IEEE 9th International Conference on Data Science in Cyberspace (DSC) | 979-8-3503-9136-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/DSC63484.2024.00043

Spherical Hierarchical Knowledge Graph Embeddings for Cybersecurity Knowledge Graph Completion

Yao Lin[1], Dong Zhu[1], Chenhui Zhang[1], Zhiqiang Zhang[2] Le Wang[1,*]
[1] Guangzhou University, Guangzhou, China
[2] Harbin Institute of Technology (Shenzhen), Shenzhen, China
{linyao, zhudong, chenhui_zhang}@e.gzhu.edu.cn, zhangzhiqiang@stu.hit.edu.cn, wangle@gzhu.edu.cn

***Abstract*—In response to the analysis demands for cybersecurity knowledge, researchers have established various abstract-level knowledge repositories that document attack behaviors, attack patterns, and vulnerability weaknesses. However, the existing knowledge repositories lack comprehensive associated knowledge. Some correct knowledge is absent from these repositories. The storage format of the current repositories necessitates a substantial investment of time and human resources to complete the missing correct knowledge. To tackle this challenge, this paper constructs a network security knowledge graph based on the four major knowledge repositories and introduces a spherical hierarchical knowledge graph embedding model. This model captures the hierarchy, symmetry, inversion, and composition patterns within the knowledge graph, facilitating the automatic completion of missing knowledge. We conducted experiments to assess the performance of our model, and the results robustly demonstrate the effectiveness of our proposed model in completing knowledge graphs.**

***Index Terms*—cybersecurity knowledge, knowledge graph embedding**

I. Introduction

Concerning attack behaviors, attack pattern classifications, and vulnerabilities within the realm of network security, the organization MITRE has initiated and manages various knowledge repositories at different levels of abstraction [1]. These knowledge repositories vary in their expressive capabilities and conceptual granularity, with higher abstract-level offering greater conceptual granularity and a more macroscopic expressive capacity. ATT&CK serves as a knowledge repositories of attack behaviors, delineating attacks from the perspective of attack intent through tactics, techniques, and sub-techniques, representing a mid-level knowledge model [1]. For instance, *TA0005* delineates the tactic of *defense evasion*, which examines how attackers evade detection during the intrusion process. *T1078* describes the technique employed by attackers to leverage existing accounts for defense evasion, persistence, privilege escalation and initial access. *T1078.001* focuses on the sub-technique of attackers acquiring credentials for default accounts. In contrast, CAPEC functions as a database of common attack patterns, categorizing and enumerating attacks based on attack mechanisms, albeit with a lower level of abstraction compared to ATT&CK. For example, *CAPEC-70* outlines how attackers exploit common or default usernames and passwords to gain unauthorized access to systems. CWE categorizes common software and hardware vulnerabilities, like *CWE-521*, which highlights the absence of mandated strong passwords. CVE, a publicly accessible repository of real vulnerabilities, details specific instances of vulnerabilities, such as *CVE-2020-4574*, which exposes a vulnerability in IBM Tivoli Key Lifecycle Manage resulting from weak password requirements. Both CWE and CVE belong to low-level knowledge bases [1]. CVE is considered to have a finer granularity, making it relatively more detailed compared to CWE. There are interrelations among the knowledge within these four major knowledge repositories. For example, *T1078.001* and *CAPEC-70* describe the same type of attack behavior and can be linked through the *RelatedTo* relation. *CVE-2020-4574* is a specific *weak password* vulnerability of *CWE-521*, linked through the *BelongTo* relation, as shown in fig 1. However, as attack methods evolve, the knowledge relations between knowledge bases are incomplete. Existing knowledge repositories are stored in document form, requiring significant human and temporal resources for updates [2], posing challenges in promptly addressing missing knowledge relations.

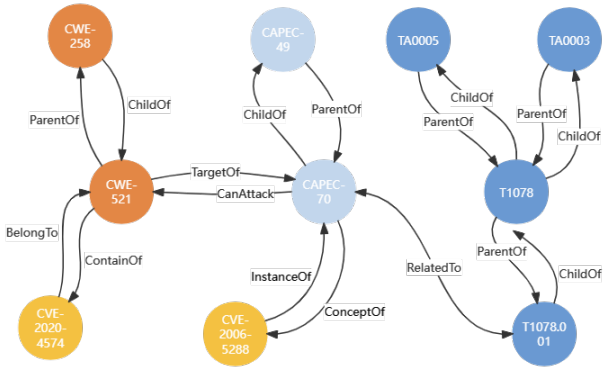


Fig. 1: An illustration of cybersecurity knowledge graph across ATT&CK, CAPEC, CWE and CVE.

Knowledge graph represents a structured semantic knowledge base that elucidates concepts and their interrelations in

*Le Wang is the corresponding author, e-mail: wangle@gzhu.edu.cn.

979-8-3503-9136-7/24/$31.00 ©2024 IEEE
DOI 10.1109/DSC63484.2024.00043

the real world, offering attributes like inferability, scalability, and linkability. Knowledge graph embedding is proved to effectively predict the implicit relations and finds wide applications in tasks such as text matching [3], question answering [4], and path reasoning [5]. By constructing a knowledge graph of cybersecurity knowledge and employing knowledge graph embedding methods, it becomes feasible to efficiently predict missing knowledge. While traditional knowledge graph embedding methods primarily focus on modeling structural patterns like symmetry, inversion, and composition, their capacity to model hierarchical structures remains limited [6]. Previous research on cybersecurity knowledge graph reasoning has succeeded in crafting versatile security knowledge graphs from diverse perspectives but has overlooked the modeling and reasoning of hierarchical structures within knowledge graphs.

In this paper, we have created a cybersecurity knowledge graph based on the latest knowledge bases of ATT&CK, CAPEC, CWE, and CVE. Our proposed knowledge graph encapsulates hierarchical, symmetric, inverse, and compositive structures to model knowledge relations within and between different repositories, encompassing 4122 entities, 15 relation types, and 12088 triples. Furthermore, we introduce a spherical hierarchical knowledge graph embedding model, employing dual-phase rotation to model symmetry, inversion, and composition structures, along with modulus scaling to capture the hierarchical structure of the knowledge graph. Our experiments on the cybersecurity knowledge graph demonstrate the superior performance of our model in completing missing knowledge, affirming the efficacy of our approach.

In summary, our contributions are as follows:

- We have established a knowledge graph called CBKG that bridges attack tactics and vulnerability weaknesses, incorporating hierarchical, symmetric, inverse, and compositive structural patterns to model knowledge relations within and between four network knowledge bases.
- We introduce a spherical hierarchical knowledge graph embedding model called SHKE, leveraging dual-dimensional phase rotation and modulus scaling to model reasoning for hierarchical, symmetric, inverse, and compositive structural patterns.
- We measure the performance of our proposed model through experimental validation and highlight the advantages of our knowledge graph embedding models in completing missing knowledge.

II. Related Work

In this section, we will describe the related work from two perspectives: cybersecurity knowledge reasoning and knowledge graph embeddings.

A. Cybersecurity Knowledge Reasoning

Modeling and reasoning with knowledge graphs in cyberspace have been popular research topics in recent years. Researchers have constructed knowledge graphs in cyberspace based on various public repositories of cybersecurity knowledge and proposed multiple reasoning methods to address different objectives.

Reasoning methods based on knowledge graph embeddings apply classical embedding models to cybersecurity knowledge graphs. Han [7] constructed a knowledge graph based on the CWE knowledge repository and used the TransE model in conjunction with Word2vec to represent and learn the structural and textual information in the knowledge graph, reasoning for CWE link prediction. Xiao [2] built a knowledge graph based on CWE, CVE, and CAPEC knowledge repositories and proposed an embedding and prediction method that combines the CNN and TransH models. Yuan et al. [8] constructed a security knowledge graph based on three major public threat knowledge repositories and designed a text-enhanced graph attention network model to represent the structure and textual knowledge of the security knowledge graph, emphasizing the importance of triple neighbor knowledge. Wang [9] also constructed a software security knowledge graph based on CWE, CVE, and CAPEC knowledge repositories. They proposed a knowledge graph representation learning method that combines Sentence-BERT and GAT, enabling link prediction and classification tasks for knowledge graph completion.

Machine learning is increasingly being applied to the processing and reasoning of unstructured cybersecurity knowledge [10] [11] [12] [13]. Pingle et al. [14] constructed a cybersecurity knowledge graph by acquiring open-source threat intelligence and used a feedforward neural network model to predict relations between cybersecurity entities. Jia [15] extracted entities from attack data using machine learning, constructed a cybersecurity knowledge graph with ontology, and derived new relation rules through a path ranking algorithm. Hemberg [16] aggregated eight public threat knowledge repositories, including CWE, CVE, and CAPEC, to build a security knowledge graph. They created bidirectional relationship knowledge by converting unidirectional links and used language models and supervised machine learning to infer new association relations.

The above methods for modeling cybersecurity knowledge graphs have constructed versatile security knowledge graphs from different perspectives. However, the current research on modeling neglects the hierarchical relations between knowledge repositories, and the proposed reasoning methods have limited capability in inferring the hierarchical structure information of knowledge graphs.

B. Knowledge Graph Embedding

Knowledge graph embedding is a crucial area within the field of knowledge graphs, focusing on learning representations of knowledge graphs. Knowledge graph embedding models emphasize the fundamental structural information of knowledge graphs, modeling structural patterns in triples. These models are primarily categorized into translational distance models, semantic matching models, and hyperbolic space models.

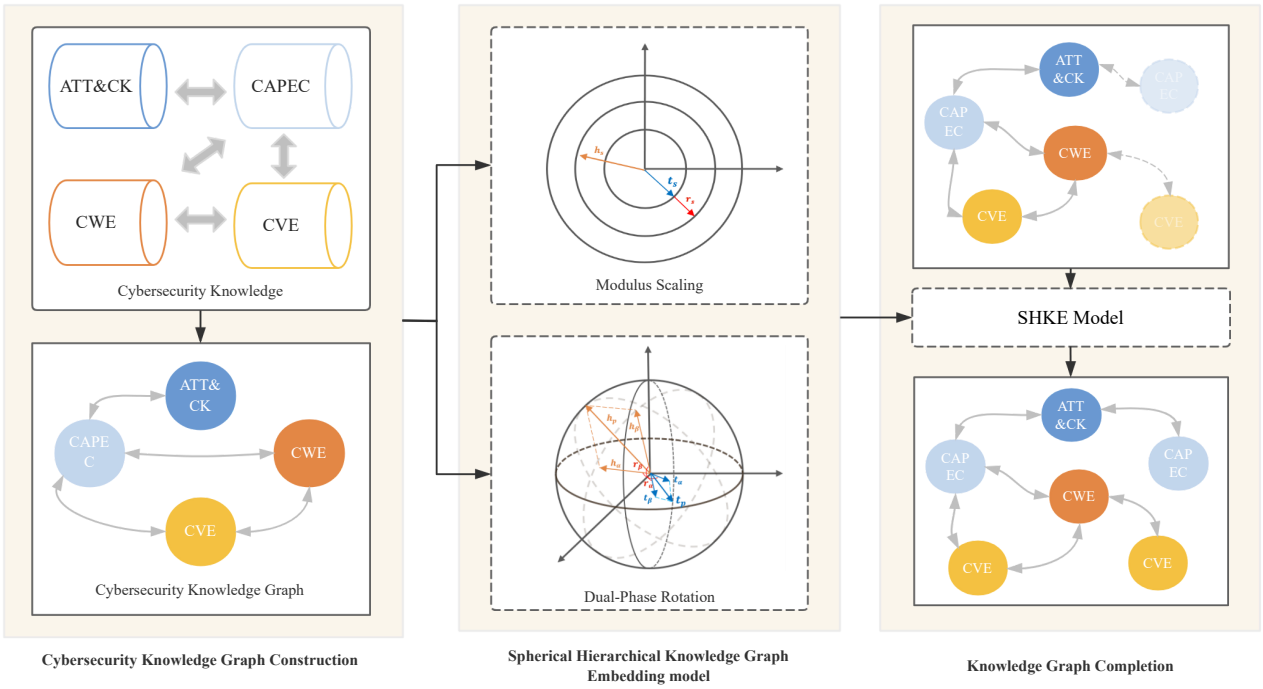


Fig. 2: The architecture of our approach, including three components: the construction of a cybersecurity knowledge graph, the embedding of spherical hierarchical knowledge graphs, and the completion of knowledge graphs.

The translation distance model represents relations as translations from head to tail. TransE [17], widely used in researches, firstly defines relations as translations from head entities to tail entities. However, TransE [17] struggles to effectively model one-to-many, many-to-one, and many-to-many relations. To address this limitation, TransH [18] projects entities onto hyperplanes specific to each relation, while TransR [19] uses relation-specific projection matrices to map entities to distinct spaces, expanding the mapping space. Building on TransR [19], TransD [20] constructs separate projection matrices for head and tail entities. Moreover, RotatE [21] projects triples into complex space, treating relations as rotations from head entities to tail entities, comprehensively modeling symmetry, inversion, and composition patterns. These models face challenges in hierarchy modeling. To tackle this issue, HAKE [6] maps entities to polar coordinates to further incorporate hierarchical structures through modulus and phase.

Semantic matching models use similarity-based scoring functions to align entities' latent semantics with relations manifested in the embedding space. RESCAL [22] represents each relation as a full-rank matrix, capturing the similarity between head and tail entities. However, to prevent overfitting, subsequent models like DistMult [23] set the relation matrix as a diagonal matrix, and ANALOGY [24] as an normal matrix. These simplified models may lack expressive power and have limitations in representing general knowledge graphs. Models like ComplEx [25] introduce complex embeddings to better capture asymmetric and inverse relations.

Hyperbolic space, with proven hierarchical representation advantages, has garnered attention. Models like MuRP [26] merge multi-relational graph data into the Poincaré ball model in hyperbolic space, effectively modeling hierarchical structures through hyperbolic distance translations, outperforming Euclidean space models in low-dimensional spaces. However, MuRP [26] struggles with encoding certain relational logical patterns. To enhance logical modeling, AttH [27] introduces hyperbolic reflections, rotations, and an attention mechanism to represent relation patterns effectively, capturing both hierarchical and logical structures. LMH-PKE [28] uses an extended Poincaré ball for embedding, capturing hierarchical structures through hyperbolic transformations while encoding multiple relations in Euclidean space.

These models focus on entities and relations as fundamental units, emphasizing hierarchy, symmetry, inversion, and composition patterns, with potential for further enhancement in modeling various complex structural patterns.

III. Approach

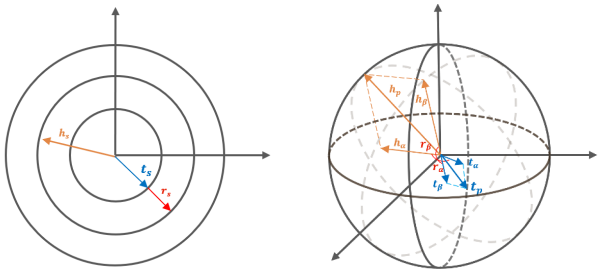
In this section, we present an overview of our methodology, as depicted in Fig. 2. Our approach consists of three components: cybersecurity knowledge graph construction, spherical hierarchical knowledge graph embedding, and the completion of knowledge graphs. The objective of the spherical hierarchical knowledge graph embedding(SHKE) model is to com-

plete entities and relations within the generated cybersecurity knowledge graph(CBKG).

A. Cybersecurity Knowledge Graph Construction

We initiated the construction of a cybersecurity knowledge graph based on the knowledge extracted from the ATT&CK, CAPEC, CWE, and CVE repositories, illustrated in Fig. 1. The knowledge graph is represented as *G=(E,R,T)*, where *E*, *R*, and *T* denote the sets of entities, relations, and triples, respectively. An entity in this context signifies a node within the knowledge graph, a relation signifies an edge connecting conceptual nodes, and a triple encapsulates a piece of knowledge. Each triple is defined as *T=(h,r,t)*, where *h*, *r*, and *t* represent the head entity, relation, and tail entity, indicating the existence of a relation *r* between the head entity *h* and the tail entity *t*. For instance, the triple *(CWE-521, ContainOf, CVE-2020-4574)* signifies an inclusion relation between *CWE-521* and *CVE-2020-4574*. This representation is visually depicted in Fig. 1, where the relation edge BelongTo connects the head entity node *CVE-2020-4574* with the tail entity node *CWE-521*.

The knowledge graph encompasses 4122 entities, 15 types of relations, and 12088 triples. The entity set encompasses 787 ATT&CK entities, 475 CAPEC entities, 934 CWE entities, and 1932 CVE entities. The relation set comprises 15 types, where we delineate 8 internal relations and 7 external relations, as detailed in Table I. Internal relations denote connections within the same knowledge repository, emphasizing unique relations specific to each repository, such as *CanPrecede* and *CanFollow* solely existing within the CAPEC repository. Additionally, common internal relations like *ChildOf* and *ParentOf* are shared across different repositories, illustrating common hierarchical structures within each repository. External relations represent connections between distinct knowledge repositories, bridging knowledge from varying hierarchical levels. Modeling external relations essentially entails capturing the hierarchical structure between different repositories. Each external relation is precisely defined to ensure accurate modeling. For example, the relation *ContainOf* delineates the hierarchical association between CWE and CVE. By effectively modeling both internal and external relations, we establish comprehensive knowledge linkages within and across cybersecurity knowledge repositories.



(a) Modulus scaling part. (b) Dual-phase rotation part.

Fig. 3: The spherical hierarchical knowledge graph embedding model(SHKE).

B. Spherical Hierarchical Knowledge Graph Embedding

To effectively model structural patterns within knowledge graphs, we introduce the Spherical Hierarchical Knowledge Graph Embedding model, SHKE. SHKE comprises two key components: modulus scaling and dual-phase rotation, tailored to capture hierarchical, symmetric, inverse, and compositive structural patterns. To differentiate between different embeddings, we utilize $\mathbf{h}, \mathbf{r}, \mathbf{t}$ to denote the embedding of head entities, relations, and tail entities, use $\mathbf{h}_s, \mathbf{r}_s, \mathbf{t}_s$ for modulus embeddings, and $\mathbf{h}_\theta, \mathbf{r}_\theta, \mathbf{t}_\theta$ for phase embeddings of head entities, relations, and tail entities. The overall embedding is formed by concatenating the modulus embedding and the phase embedding.

Modulus Scaling. Within the cybersecurity knowledge graph we have constructed, hierarchical structures are prevalent. For instance, the triple *(CVE-2006-5288, InstanceOf, CAPEC-70)* signifies that the CAPEC entity *CAPEC-70* encompasses the CVE entity *CVE-2006-5288*, where *CAPEC-70* holds a higher hierarchical position than *CVE-2006-5288*. To address these hierarchical structures, we employ modulus scaling. Specifically, we posit that entities at higher levels possess small moduli. When dealing with head and tail at varying levels, we apply scaling transformations based on relation modulus. As depicted in Fig. 3a, for head entities positioned lower in the hierarchy than the tail entity, the relation modulus embedding $\mathbf{r}_s$ scales down the modulus embedding of the head entity $\mathbf{h}_s$ to a smaller embedding, approaching the modulus embedding of the tail entity $\mathbf{t}_s$. This ensures that the modulus embeddings of head and tail entities at different hierarchy levels are adjusted to be more similar in magnitude.The modulus scaling operation is represented by the multiplication of modulus embeddings, and we can formulate the modulus part as follows:

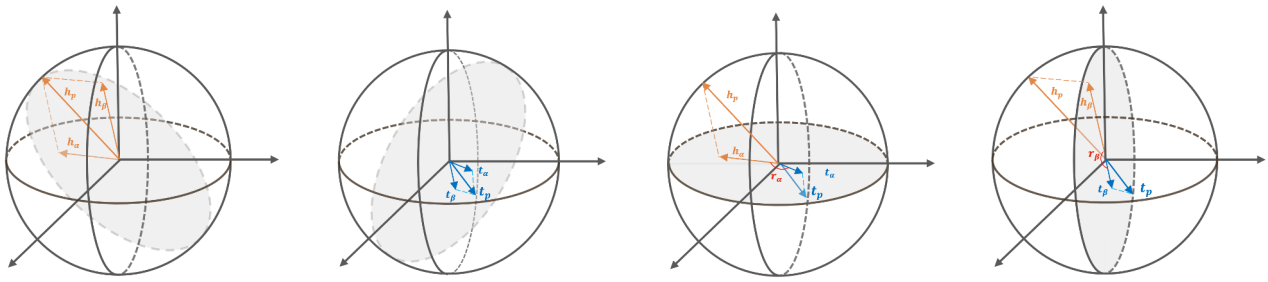
$$\mathbf{h}_s \circ \mathbf{r}_s = \mathbf{t}_s, \tag{1}$$

The corresponding modulus distance function is defined as:

$$d_{\mathbf{s}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h}_s \circ \mathbf{r}_s - \mathbf{t}_s\|, \tag{2}$$

where $\circ$ denotes element-wise multiplication of embeddings.

Dual-Phase Rotation. The Cybersecurity knowledge graph also exhibits symmetric, inverse, and compositive structural patterns. For example, the triple *(T1078.01, RelatedTo, CAPEC-70)* indicates the connection between the ATT&CK sub-technique *T1078.01* and the attack pattern *CAPEC-70*, which can also be expressed as *(CAPEC-70, RelatedTo, T1078.01)*, where the relation *RelatedTo* demonstrates symmetry. Inversion involves the interchange of head and tail in a triple, necessitating the use of an opposite relationship for representation, such as *ChildOf* and *ParentOf* relations. Composition refers to the formation of a new triple by combining two triples with identical relations. For instance, the triples *(CAPEC-165, CanFollow, CAPEC-644)* and *(CAPEC-644, CanFollow, CAPEC-122)* can be amalgamated to form the triple *(CAPEC-165, CanFollow, CAPEC-122)*. Utilizing a dual-dimensional phase embedding, we construct a spherical



(a) Head phase embedding (b) Tail phase embedding (c) Horizontal rotation (d) Vertical rotation

Fig. 4: Detailed decomposition of the dual-phase rotation component of our SHKE model.

space to model the transformations of these structural patterns through dual-phase rotation, as shown in Fig. 3b. Specifically, we decompose the phase embedding of head entities $\mathbf{h}_\theta$, relations $\mathbf{r}_\theta$, and tail entities $\mathbf{t}_\theta$ into sub-phase embeddings $\mathbf{h}_\alpha, \mathbf{r}_\alpha, \mathbf{t}_\alpha$ in the horizontal plane and $\mathbf{h}_\beta, \mathbf{r}_\beta, \mathbf{t}_\beta$ in the vertical plane, as shown in Fig. 4a and Fig. 4b. The rotation of phase in space can also be decomposed into rotations in the horizontal plane and the vertical plane. As illustrated in Fig. 4c, the horizontal subphase embedding of the head $\mathbf{h}_\alpha$ is obtained by applying a rotational transformation to the horizontal subphase embedding of the relation $\mathbf{r}_\alpha$, resulting in the horizontal subphase embedding of the tail entity $\mathbf{t}_\alpha$. Similarly, in the vertical plane, the phase embeddings of the head $\mathbf{h}_\beta$ and the tail entity $\mathbf{t}_\beta$ are transformed by the vertical subphase embedding of the relation$\mathbf{r}_\beta$, as depicted in Fig. 4d. Phase rotation is represented by the addition of phase embeddings, and we can formulate the dual-phase part as follows:

$$\begin{cases} (\mathbf{h}_\alpha + \mathbf{r}_\alpha) \bmod 2\pi = \mathbf{t}_\alpha \\ (\mathbf{h}_\beta + \mathbf{r}_\beta) \bmod 2\pi = \mathbf{t}_\beta, \end{cases} \tag{3}$$

The corresponding dual-phase distance function is defined as:

$$\begin{aligned} d_\theta(\mathbf{h}, \mathbf{r}, \mathbf{t}) = &\left\| \sin\left((\mathbf{h}_\alpha + \mathbf{r}_\alpha - \mathbf{t}_\alpha) \right) \right\| \\ &+ \left\| \sin\left((\mathbf{h}_\beta + \mathbf{r}_\beta - \mathbf{t}_\beta) \right) \right\|, \end{aligned} \tag{4}$$

where $\sin()$ is the sine function used to calculate the distance between phases. The dual-dimensional phase spherical space we have constructed further expands the modeling space for complex structures under controllable conditions, enhancing the modeling capability for symmetric, inverse, and compositive structural patterns. By integrating the distance function of the modulus and phase components, the corresponding score function of the SHKE model is as follows:

$$f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = -d_s(\mathbf{h}, \mathbf{r}, \mathbf{t}) - \lambda d_\theta(\mathbf{h}, \mathbf{r}, \mathbf{t}), \tag{5}$$

where λ is a parameter that learned by the model, serving to balance the weights of the phase and modulus scores.

C. Model Training

Negative sampling is an effective method for enhancing the learning capability of knowledge graph embedding models. In a knowledge graph, the triples that are already known and present are considered positive triples, representing correct knowledge. For each positive triple (h, r, t), the negative sampling method samples entities to replace either the head entity or the tail entity, creating wrong triples (h', r, t'), which we refer to as negative triples. To optimize our distance-based SHKE model, we utilize a loss function based on self-adversarial negative sampling [21]:

$$\begin{aligned} \mathcal{L} = &- \sum_{(h,r,t) \in T} \log \sigma(\gamma - f(\mathbf{h}, \mathbf{r}, \mathbf{t}) \\ &- \sum_{(h',r,t') \in T'} p(h', r, t') \log \sigma(f(\mathbf{h}', \mathbf{r}, \mathbf{t}') - \gamma), \end{aligned} \tag{6}$$

where γ represents a fixed margin, σ denotes the sigmoid function, (h, r, t) corresponds to the positive triplets, while (h', r, t') corresponds to the negative triplets. Additionally,

$$p\left(h'_j, r, t'_j\right) = \frac{\exp \alpha f\left(\mathbf{h}'_j, \mathbf{r}, \mathbf{t}'_j\right)}{\sum_{(h',r,t') \in T'} \exp \alpha f\left(\mathbf{h}', \mathbf{r}, \mathbf{t}'\right)}, \tag{7}$$

represents the probability distribution for negative sampling, where $\left(h'_j, r, t'_j\right)$ is the ith negative triplet, and α represents the sampling temperature.

IV. Experiment

In this section, we provide a detailed overview of our experimental setup followed by a demonstration of the effectiveness of the proposed model SHKE on the constructed dataset CBKG.

A. Experimental Settings

a) Dataset: Our research is based on constructing a cybersecurity knowledge graph using the ATT&CK, CAPEC, CWE, and CVE databases. We extracted knowledge IDs from the knowledge base using regular expressions, such as ATT&CK IDs, CAPEC IDs, CWE IDs, and CVE IDs, as entities. The relation construction involved two parts: extracting existing knowledge relations within the knowledge base as internal relations and defining external relations between knowledge bases based on expert knowledge. The cybersecurity knowledge graph(CBKG) we built comprises 4122 entities (781 ATT&CKs, 475 CAPECs, 934 CWEs, 1932 CVEs), 15 types of relations (8 internal, 7 external), and 12088 triples. To further validate the effectiveness of the proposed SHKE model, we evaluated SHKE's performance on three commonly used

TABLE I: Distribution of entities and relations in our cybersecurity knowledge graph(CBKG).

Type	Entity				Relation		Triple
	ATT&CK	CAPEC	CWE	CVE	Internal Relation	External Relation	
Number	781	475	934	1,932	8	7	12,088

knowledge graph datasets: WN18RR [29], FB15K-237 [30], and YAGO3-10 [31]. WN18RR [29] is a subset extracted from WordNet [17], which captures knowledge associations among English words. FB15K-237 [30] is a subset of FB15K [17], which represents real-world knowledge with various topics and types. YAGO3-10 [31] is a subset of YAGO3 [31], a knowledge graph constructed based on information extracted from Wikipedia. Details of these datasets are summarized in Table II.

b) Hyperparameters: We used the Adam [32] optimizer with embedding dimensions $d \in \{100, 200, 500\}$, batch sizes $b \in \{512, 1024, 2048\}$, self-adversarial sampling temperatures $\alpha \in \{0.5, 1.0\}$, fixed margins $\gamma \in \{3, 5, 6, 9, 12\}$, and balancing weight $\lambda \in \{0.5, 1.0, 1.5\}$.

c) Evaluation Settings: We evaluate the performance of our SHKE model through a link prediction task [17], which involves predicting another element when two elements of a triple are known. For example, predicting the most likely tail entity *t* for a given head entity *h* and relation *r*. The evaluation process involves sequentially disrupting every triples in the test set by replacing the head entity or tail entity with all other entities to create candidate triples. Scores are then calculated for all candidate triples using the score function, and all candidate triples are ranked based on these scores. The ranking of the correct entity is recorded as the ranking of the triple. To ensure that predictions are made for unknown triples, the filter setting [17] is used to filter out positive triples known in the dataset from the candidate triples. The model's performance is evaluated based on the rankings of all predicted triples in the test set, using evaluation metrics such as Hits@n and Mean Reciprocal Rank (MRR).Hits@n represents the proportion of triples within the top n rankings among all predicted triples. For example, Hits@10 represents the proportion of triples within the top 10 rankings. Hits@1, Hits@3, and Hits@10 are the main focus of evaluation. Additionally, MRR is used as an evaluation metric, which represents the average reciprocal rank of each predicted triple.

TABLE II: Number of entities, relations, and observed triples in each split for datasets.

Dataset	E	R	Train	Valid	Test
CBKG	4,122	15	10,274	725	1089
WN18RR	40,493	11	86,835	3,034	3,134
FB15K-237	14,541	237	272,115	17,535	20,466
YAGO3-10	123,182	37	1,079,040	5,000	5,000

B. Main Results

In this section, we compare the performance of our proposed model (SHKE) with existing methods including TransE, TransD, TransR, DistMult, ComplEx, ANALOGY, RotatE, and HAKE. Table III presents the performance of the SHKE model and several previous models.Our SHKE model achieves optimal performance on four evaluation metrics compared to existing methods.

The CBKG dataset we constructed includes hierarchical, symmetric, inverse, and compositive structural patterns. Among the aforementioned models, HAKE is the only one that models all structural patterns. Compared to HAKE, SHKE demonstrates a 0.5% improvement in MRR, a 0.4% improvement in Hits@1, a 1.2% improvement in Hits@3, and a 0.6% improvement in Hits@10. SHKE can be considered as an enhancement of HAKE, expanding the modeling capabilities for symmetry, inversion, composition, and other patterns, thus demonstrating stronger modeling capabilities. While RotatE models symmetry, inversion, and combination patterns through rotation, it cannot effectively capture hierarchical structures. In comparison, SHKE outperforms RotatE with 0.6% improvement in MRR, a 1.3% improvement in Hits@3, and a 0.7% improvement in Hits@10. These results highlight the superiority of SHKE over RotatE and HAKE, as it not only models hierarchical structural patterns better but also exhibits better modeling capabilities for symmetry, inversion, and composition patterns. Consequently, SHKE is better equipped to capture complex structural patterns within the cybersecurity knowledge graph and demonstrates advanced capabilities in completing missing security knowledge.

The evaluation is also performed on three other commonly used datasets: WN18RR, FB15k-237, and YAGO3-10. WN18RR primarily consists of symmetric and hierarchical patterns. FB15k-237 contains more diverse relationship types, including general relationships that do not conform to specific structural patterns like hierarchy and symmetry. YAGO3-10, on the other hand, is a large-scale dataset with prominent hierarchical structural properties. Among the existing models, HAKE achieves the best performance on these three datasets. However, SHKE surpasses HAKE in performance on WN18RR, with a 0.1% improvement in Hits@1 and a 0.3% improvement in Hits@10. On FB15k-237 dataset, SHKE demonstrates a 0.1% improvement in MRR, a 0.3% improvement in Hits@3, and a 0.2% improvement in Hits@10. Furthermore, SHKE outperforms HAKE on YAGO3-10 with a 0.3% improvement in MRR, a 0.5% improvement in Hits@3 and a 0.4% improvement in Hits@10. The limitations of HAKE in terms of its modeling space and computational constraints hinder its performance on large-scale knowledge graphs. In contrast, SHKE expands the modeling space while controlling computational costs, demonstrating superior performance in representing large-scale knowledge graphs.The

TABLE III: Evaluation results on CBKG, WN18RR, FB15k-237 and YAGO3-10 datasets. The best scores are in bold.

	CBKG				WN18RR				FB15k-237				YAGO3-10			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	.782	.738	.815	.850	.226	-	-	.501	.294	-	-	.465	-	-	-	-
TransD	.756	.701	.799	.837	-	-	-	-	.284	.181	.329	.485	-	-	-	-
TransR	.627	.524	.703	.787	-	-	-	-	-	-	-	-	-	-	-	-
DistMult	.633	.500	.743	.862	430	.390	.440	.490	.241	.155	.263	.419	.340	.240	.380	.540
ComplEx	.796	.744	.840	.867	.440	.410	.460	.510	.247	.158	.275	.428	.360	.260	.400	.550
ANALOGY	.767	.702	.828	.860	.403	.361	.428	.474	-	-	-	-	-	-	-	-
RotatE	.831	.809	.840	.871	.476	.428	.492	.571	.338	.241	.375	.533	.495	.402	.550	.670
HAKE	.832	.809	.841	.872	.496	.452	.516	.582	.346	.250	.381	.542	.545	.462	.596	.694
SHKE	**.837**	**.813**	**.853**	**.878**	**.497**	**.452**	**.516**	**.585**	**.347**	**.250**	**.384**	**.544**	**.548**	**.463**	**.601**	**.698**

outstanding performance of SHKE on these three commonly used datasets validates its ability to effectively model structural patterns in knowledge graphs. It is worth noting that SHKE's versatility allows it to be applied to knowledge graphs in various domains, enabling knowledge reasoning beyond network security.

V. Conclusion

This paper presents the construction of a cybersecurity knowledge graph based on the ATT&CK, CAPEC, CWE, and CVE knowledge repositories. To complete the missing knowledge in the cybersecurity knowledge graph, we propose a novel knowledge graph embedding model called SHKE. We conducted a series of experiments on four real datasets to evaluate the performance of our model. The experimental results demonstrate the superiority of our model over existing approaches, specifically in terms of knowledge modeling for complex structural patterns such as hierarchy, symmetry, inversion, and composition. Furthermore, our model serves as a general knowledge graph embedding model for predicting entity and can be applied to knowledge graphs in various domains. It is worth noting that knowledge graphs consisting solely of basic triples may exhibit certain semantic deficiencies. In the future, we plan to expand the textual description information of the knowledge graph, model both the semantic and structural aspects of the knowledge graph, and enhance the accuracy of knowledge completion. This will enable us to construct more comprehensive knowledge graphs to facilitate cybersecurity knowledge analysis.

Acknowledgment

This work is supported in part by Guangdong Basic and Applied Basic Research Foundation (2023A1515011698), Guangdong High-level University Foundation Program (SL2022A03J00918), Major Key Project of PCL (PCL2022A03), and National Natural Science Foundation of China (Grant No. 62372137).

References

[1] Blake E. Strom, A. Applebaum, Doug Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody Thomas. Mitre attck : Design and philosophy. 2018.

[2] Huining Xiao, Xing Zhenchang, Xiaohong Li, and Hao Guo. Embedding and predicting software security entity relationships: A knowledge graph based approach. *Lecture Notes in Computer Science,Lecture Notes in Computer Science*, Jan 2019.

[3] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, Nov 2014.

[4] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, AlexanderJ. Smola, and Le Song. Variational reasoning for question answering with knowledge graph. *Cornell University - arXiv,Cornell University - arXiv*, Sep 2017.

[5] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 5329–5336, Aug 2019.

[6] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 3065–3072, Jun 2020.

[7] Zhuobing Han, Xiaohong Li, Hongtao Liu, Zhenchang Xing, and Zhiyong Feng. Deepweak: Reasoning common software weaknesses via knowledge graph embedding. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Mar 2018.

[8] Liu Yuan, Yude Bai, Zhenchang Xing, Sen Chen, Xiaohong Li, and Zhidong Deng. Predicting entity relations across different security databases by using graph attention network. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, Jul 2021.

[9] Yan Wang, Xiaowei Hou, Xiu Ma, and Qiujian Lv. *A Software Security Entity Relationships Prediction Framework Based on Knowledge Graph Embedding Using Sentence-Bert*, page 501–513. Jan 2022.

[10] Zhiqiang Zhang, Le Wang, Guangyao Chen, Zhaoquan Gu, Zhihong Tian, Xiaojiang Du, and Mohsen Guizani. Stg2p: A two-stage pipeline model for intrusion detection based on improved lightgbm and k-means. *Simulation Modelling Practice and Theory*, page 102614, Nov 2022.

[11] Yongyan Guo, Zhengyu Liu, Cheng Huang, Jiayong Liu, Wangyuan Jing, Ziwang Wang, and Yanghao Wang. *CyberRel: Joint Entity and Relation Extraction for Cybersecurity Concepts*, page 447–463. Jan 2021.

[12] Hyeonseong Jo, Jinwoo Kim, Phillip Porras, Vinod Yegneswaran, and Seungwon Shin. Gapfinder: Finding inconsistency of security information from unstructured text. *IEEE Transactions on Information Forensics and Security*, 16:86–99, Jan 2021.

[13] Zhiqiang Zhang, Le Wang, Junyi Zhu, Dong Zhu, Zhaoquan Gu, and Yanchun Zhang. MIM: A multiple integration model for intrusion detection on imbalanced samples. *World Wide Web (WWW)*, 27(4):47, 2024.

[14] Aditya Pingle, Aritran Piplai, Sudip Mittal, Anupam Joshi, James Holt, and Richard Zak. Relext: relation extraction using deep learning approaches for cybersecurity knowledge graph improvement. In *ASONAM*, pages 879–886. ACM, 2019.

[15] Yan Jia, Yulu Qi, Huaijun Shang, Rong Jiang, and Aiping Li. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering*, 4(1):53–60, 2018. Cybersecurity.

[16] Erik Hemberg, Matthew J. Turner, Nick Rutar, and Una-May O'reilly. Enhancements to threat, vulnerability, and mitigation knowledge for cyber analytics, hunting, and simulations. *Digital Threats: Research and Practice*, page 1–33, Mar 2024.

[17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Le Centre pour la Communication Scientifique Directe - HAL - Inria,Le Centre pour la Communication Scientifique Directe - HAL - Inria*, Dec 2013.

[18] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun 2022.

[19] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun 2022.

[20] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Jan 2015.

[21] Zhiqing Sun, Zhihong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv: Learning,arXiv: Learning*, Feb 2019.

[22] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. *International Conference on Machine Learning,International Conference on Machine Learning*, Jun 2011.

[23] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *Cornell University - arXiv,Cornell University - arXiv*, Dec 2014.

[24] Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. *International Conference on Machine Learning,International Conference on Machine Learning*, Jul 2017.

[25] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. *International Conference on Machine Learning,International Conference on Machine Learning*, Jun 2016.

[26] Ivana Balazevic, Carl Allen, and TimothyM. Hospedales. Multi-relational poincaré graph embeddings. *arXiv e-prints,arXiv e-prints*, May 2019.

[27] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jan 2020.

[28] Dong Zhu, Yao Lin, Haonan Tan, Le Wang, and Zhaoquan Gu. Lightweight multi-semantic hierarchy-aware poincaré knowledge graph embeddings. In *ICPADS*, pages 1358–1364. IEEE, 2023.

[29] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun 2022.

[30] Kristina Toutanova and Danqi 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*, Jan 2015.

[31] Farzaneh Mahdisoltani, Joanna Biega, and FabianM. Suchanek. Yago3: A knowledge base from multilingual wikipedias. *Conference on Innovative Data Systems Research,Conference on Innovative Data Systems Research*, Jan 2013.

[32] DiederikP. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv: Learning,arXiv: Learning*, Dec 2014.