CyberEA: An Efficient Entity Alignment Framework for Cybersecurity Knowledge Graph

Yue Huang, Yongyan Guo, and Cheng Huang(✉)

School of Cyber Science and Engineering, Sichuan University, Chengdu, China
opcodesec@gmail.com

Abstract. The Cybersecurity Knowledge Graph (CKG) represents an invaluable integrated resource designed to support critical functions, including vulnerability mining and defense against cyber threats. Integrating multiple knowledge sources becomes easier with the application of entity alignment, a promising strategy that transcends the boundaries between disparate cybersecurity knowledge bases. Despite this potential, the inherent sparsity and specialization of various CKGs have caused significant performance reductions in current entity alignment methodologies when employed for CKG entity alignment tasks. This paper introduces an effective and efficient entity alignment framework, named CyberEA. This framework utilizes similarity interaction and entity type constraints for an initial entity alignment, supplemented by logical rules for completing the knowledge graph. Subsequently, CyberEA generates entity embeddings from multiple perspectives-name, attribute, and structure. CyberEA implements a Graph Convolutional Network (GCN) to train the entity alignment model and adopts Least Squares Support Vector Machines (LS-SVM) to integrate these perspectives. Experimental validation on multi-type entity datasets reveals that CyberEA consistently surpasses other contemporary entity alignment methods in metrics such as Hits@n, Mean Reciprocal Rank (MRR), and Mean Rank (MR).

Keywords: Entity Alignment · Cybersecurity · Knowledge Graph · Multi-view · Graph Convolutional Network

1 Introduction

The rapid evolution of the Internet has induced a transformation towards a digital and intelligent environment across various industries. Nonetheless, cybersecurity continues to grapple with significant threats posed by the complex, persistent, and covert nature of vulnerabilities, attack patterns, and malware. As a response, knowledge-driven solutions have emerged [8,22,37]. Inspired by the natural language processing (NLP) technology, Google introduced the concept of the knowledge graph (KG) in 2012. A KG is a structured semantic database

H. Duan et al. (Eds.): SecureComm 2023, LNICST 567, pp. 41–62, 2025.
https://doi.org/10.1007/978-3-031-64948-6_3

that symbolically depicts real-world concepts and their interrelations. Its fundamental unit is the "entity-relation-entity" triple.

The cybersecurity knowledge graph (CKG) [22], an application of KG in the cybersecurity realm, amalgamates heterogeneous and voluminous data from multiple sources, enabling the aggregation, analysis, and application of security knowledge. The construction of a typical multi-type CKG [29] relies on named entity recognition [10] and relation extraction [12] techniques, which extract entities and relations from structured and unstructured data. This procedure yields numerous triples visualized through a graph structure, as partially exemplified in Fig. 1.

CKGs draw from diverse data sources including prominent cybersecurity knowledge repositories such as ATT&CK [30], CAPEC [18], D3FEND [15], databases like CVE and CWE for vulnerability enumeration, and open threat intelligence platforms like MISP [40]. ATT&CK [30], developed by MITRE, is a comprehensive cybersecurity action framework acting as a public knowledge base of adversarial tactics and techniques based on real-world attack observations. CAPEC [18] offers a classified dataset cataloguing common attack patterns, while D3FEND [15], launched by MITRE in 2021, focuses on security defense technologies, providing comprehensive enterprise solutions that integrate multiple capabilities. CVE furnishes standardized nomenclature for globally recognized information security flaws or vulnerabilities, while CWE, a community-driven project, lists common software and hardware security vulnerabilities. The continually evolving CKG, SEPSES [16], integrates vulnerabilities and weaknesses from sources such as CAPEC, CPE, CVE, CVSS, and CWE, serving as a query engine for security-related information. MISP [40], an open-source threat intelligence platform, houses an extensive repository of cybersecurity knowledge, facilitating threat analysis and sharing, as well as providing free access to structured cyber threat information and taxonomy.

Despite their utility, individual CKGs often exhibit limitations in scale and rapidly evolving cybersecurity landscape often leads to omission of key information and a bias in field tasks. For instance, different knowledge sources may use varying literal representations for the same threat actor or cybersecurity behavior (attack pattern/course of action). By aligning disparate CKGs, one can broaden the knowledge boundaries, fuse cybersecurity knowledge under various frameworks, and use the resultant fused CKG for tasks such as vulnerability mining and intrusion detection. Furthermore, a robustly developed fused CKG can provide crucial decision support in cybersecurity, facilitating comprehensive threat assessments and rapid deployment of diverse security mechanisms.

Entity alignment, a key technique for knowledge fusion [34,46,48,49], helps ascertain whether two entities from distinct knowledge sources refer to the same object. Traditionally, researchers utilized various string-based characteristics for entity alignment, but advancements in knowledge representation learning have catalyzed the development of entity alignment methods based on it, which demonstrate superior performance. These methods can be bifurcated into two categories: triple embedding [4,6,7,13,14,21,25,31,32,38,41,47,50] and

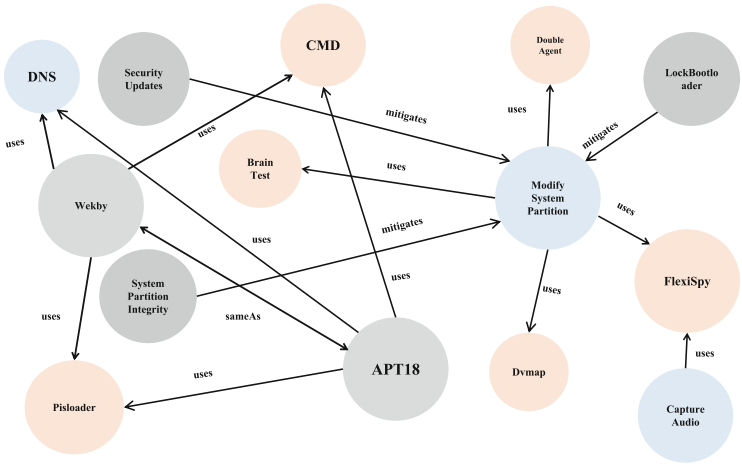

Fig. 1. Partial representation of a cybersecurity knowledge graph. This figure illustrates several types of entities (e.g., threat actor - APT18, attack pattern - capture audio) and relations (e.g., uses, mitigates).

path/neighbor embedding [5,11,20,33,42–44]. These methods have garnered impressive results on general domain datasets [3,27,39].

While considerable progress has been made in entity alignment within knowledge graphs, the task of entity alignment in cybersecurity remains a critical area of focus. In the cybersecurity field, entity alignment involves aligning disparate entities such as vulnerabilities, attack patterns, courses of action, malware, threat actors, and security tools, derived from different knowledge sources. However, due to the unique characteristics of Cybersecurity Knowledge Graphs (CKGs), entity alignment presents several challenges compared to general domain KGs. These challenges, based on our analysis of existing data, include:

Knowledge Representation: Different CKGs may adopt varying modes of knowledge expression. For instance, the same attack pattern could be represented as `Spearphishing via Service` in one knowledge source, and simply as `Spear Phishing` in another.

Structural Differences: Significant structural variances exist among CKGs. For instance, CKGs based on knowledge bases typically display low sparsity and are rich in structural information. In contrast, those built from unstructured or semi-structured data often exhibit high sparsity with numerous isolated nodes, thereby diminishing the effectiveness of traditional entity alignment methods that rely heavily on structural information.

Unique Language Characteristics: The language used in expressing cybersecurity knowledge has unique characteristics. Primarily, it includes a substantial number of domain-specific terms and abbreviations, such as `XSS, SQL, DNS,` and `APT`. For example, `Aurora Panda` does not denote an animal but represents

a specific threat actor known as `ATP17`. Furthermore, the lengths of different entities can exhibit substantial variance. An attack pattern like `Exposure of Version-Control Repository to an Unauthorized Control Sphere` consists of nine words, whereas the threat actor `APT27` is represented by a single word.

To address these challenges, this study introduces CyberEA, an entity alignment framework tailored for Cybersecurity Knowledge Graphs (CKGs), which is designed to bridge semantic gaps and combat sparsity. Our contributions are threefold:

- We present CyberEA, an innovative entity alignment framework specifically designed for CKGs. It incorporates a range of techniques aimed at bridging semantic gaps and combating the issue of sparsity within CKGs.
- To mitigate the prevalent issue of sparsity in CKGs, we integrate a knowledge graph completion module into CyberEA. We also include a multi-view embedding module to enhance the richness of semantic information within CyberEA.
- We empirically evaluate the efficacy of CyberEA using a variety of multi-type entity datasets. The results demonstrate a significant performance improvement over existing methods, with CyberEA achieving a substantial 12.877% improvement in Hits@1 compared to the highest performing pre-existing models.

2 Related Work

2.1 Entity Alignment in the General Domain

Entity alignment is a fundamental technology for knowledge fusion. It is primarily categorized into two methods [34]: triple-based models and path/neighbor models.

Triple embedding methods aim to encode a knowledge graph (KG) into a low-dimensional embedding space. Models such as TransE [4], TransR [21], TransH [41], IPTransE [50], and MTransE [7] perceive a relation as the translation between the head entity and the tail entity. Other models like AttrE [38], IMUSE [13], and JAPE [31] additionally use attributes of entities. MultiKE [47] creates three views: attribute, entity name, and structural information, employing various combination strategies to enrich semantic information. BootEA [32] and SEA [25] address the issue of insufficient seed datasets via semi-supervised training. Path-based embedding exploits the long-term dependency of relations spanning over relation paths, while neighborhood-based embedding utilizes the graph structure constituted by a myriad of relations between entities to model KGs. GCN-Align [42] pioneers the introduction of graph convolutional networks (GCN) [17] into entity alignment, simplifying model complexity and significantly improving alignment effectiveness. Alinet [33] leverages distant neighbors to mitigate the issue of heterogeneous entity neighbors, while RDGCN [43] overcomes the inadequate use of relationships via a dual graph, controlling noise accumulation through highway gates. Other notable works include MuGNN [5], which

encodes KGs based on multi-channels, and RSN4EA [11], which proposes recurrent skipping networks to exploit long-term relational dependencies in KGs. OntoEA [45] utilizes ontology information to facilitate entity alignment. These works primarily rely on general domain datasets such as DBpedia (DBP) [3], Wikidata [39], and YAGO [27]. DBP15k, a cross-language dataset, features four language-specific KGs extracted from English, Chinese, French, and Japanese DBpedia, each containing around 65k–106k entities.

2.2 Entity Alignment in CKGs

Currently, there is scant literature on entity alignment in CKGs, with most research in CKG entity alignment restricted to vulnerability knowledge graphs. The recent Graph Neural Network (GNN)-based model, CEAM [26], introduced for entity alignment, incorporates asymmetric masked aggregation and partition attention, demonstrating superior performance over traditional methods. However, a comprehensive CKG comprises not only vulnerabilities but also vital cybersecurity knowledge domains like attack patterns, course of action, and threat actors. Aligning vulnerabilities alone cannot effectively integrate the complexity of cybersecurity knowledge. Hence, there's a need for further research into entity alignment techniques for multi-entity type CKGs.

2.3 Graph Convolutional Networks

Graph Neural Networks (GNN) [28] are favored for their powerful modeling capabilities for non-Euclidean data. They excel in tasks such as node classification and link prediction. The potent capability of GNNs enables capturing structural relations among non-Euclidean data, yielding more insights compared to the analysis of isolated data. Recent work has given rise to various GNN-based models. Graph Convolutional Networks (GCN) [17], for instance, are a type of neural network architecture that employs graph structure and convolution mode to aggregate node information from neighboring nodes. In our proposed framework, CyberEA, we utilize GCN to model our knowledge graph.

3 Methodology

This section presents CyberEA, a novel entity alignment framework for Cyber Knowledge Graphs (CKGs). Given the structural variability of different CKGs, CyberEA first executes a preliminary entity alignment to accomplish KG completion. Subsequently, CyberEA generates graph embeddings from three dimensions: entity names, entity attributes, and KG structure, utilizing Graph Convolutional Networks (GCNs) for training the alignment model. Lastly, CyberEA adopts Least Squares Support Vector Machines (LS-SVM) to ascertain the optimal weights for each view. The architecture of CyberEA is illustrated in Fig. 2.

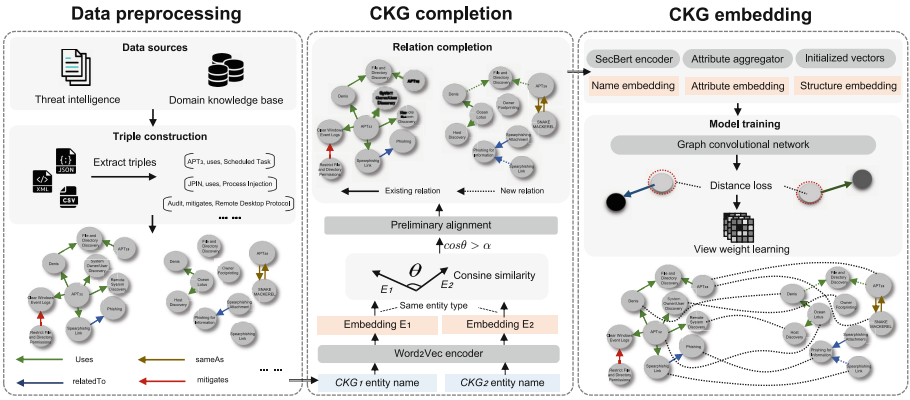

Fig. 2. The CyberEA framework extracts triples from various cybersecurity knowledge bases, completing CKGs through the similarity of entity name embedding and type constraint. Two CKGs are then encoded through three views: entity name, entity attribute, and CKGs' structure. Lastly, CyberEA employs GCN to train the alignment model and learn the weights of different views using LS-SVM.

3.1 Data Preprocessing

This phase elaborates on data preprocessing procedures applied to the datasets. To ensure a standardized representation of information across cybersecurity domains, we utilized the Unified Cyber Ontology (UCO) [36], a community-driven ontology. By adhering to the UCO 2.0 framework and considering the peculiarities of our datasets, we constructed an ontology graph (Table 1) consisting of eight distinct entity types: `Indicator`, `Threat Actor`, `Attack Pattern`, `Malware`, `Tool`, `Campaign`, `Course of Action`, and `Vulnerability`. Furthermore, we identified six types of relations, specifically `relatedTo`, `use`, `sameAs`, `mitigate`, `hasProduct`, and `indicates`.

The raw files, sourced from varied origins, were primarily in JSON, XML, or CSV formats. We performed data processing activities to extract relational triples of the form (entity, relationship, entity) and attribute triples of the form (entity, attribute, value). Entities with non-English names were excluded from the dataset. Additionally, each entity was assigned a unique Universal Unique Identifier (UUID) to streamline identification. Consequently, two Cyber Knowledge Graphs (CKGs) were constructed based on these triples.

3.2 CKG Completion

The significant structural differences between the two CKGs in our dataset substantially impede the utilization of structural information. Therefore, in the next phase, CyberEA completes the two CKGs to yield structures that are relatively rich and informative.

Preliminary Alignment. To facilitate preliminary alignment, we establish two conditions:

Table 1. Six types of relation and eight types of entity.

Relation	Uses, Indicates, AttributedTo, HasVulnerability, Mitigates, SameAs
Entity	Threat-Actor, Attack-Pattern, Course-of-Action, Malware, Vulnerability, Software, Indicator, Campaign

- The aligned entities should be of the same entity type, as indicated by the type attribute of the entity.
- The similarity of the name embedding for the aligned entity should exceed the threshold α.

Each entity's name is processed into a token sequence $X = \{x_1, x_2, x_3, \ldots, x_n\}$, following the removal of stopwords. Word2Vec [24] is a widely-used language model in natural language processing, facilitating unsupervised semantic knowledge learning from large text corpora. The Skip-Gram algorithm, a variant of the Word2Vec model, learns continuous feature representations of words by optimizing the objective of neighborhood preservation. We train the Word2Vec model using the names and descriptions of all entities and apply the Skip-Gram algorithm. This yields a vector representation v for each token with a vector sequence:

$$\{x_1, x_2, \ldots, x_n\} \overset{Skip-Gram}{\longrightarrow} \{vec_1, vec_2, \ldots, vec_n\} \tag{1}$$

For each entity name, its embedding V is derived by taking the average of all token embedding vectors:

$$V = \frac{1}{N} \sum_{i=1}^{N} vec_i \tag{2}$$

Preliminary entity alignment is accomplished by calculating the cosine similarity between the embedding vectors of two entity names. For entities e_i and e_j, the similarity is computed as follows:

$$sim(e_i, e_j) = \frac{V_i \cdot V_j}{|V_i||V_j|} \tag{3}$$

Completion Rules. We model the two CKGs as two graphs, $\mathcal{G}_a$ and $\mathcal{G}_b$, containing entity sets $\mathcal{E}_a$ and $\mathcal{E}_b$, and relation sets $\mathcal{R}_a$ and $\mathcal{R}_b$, respectively. Entity alignment aims to identify a set of identical entities $\mathcal{M} = \{(e_i, e_j) \in \mathcal{E}_a \times \mathcal{E}_b \mid e_i \equiv e_j\}$, where '$\equiv$' denotes the equivalence relationship. In CKGs, only one relation exists between two entities. It is then clear that when $e_{i1} \equiv e_{j1}$ and $e_{i2} \equiv e_{j2}$, we have $r_{i12} \equiv r_{j12}$, where $r_{i12} \in \mathcal{R}_a$, $r_{j12} \in \mathcal{R}_b$.

Relation Completion. We deem entities e_i and e_j as preliminarily aligned when $sim(e_{1i}, e_{2i}) > \alpha$ and both entities share the same type. If a relation r_{i12} exists between e_{i1} and e_{i2}, but no relation is present between e_{j1} and e_{j2}, and

if $e_{i1} \equiv e_{j1}$ and $e_{i2} \equiv e_{j2}$, then we generate a relation r_{j12} between e_{j1} and e_{j2}, where $r_{i12} \equiv r_{j12}$. This operation completes the two CKGs, thus enhancing their structural information. Figure 3(a) provides an example, where entities encircled in red are preliminarily aligned, and the dotted arrow represents the completed relations. We demonstrate the process of relation completion in Algorithm 1.

Algorithm 1. Relation Completion

Require: $\mathcal{G}_a$: graph with entity set $\mathcal{E}_a$ and relation set $\mathcal{R}_a$
Require: $\mathcal{G}_b$: graph with entity set $\mathcal{E}_b$ and relation set $\mathcal{R}_b$
Ensure: $\mathcal{G}_a$, $\mathcal{G}_b$: graphs with completed relations
$\mathcal{M} \leftarrow (e_i, e_j) \mid e_i \in \mathcal{E}_a, e_j \in \mathcal{E}_b, e_i \equiv e_j$ {Entity Alignment}
for $(e_{i1}, e_{j1}) \in \mathcal{M}$ **do**
  for $(e_{i2}, e_{j2}) \in \mathcal{M}$ **do**
    if $r_{i12} \in \mathcal{R}_a$, $r_{j12} \notin \mathcal{R}_b$, and $e_{i1} \equiv e_{j1}$, $e_{i2} \equiv e_{j2}$ **then**
      Add relation r_{j12} between e_{j1} and e_{j2} to $\mathcal{G}_b$
      $r_{i12} \equiv r_{j12}$
    end if
  end for
end for
for $(e_{i1}, e_{j1}) \in \mathcal{M}$ **do**
  for $(e_{i2}, e_{j2}) \in \mathcal{M}$ **do**
    if $r_{i12} \notin \mathcal{R}_a$, $r_{j12} \in \mathcal{R}_b$, and $e_{i1} \equiv e_{j1}$, $e_{i2} \equiv e_{j2}$ **then**
      Add relation r_{i12} between e_{i1} and e_{i2} to $\mathcal{G}_a$
      $r_{i12} \equiv r_{j12}$
    end if
  end for
end for
return $\mathcal{G}_a$, $\mathcal{G}_b$

In subsequent stages, only the newly generated relations are used, and the preliminary aligned entity pairs are disregarded. This means that the preliminary aligned entities do not influence later steps. If any noise is introduced, it is mainly due to erroneous relations (indirect errors) rather than incorrect entity pairs in the preliminary alignment (direct errors). For example, consider the scenario where similarity matching results in preliminary alignment of the entities `Controller Authentication` and `Domain Controller Authentication`. While these are not identical entities and share a `relatedTo` relationship, we utilize the indirect relationship information instead of the direct alignment information. This greatly reduces noise induced by misaligned entities. Therefore, we believe that the benefits of such a completion operation outweigh the potential noise.

It is noteworthy that relation completion is non-transitive, meaning it is not performed iteratively, to minimize the impact of inaccurate connections.

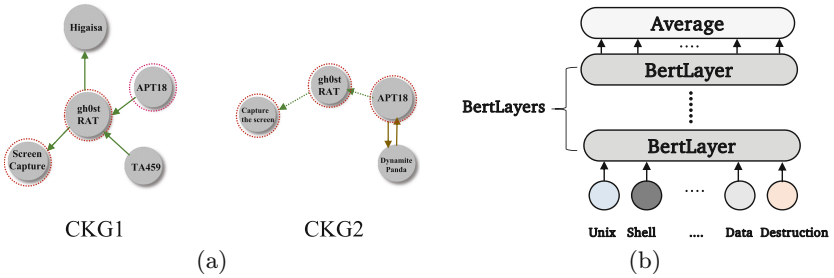

Fig. 3. (a) Example of relation completion. Entities encircled in red are preliminarily aligned, and the dotted arrow indicates the completed relations. (b) SecBert encoder. (Color figure online)

3.3 CKG Embedding

The embedding of CKGs in CyberEA is achieved using Graph Convolutional Networks (GCN), which have demonstrated remarkable efficacy in various graph-based applications. The success of GCNs is attributable to their ability to utilize the feature representations of neighboring nodes when deriving features for a central node. A typical GCN architecture consists of multiple graph convolutional layers, each performing graph convolutional operations on the features propagated from its preceding layer to produce new feature representations. This propagation can be described as a nonlinear transformation, integrating the feature representations of neighboring nodes with the central node to yield a novel feature representation for the latter. The propagation mechanism between two successive GCN layers is expressed as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \tag{4}$$

Here, A represents the adjacency matrix, I the identity matrix, $\tilde{A} = A + I$, $\tilde{D}$ is the degree matrix of $\tilde{A}$, H denotes the node feature matrix, and σ symbolizes the activation function.

Similar to RDGCN [43] and MultiKE [47], we extract entity embeddings from three perspectives: entity name, entity attributes, and CKG structure. The subsequent sections provide a detailed account of the embedding methods utilized for each of these perspectives.

Attribute Embedding. Many traditional methods employ attribute embedding to bolster the similarity measure of entities. According to recent study [34], attribute embedding is generally classified into two types: attribute correlation embedding and attribute literal embedding. While the former accounts for the correlations among attributes, the latter represents attribute information of entities. This paper opts for the literal attribute embedding approach due to its effectiveness in the context of CKGs. For CKGs, entities of the same type do not necessarily share the same attributes, which can lead to potential discrepancies in attribute correlation embedding and introduce noise. For

instance, the threat actor `GoldMax` possesses four attributes, while its identical entity `SUNSHUTTLE` only has one. Given these considerations, we choose to use literal attribute embedding to represent attribute information of entities.

An entity typically comprises multiple attributes and corresponding attribute values. Following the method of previous study [2], we employ an aggregator to obtain the entity attribute embedding. Specifically, for entity e, its attributes $a_1, a_2, ..., a_n$ correspond to the values $v_1, v_2, ..., v_n$. We then construct a word set W from these attributes and attribute values. Each word $w \in W$ is represented by a vector V_w, which can be obtained through pre-training methods (Skip-gram is utilized in our case). $p(w)$ represents the frequency of word w. The aggregator operates as follows:

We first formulate sentences using the direct concatenation of each entity's attributes and generate sentence sets $s \in S$ as:

$$[a_1; v_1; a_2; v_2; ...; a_n; v_n] \tag{5}$$

For each word, we compute its word representation V_w and word frequency $p(w)$, then determine the weight of each word using the formula:

$$weight = \frac{a}{p(w) + a} \tag{6}$$

where a is a hyperparameter. We then calculate the embedding V_s for each sentence s:

$$V_s = \frac{1}{|s|} \sum_{w \in s} \frac{a}{p(w) + a} V_w \tag{7}$$

To mitigate the influence of the common semantics on the embedding, we concatenate all sentence embeddings into a matrix X_v and compute the first singular vector u of X_v. The final attribute view embedding, E_a, is then obtained through the following correction:

$$E_a = V_s - uu^\top V_s \tag{8}$$

Name Embedding. To encapsulate the semantic information encapsulated by entity names, we employ the `SecBert` model for entity embedding. The Bidirectional Encoder Representation from Transformers (BERT) [9], a pre-training language model, has gained substantial recognition for its exceptional performance. Furthermore, several pre-training models, such as `SecBert`, have been specifically designed for the cybersecurity domain [1]. Notably, `SecBert` is a BERT model trained on cybersecurity-specific corpora, such as APTnotes, Stucco-Data, and CASIE, to garner domain-specific knowledge.

To derive entity name embeddings, we encode each token of the entity name using `SecBert`. This operation results in a matrix of dimensions (768, 3), which we then flatten into a 2304-dimensional vector, denoted as V_n. The final name embedding is computed by taking the average of all token embeddings. This

process can be represented as $Avg(F(\text{SecBert}(e_i)))$, where $F(\cdot)$ indicates the flattening operation and $Avg(\cdot)$ calculates the average of the token embeddings.

Structure Embedding. We use random initialization vectors to represent structure embeddings (i.e., relation embeddings) E_s. Unlike attribute embeddings and name embeddings, structure embeddings change continuously during training time.

Finally, we obtain attribute embedding E_a, name embedding E_n and structure embedding E_s.

Connectivity Matrix. Given that entities may exert different degrees of influence on their neighboring entities under different types of relations, we leverage the connectivity matrix to refine the adjacency matrix A. Specifically, we employ the approach proposed in GCN-Align [42] to compute the connectivity matrix. For each relation type r, we determine its forward and reverse connectivity by following the method:

$$\text{forward}(r) = \frac{n_r(h)}{n_r(p)} \quad \text{reverse}(r) = \frac{n_r(t)}{n_r(p)} \tag{9}$$

where $n_r(h)$, $n_r(t)$ is the number of head entities and tail entities in relation r triples, and $n_r(p)$ is the number of triples of relation r. We measure the influence of the i-th entity over the j-th entity as follows:

$$a_{ij} = \sum_{\langle e_i, r, e_j \rangle \in \mathcal{G}} \text{forward}(r) + \sum_{\langle e_j, r, e_i \rangle \in \mathcal{G}} \text{reverse}(r) \tag{10}$$

Calculation of Distance. Entity alignment aims to reduce the vector distance between the same entities in the same embedding space. We measure the distance between e_i and e_j according to the following formula:

$$\begin{aligned} \text{distance}\,(e_i, e_j) = w_n \frac{|E_n\,(e_i) - E_n\,(e_j)\,|}{d_n} \\ + w_a \frac{|E_a\,(e_i) - E_a\,(e_j)\,|}{d_a} + w_s \frac{|E_s\,(e_i) - E_s\,(e_j)\,|}{d_s} \end{aligned} \tag{11}$$

where w_n, w_s, w_a is the weights of different views, and $w_n + w_s + w_a = 1$. $E_n(x), E_a(x), E_s(x)$ is name embedding, attribute embedding and structure embedding respectively. d_n, d_a, d_s is the dimension of embedding.

Training Model. We use some alignment seeds to train CyberEA. The objective of the training process is to minimize the loss function $\mathcal{L}$:

$$\begin{aligned} \mathcal{L} = [\sum_{e_i, e_j \in \mathcal{T}} \sum_{e_i^{'}, e_j^{'} \in \mathcal{T}'} (|E(e_i) - E(e_j)| + \gamma \\ - |E(e_i^{'}) - E(e_j^{'})|)]_+ \end{aligned} \tag{12}$$

where $\mathcal{T}$ is the training data set, γ is the margin hyper-parameter, $e_i^{'}$ and $e_j^{'}$ are negative seeds. It requires that the distance between positive seed pairs be

as close as possible while the distance between negative seed pairs is as large as possible. $E(e)$ is the entity embedding in different views. We use SGD to minimize the above functions, and train the embedding of the entity name, attribute value, and structure information views L_n, L_a, and L_s, respectively.

View Integration. To circumvent the tedious task of manually setting weight hyperparameters, we leverage the Least Squares Support Vector Machines (LS-SVM) model [35] to learn view weights, following the approach suggested in [23]. The objective function of LS-SVM optimization utilizes a binary norm, while equality constraints substitute the inequality constraints typical of the standard SVM algorithm.

To determine the weight of different views, we transform the entity alignment problem into a classification problem. The specific training method is as follows:

$$\mathcal{L}\text{svm} = C\sum_{l=1}^{m}[y_l \cdot \max(0, 1 - \mathbf{w}^\top \mathbf{x}_l) + (1 - y_l)\cdot \max(0, 1 + \mathbf{w}^\top \mathbf{x}_l)] + \frac{1}{2}\mathbf{w}^\top \mathbf{w} \tag{13}$$

where $\mathbf{w} = [w_a, w_n, w_s]$. $\mathbf{x}_l = [sim_n, sim_a, sim_s]$ is similarity vector. The terms sim_n, sim_a, and sim_s denote the name view similarity, attribute view similarity, and structure view similarity of a pair of entities, respectively. We utilize cosine similarity to compute these metrics. y denotes the label: if the entity pairs are positive samples, then $y = 1$, otherwise $y = 0$.

4 Experiments

4.1 Experimental Setup

Datasets. We construct two CKGs containing 16029 entities. The data of CKG1 comes from open knowledge bases, including ATT&CK, CAPEC, CWE, and D3FEND. The data of CKG2 comes from open-source threat intelligence platforms, including MISP, Unit42, etc. The details of the dataset are shown in the Table 2. One reason for combining multiple "open knowledge bases" for CKG1 and multiple "open-source threat intelligence platforms" for CKG2 is to enhance the diversity and comprehensiveness of the datasets. By incorporating data from multiple sources, we can capture a wider range of cybersecurity concepts, attack patterns, and threat actors, and improve the coverage and accuracy of the CKGs. Additionally, different knowledge sources may have different focuses, structures, and vocabularies, and by combining them, we can achieve a more complete and nuanced representation of cybersecurity knowledge. Furthermore, the use of multiple knowledge sources can also help to mitigate the issue of missing or incomplete data that may exist in any one source.

Specifically, we construct relation triples and attribute triples as follows:

- For each entity, we added attributions based on knowledge sources, such as the entity's description, the threat actor's alias, and the vulnerability killchain.

– We tagged each type of entity as an attribute of the entity. For example, `APT27` has a label of `Threat Actor` and `Local Account Monitoring` has a label of `Course of Action`.

Then we obtained alignment seeds based on four ways:

– We utilized open mapping files that are linked to different knowledge sources to acquire the entity alignment seeds. For instance, official mapping files are readily available for CAPEC and ATT&CK, D3FEND and ATT&CK. Since CVE vulnerabilities can be directly aligned by their number, we did not consider them in this study.
– To obtain the seeds, we applied the entity similarity metric Levenshtein [19] and performed a manual filtering procedure. We initially set a threshold to identify all entity pairs with similarity scores greater than the threshold and then conducted a manual filtering process to eliminate entities that do not match. In this process, we utilized attributes for filtering to reduce the manual annotation effort. Specifically, when aligned entities share the same attribute, they must also possess the same attribute value. For instance, a pair of aligned vulnerabilities should have identical `platform` attributes.
– We obtained the alignment seeds for malware, tools, and threat actors based on their entity aliases. For instance, `APT28` has aliases such as `Pawn Storm`, `Sofacy Group`, and `STRONTIUM`.
– Furthermore, we created a cybersecurity abbreviations mapping dictionary (e.g., APT: Advanced Persistent Threat) to facilitate the selection of alignment seeds. Specifically, we converted all abbreviations found in the text into their full names and then applied the similarity matching method in the previous step. This approach proved to be effective in improving the matching accuracy.
– Finally, some identical attack patterns may share common attributes and attribute values, such as the same number in ATT&CK. Based on this observation, we obtained additional alignment seeds.

We removed the duplicate alignment seeds, and finally, we obtained 4116 pairs of alignment seeds. We use the same training/testing split with previous works, 70% for training and 30% for testing.

Table 2. Statistic of dataset.

CKG	Ent	Attr	Triple
CKG1	5816	16713	23429
CKG2	10213	22215	12187

Parameters. We train CyberEA for 5000 epochs, the training GPU is GeForce RTX 2080 Ti (11 GB). We empirically set the hyperparameters according to

GCN-Align's code[1]: $d_n = 200$, $d_v = 50$, $d_s = 150$, $\theta = 10^{-3}$, $\gamma = 3.0$, initial learning_rate = 20, number of negative samples for each positive seed is 5, and the number of GCN layers is 2. Some hyperparameters are chosen by trying different configurations (as shown in Table 3). We use ReLU as an activation function.

Table 3. Hyperparameter configurations.

Hyperparameter	Values	Optimal Value
Window size of `Skip-Gram`	$2, 3, 4, 5$	3
α	$0.75, 0.80, 0.85, 0.90, 0.95$	0.85
Dropout	$0.1, 0.2, 0.3, 0.4, 0.5$	0.2
C	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$	10^{-3}
a	$10^{-4}, 10^{-3}, 10^{-2}$	10^{-3}

Details of Embedding. We utilize the pre-training model `SecBert` for name embedding through the `huggingface` library[2]. The structure of `SecBert` includes an encoder and a CLS token; we extract the encoder and discard the CLS token. There are 6 Bert layers in the encoder. We take its output, a (768, 3) matrix for each word, and flatten it into a 2304-dimensional vector. We obtain attribute embeddings for all entities using the `Skip-Gram` method. Finally, we get two arrays with shapes of (16029, 100) and (16029, 2304) through `Skip-Gram` and `SecBert`, respectively. For the feature matrix in GCN, we use the `Scipy` package to process them into sparse matrices, reducing computation and training time.

Metrics. Hits@n measures the proportion of correctly aligned entity pairs that appear in the top n predicted entity pairs. A higher score indicates better performance in entity alignment. For instance, Hits@1 represents the proportion of correctly aligned entity pairs at the top of the list, while Hits@10 represents the proportion of correctly aligned entity pairs within the top 10.

Mean Reciprocal Rank (MRR) is the average reciprocal rank of correctly aligned entity pairs, indicating how high these pairs appear in the predicted entity pairs list. A higher MRR score signifies better alignment performance. Mean Rank (MR) is the average rank of correctly aligned entity pairs in the predicted entity pairs list, where a smaller score indicates better alignment performance, as correctly aligned pairs appear higher in the predicted list. Mean Reciprocal Rank (MRR) and Mean Rank (MR) compute scores as follows:

$$\mathrm{MRR} = \frac{1}{|S_t|}\sum_{i=1}^{|S_t|}\frac{1}{\mathrm{rank}_i} = \frac{1}{|S_t|}\left(\frac{1}{\mathrm{rank}_1} + \ldots + \frac{1}{\mathrm{rank}_{|S_t|}}\right) \tag{14}$$

[1] https://github.com/1049451037/GCN-Align.
[2] https://huggingface.co/jackaduma/SecBERT.

$$\mathrm{MR} = \frac{1}{|S_t|}\sum_{i=1}^{|S_t|}\mathrm{rank}_i = \frac{1}{|S_t|}\left(\mathrm{rank}_1 + \ldots + \mathrm{rank}_{|S_t|}\right) \tag{15}$$

where S_t is the triple set and rank_i represents the prediction rank of the i-th entity. For MR, a smaller value indicates better alignment, while for MRR, a larger value is preferred (Table 4).

Table 4. Training time on SecBert and Skip-Gram.

Name	Attribute	Avg Time	Hits@1	MR	MRR
Skip-Gram	Skip-Gram	311 s	33.101	175	0.387
SecBert	Skip-Gram	1029 s	43.217	68	0.538
SecBert	SecBert	1770 s	43.471	67	0.542

Time Constraint. The length of attribute information is substantial, making the use of `SecBert` embeddings impractical due to long training times. Furthermore, experimental results indicate that embedding entity attributes with either `SecBert` or `Skip-Gram` does not result in a significant difference (fluctuating within 1%) in the final outcomes. The training time (epoch=5000) for solely embedding names with BERT is 1029 s, whereas embedding both names and attributes with `SecBert` takes 1770 s. As such, the marginal improvement gained from using `SecBert` to embed attributes instead of `Skip-Gram` does not justify the increase in time required. Therefore, to balance effectiveness and time efficiency, we will utilize `Skip-Gram` to embed attribute values.

Baselines. We compared eight entity alignment models to evaluate their effectiveness in entity alignment: GCN-Align, AttrE, BootEA, IPTransE, MultiKE, RDGCN, Alinet, and OntoEA. GCN-Align [42] is the baseline model for GCN-based embedding, while AttrE [38], MultiKE [47], and RDGCN [43] consider literal embedding. BootEA [32] is a semi-supervised model that uses the bootstrapping method, while IPTransE [50] and Alinet [33] both only consider structural information in KGs. MultiKE and RDGCN have shown to yield the best results in some cross-language datasets. OntoEA incorporates ontology information to enhance entity alignment and has shown to outperform RDGCN in some datasets. In addition, we also experimented with several traditional models, such as TransE, TransR, and MtransE. However, the results revealed that their Hits@1 were all less than 1%. Therefore, we excluded these models from the model comparison. We showed the details of these baselines and CyberEA in Table 5 and use open benchmark OpenEA[3] to test the performance of baselines.

[3] https://github.com/nju-websoft/OpenEA.

Table 5. Details of baselines and CyberEA. "Ebd." means embedding.

Method	Relation Ebd.	Attribute Ebd.	Ebd. Distance	Learning
GCN-Align [42]	Neighbor	Correlation	Manhattan	Supervised
AttrE [38]	Triple	Literal	Cosine	Supervised
BootEA [32]	Triple	–	Cosine	Semi-Supervised
IPTransE [50]	Path	–	Euclidean	Semi-Supervised
MultiKE [47]	Triple	Literal	Cosine	Supervised
RDGCN [43]	Neighbor	Literal	Manhattan	Supervised
Alinet [33]	Neighbor	–	Euclidean	Supervised
OntoEA [45]	Triple	Literal	Cosine	Supervised
CyberEA	Neighbor	Literal	Manhattan	Supervised

Table 6. Comparison results.

Method	Hits@1	Hits@5	Hits@10	Hits@50	Hits@100	MR	MRR
GCN-Align [42]	14.842	24.209	27.494	38.686	46.229	223	0.208
AttrE [38]	21.290	33.455	39.051	53.285	61.071	174	0.274
BootEA [32]	21.411	33.577	36.861	48.662	54.380	180	0.288
IPTransE [50]	16.667	27.494	29.805	39.781	45.255	247	0.219
MultiKE [47]	30.340	40.291	42.233	51.456	58.738	166	0.331
RDGCN [43]	27.476	32.875	38.760	44.810	52.108	182	0.308
Alinet [33]	8.151	13.869	16.302	21.655	28.691	315	0.116
OntoEA [45]	7.122	10.488	16.098	25.983	30.010	609	0.108
CyberEA -c	23.978	37.093	43.545	59.015	67.186	198	0.319
CyberEA -b	33.101	47.466	53.510	62.749	69.597	175	0.401
CyberEA	**43.217**	**63.550**	**67.791**	**80.064**	**85.553**	**68**	**0.538**

4.2 Comparative Analysis

As demonstrated in Table 6, CyberEA significantly outperforms mainstream models across all metrics. Specifically, CyberEA's Hit@1 is 12.88% higher than the second most effective model, MultiKE, which translates into a relative increase of 29.8%. Additionally, CyberEA dramatically outperforms RDGCN, with a Hit@1 score that is relatively 57.29% higher.

In terms of entity alignment within the same language, models such as MultiKE and RDGCN, which consider semantic information, generally outperform models like Alinet and IPTransE, which rely exclusively on structural information. The unsupervised model BootEA also exhibits strong performance. By generating potential alignment entities through iterative processing, BootEA effectively mitigates the problem of insufficient seed entity pairs in the Cross-Knowledge Graph (CKG) entity alignment task. However, Alinet's performance

is notably weak, primarily due to the abundance of isolated entities and subgraphs in CKGs. This greatly limits Alinet's ability to aggregate long-distance neighbors. Although OntoEA incorporates ontology information, its effectiveness is limited. This is mainly due to the relatively few types of ontologies in CKGs and the fact that each entity's ontology is unique and certain, regarded as an attribute of the entity. Therefore, the introduction of additional ontology information might not yield significant improvements. These results suggest that models traditionally considered high-performing may not necessarily perform as expected across different datasets.

4.3 Ablation Analysis

Module Ablation. We conducted ablation experiments to evaluate the effectiveness of each module in CyberEA. Specifically, we evaluated the performance of two variations of CyberEA: CyberEA -c, which omits the KG completion module, and CyberEA -b, which employs `Skip-Gram` instead of `SecBert`.

Table 6 reveals that both modules significantly contribute to the entity alignment task. The KG completion module plays a crucial role in improving CyberEA's performance by reducing graph sparsity, resulting in a 19.239% increase in Hits@1. Additionally, using `SecBert` embeddings enhances the performance by 10.116%, indicating that semantic information is critical in CKGs, and the `SecBert` model can provide more significant semantic benefits than the word2Vec model.

View Ablation. To explore the effect of different views on entity alignment results, we removed different views separately: -stru is CyberEA without structure view, -attr is CyberEA without attribute view, -name is CyberEA without name view. The results are shown in Table 7.

From the ablation results of different views, we found that the name view played the most important role in entity alignment, which was 11.627%, further reflecting the important role of semantic information in the entity alignment task in CKGs. In addition, the promotion effect of structure view and attribute view on entity alignment is not so remarkable, which are 3.04% and 3.815%, respectively. Overall, all three views facilitate the result of entity alignment. The result also proves that semantic information is much more needed than structural information in the entity alignment of the same language.

Table 7. View ablation results.

Model	Hits@1	MR	MRR
CyberEA -stru	40.177	110	0.502
CyberEA -attr	39.402	117	0.441
CyberEA -name	31.590	180	0.391
CyberEA	**43.217**	**68**	**0.538**

5 Case Study

In our case study, we selected two representative models to explore: GCN-Align and RDGCN. GCN-Align acts as the baseline model for CyberEA and RDGCN, yet it primarily relies on structural information, with no integration of semantic data. On the other hand, RDGCN is a GCN-based model that incorporates the literal embedding of entity names and attributes, leading to enhanced performance on cross-lingual datasets. Table 8 showcases some instances of entity alignments across various types in CyberEA. The data clearly demonstrate that CyberEA is highly effective in aligning a diverse range of entity types.

CyberEA Versus GCN-Align. Our comparison with GCN-Align revealed a discrepancy in entity alignment. For instance, Pair 1: `Masquerading : Space after Filename`, `Adding a Space to a File Extension` was not aligned under GCN-Align. However, this pair was aligned in both CyberEA and RDGCN. This is attributable to the GCN-Align model's lack of name and attribute embedding, making it difficult for the model to learn semantic information.

CyberEA Versus RDGCN. Conversely, in our comparison with RDGCN, we found that entities `Orz` and `AIRBREAK` were not aligned in RDGCN but were aligned under CyberEA. Specifically, `Orz` functions as an alias for `AIRBREAK` and has several relations, such as `use` with `Process Hollowing` and `use` with `Windows Command Shell` in CKG1. However, in CKG2, `AIRBREAK` is an isolated entity. Notably, the successful alignment of `Orz` and `AIRBREAK` following CKG completion demonstrates that CyberEA can enhance structural information through CKG completion. Additionally, our observations suggest that CyberEA correctly matches a higher number of threat actors, malware, and tools that share aliases when compared to RDGCN. This highlights the fact that CyberEA prioritizes not just semantic information, but also structural data.

Table 8. Case study examples.

Entity Pairs	CyberEA	GCN-Align	RDGCN
Entity1: Masquerading: Space after Filename Entity2: Adding a Space to a File Extension	✓	×	✓
Entity1: Gooligan Entity2: Ghost Push	✓	×	×
Entity1: APT19 Entity2: Codoso	✓	×	×
Entity1: Remote System Discovery Entity2: Host Discovery	✓	×	×
Entity1: Orz Entity2: AIRBREAK	✓	×	×
Entity1: APT3 Entity2: Gothic Panda	✓	×	×

✓: aligned, ×: not aligned.

6 Conclusion

This paper delves into the task of entity alignment in Cybersecurity Knowledge Graphs (CKGs), aiming to consolidate knowledge from multiple sources. To achieve this, we introduce CyberEA, a robust framework designed for entity alignment in CKGs. Comprising a Knowledge Graph (KG) completion module and a multi-view embedding module, CyberEA effectively tackles the challenges of sparsity and semantic heterogeneity inherent to CKGs. As a result, it delivers substantial performance enhancements over traditional models across our experimental datasets. Ablation experiments further enable us to discern the individual contributions of each module to CyberEA's performance.

As part of future work, we plan to delve into the pipeline for constructing CKGs from unstructured data, and address the challenge of entity alignment among multiple CKGs. Additionally, we aim to extend the alignment process to other forms of unstructured cybersecurity data, beyond just the knowledge graph.

Acknowledgement. This work was supported in part by National Key Research and Development Program of China (No. 2021YFB3100500), Sichuan Science and Technology Program (No. 2023YFG0162), and National Natural Science Foundation of China (61902265).

References

1. Aghaei, E., Niu, X., Shadid, W., Al-Shaer, E.: SecureBERT: a domain-specific language model for cybersecurity. In: Li, F., Liang, K., Lin, Z., Katsikas, S.K. (eds.) SecureComm 2022. LNICST, vol. 462, pp. 39–56. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25538-0_3
2. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations (2017)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
5. Cao, Y., Liu, Z., Li, C., Li, J., Chua, T.S.: Multi-channel graph neural network for entity alignment. arXiv preprint arXiv:1908.09898 (2019)
6. Chen, M., Tian, Y., Chang, K.W., Skiena, S., Zaniolo, C.: Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. arXiv preprint arXiv:1806.06478 (2018)
7. Chen, M., Tian, Y., Yang, M., Zaniolo, C.: Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. arXiv preprint arXiv:1611.03954 (2016)
8. Conti, M., Dargahi, T., Dehghantanha, A.: Cyber threat intelligence: challenges and opportunities. In: Dehghantanha, A., Conti, M., Dargahi, T. (eds.) Cyber Threat Intelligence. Advances in Information Security, vol. 70, pp. 1–6. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73951-9_1

9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Gao, C., Zhang, X., Han, M., Liu, H.: A review on cyber security named entity recognition. Front. Inf. Technol. Electron. Eng. **22**(9), 1153–1168 (2021)
11. Guo, L., Sun, Z., Hu, W.: Learning to exploit long-term relational dependencies in knowledge graphs. In: International Conference on Machine Learning, pp. 2505–2514. PMLR (2019)
12. Guo, Y., et al.: CyberRel: joint entity and relation extraction for cybersecurity concepts. In: Gao, D., Li, Q., Guan, X., Liao, X. (eds.) ICICS 2021. LNCS, vol. 12918, pp. 447–463. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86890-1_25
13. He, F., et al.: Unsupervised entity alignment using attribute triples and relation triples. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) DASFAA 2019. LNCS, vol. 11446, pp. 367–382. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18576-3_22
14. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: 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), pp. 687–696 (2015)
15. Kaloroumakis, P.E., Smith, M.J.: Toward a knowledge graph of cybersecurity countermeasures. Corporation, Editor (2021)
16. Kiesling, E., Ekelhart, A., Kurniawan, K., Ekaputra, F.: The SEPSES knowledge graph: an integrated resource for cybersecurity. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11779, pp. 198–214. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30796-7_13
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
18. Kotenko, I., Doynikova, E.: The CAPEC based generator of attack scenarios for network security evaluation. In: 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol. 1, pp. 436–441. IEEE (2015)
19. Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, pp. 707–710. Soviet Union (1966)
20. Li, C., Cao, Y., Hou, L., Shi, J., Li, J., Chua, T.S.: Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2723–2732 (2019)
21. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI Conference on Artificial Intelligence (2015)
22. Liu, K., Wang, F., Ding, Z., Liang, S., Yu, Z., Zhou, Y.: Recent progress of using knowledge graph for cybersecurity. Electronics **11**(15), 2287 (2022)
23. Liu, Z., Cao, Y., Pan, L., Li, J., Chua, T.S.: Exploring and evaluating attributes, values, and structures for entity alignment. arXiv preprint arXiv:2010.03249 (2020)
24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
25. Pei, S., Yu, L., Hoehndorf, R., Zhang, X.: Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In: The World Wide Web Conference, pp. 3130–3136 (2019)

26. Qin, Y., Liao, X.: Cybersecurity entity alignment via masked graph attention networks. arXiv preprint arXiv:2207.01434 (2022)
27. Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., Weikum, G.: YAGO: a multilingual knowledge base from Wikipedia, wordnet, and geonames. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 177–185. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_19
28. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
29. Shen, G., Wang, W., Mu, Q., Pu, Y., Qin, Y., Yu, M.: Data-driven cybersecurity knowledge graph construction for industrial control system security. Wirel. Commun. Mob. Comput. **2020** (2020)
30. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: MITRE ATT&CK: design and philosophy. In: Technical report. The MITRE Corporation (2018)
31. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 628–644. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_37
32. Sun, Z., Hu, W., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI, vol. 18, pp. 4396–4402 (2018)
33. Sun, Z., et al.: Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 222–229 (2020)
34. Sun, Z., et al.: A benchmarking study of embedding-based entity alignment for knowledge graphs. arXiv preprint arXiv:2003.07743 (2020)
35. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**, 293–300 (1999)
36. Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A.: UCO: a unified cybersecurity ontology. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
37. Tao, Y., Li, M., Hu, W.: Research on knowledge graph model for cybersecurity logs based on ontology and classified protection. In: Journal of Physics: Conference Series, vol. 1575, p. 012018. IOP Publishing (2020)
38. Trisedya, B.D., Qi, J., Zhang, R.: Entity alignment between knowledge graphs using attribute embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 297–304 (2019)
39. Vrandečić, D., Krötzsch, M.: WikiData: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
40. Wagner, C., Dulaunoy, A., Wagener, G., Iklody, A.: MISP: the design and implementation of a collaborative threat intelligence sharing platform. In: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, pp. 49–56 (2016)
41. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
42. Wang, Z., Lv, Q., Lan, X., Zhang, Y.: Cross-lingual knowledge graph alignment via graph convolutional networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 349–357 (2018)
43. Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., Zhao, D.: Relation-aware entity alignment for heterogeneous knowledge graphs. arXiv preprint arXiv:1908.08210 (2019)

44. Wu, Y., Liu, X., Feng, Y., Wang, Z., Zhao, D.: Jointly learning entity and relation representations for entity alignment. arXiv preprint arXiv:1909.09317 (2019)
45. Xiang, Y., Zhang, Z., Chen, J., Chen, X., Lin, Z., Zheng, Y.: OntoEA: ontology-guided entity alignment via joint knowledge graph embedding. arXiv preprint arXiv:2105.07688 (2021)
46. Zeng, K., Li, C., Hou, L., Li, J., Feng, L.: A comprehensive survey of entity alignment for knowledge graphs. AI Open **2**, 1–13 (2021)
47. Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., Qu, Y.: Multi-view knowledge graph embedding for entity alignment. arXiv preprint arXiv:1906.02390 (2019)
48. Zhao, X., Zeng, W., Tang, J., Wang, W., Suchanek, F.: An experimental study of state-of-the-art entity alignment approaches. IEEE Trans. Knowl. Data Eng. (2020)
49. Zhao, X., Zeng, W., Tang, J., Wang, W., Suchanek, F.M.: An experimental study of state-of-the-art entity alignment approaches. IEEE Trans. Knowl. Data Eng. **34**(6), 2610–2625 (2020)
50. Zhu, H., Xie, R., Liu, Z., Sun, M.: Iterative entity alignment via knowledge embeddings. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (2017)