electronics MDPI

Article

A Knowledge Graph Completion Algorithm Based on the Fusion of Neighborhood Features and vBiLSTM Encoding for Network Security

Wenbo Zhang [1,*], Mengxuan Wang [1], Guangjie Han [2], Yongxin Feng [1] and Xiaobo Tan [1]

[1] College of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China; skd13361375572@163.com (M.W.); fengyongxin@263.net (Y.F.); tanxiaobo@sylu.edu.cn (X.T.)
[2] Department of Internet of Things Engineering, Hohai University, Changzhou 213022, China; hanguangjie@gmail.com
* Correspondence: zhangwenbo@sylu.edu.cn

Abstract: Knowledge graphs in the field of network security can integrate diverse, heterogeneous, and fragmented network security data, further explore the relationships between data, and provide support for deep analysis. Currently, there is sparse security information in the field of network security knowledge graphs. The limited information provided by traditional text encoding models leads to insufficient reasoning ability, greatly restricting the development of this field. Starting from text encoding, this paper first addresses the issue of the inadequate capabilities of traditional models using a deep learning model for assistance. It designs a vBiLSTM model based on a word2vec and BiLSTM combination to process network security texts. By utilizing word vector models to retain semantic information in entities and extract key features to input processed data into BiLSTM networks for extracting higher-level features that better capture and express their deeper meanings, this design significantly enhances understanding and expression capabilities toward complex semantics in long sentences before inputting final feature vectors into the KGC-N model. The KGC-N model uses feature vectors combined with graph structure information to fuse forward and reverse domain features and then utilizes a Transformer decoder to decode predictions and complete missing information within the network security knowledge map. Compared with other models using evaluation metrics such as MR, MRR demonstrates that employing our proposed method effectively improves performance on completion tasks and increases comprehension abilities toward complex relations, thereby enhancing accuracy and efficiency when completing knowledge graphs.

Keywords: cybersecurity; knowledge graph completion; text encoding; vBiLSTM; KGC-N

check for updates

Citation: Zhang, W.; Wang, M.; Han, G.; Feng, Y.; Tan, X. A Knowledge Graph Completion Algorithm Based on the Fusion of Neighborhood Features and vBiLSTM Encoding for Network Security. *Electronics* **2024**, *13*, 1661. https://doi.org/10.3390/electronics13091661

Academic Editor: Domenico Rosaci

Received: 25 March 2024
Revised: 15 April 2024
Accepted: 19 April 2024
Published: 25 April 2024

1. Introduction

In contemporary society, the rapid development of big data technologies has led to a complex cyber environment, and the cybersecurity issue has become more serious [1]. The security state of cyberspace is the core support for critical information infrastructure. It is directly related to national security and public welfare [2]. To assess the cybersecurity defense ability of operators and ensure the normal operation of critical information infrastructures, the establishment of a security assessment indicator system for critical information infrastructures is crucial [3]. Despite the announcement of a three-tier indicator system for the security of critical information infrastructures by national authorities, the specificity and granularity of these assessment indicators can be further improved [4]. A feasible solution consists of employing knowledge graph technology to construct a cybersecurity knowledge graph and exploring related reasoning techniques to obtain a more refined four-tier security assessment indicator system [5,6].

The objective of completing a knowledge graph is to deduce absent entities or connections within the graph by leveraging existing knowledge or incorporating novel, plausible triples, thereby enhancing its comprehensiveness [7]. Although several large open-domain knowledge graphs exist, such as Wikidata, Freebase, OpenCyc, and Geonames, they are not yet complete [8]. This is particularly true in specialized domains such as cybersecurity, where, despite appearing confined, constant evolution occurs with the emergence of new knowledge [9]. An almost complete knowledge graph is significant for higher-level applications such as question-answering systems, where the ability to provide accurate answers depends on the knowledge graph that contains the relevant information [10]. Consequently, an evolving and comprehensive cybersecurity knowledge graph is essential to support various applications in this field [11].

Since a cybersecurity knowledge graph contains more domain-specific professional knowledge than open-domain knowledge graphs, it is sparser, which limits its application range [12]. The existing knowledge graph completion techniques include semantic matching models based on tensor/matrix factorization such as RESCAL and ComplEx, neural network models such as ConvE, and translation models such as TransE and their extensions [13]. Even though these techniques for enriching knowledge graphs have the potential to transform an initially sparse graph into a more comprehensive one, generic completion techniques may not be suitable for cybersecurity knowledge graphs [14].

Because the network security knowledge graph is sparse, representation learning in the network security field also faces insufficient structural information [15,16]. It has been shown that there are three main issues in current representation learning. Firstly, the existing models only represent a certain type of entity and do not consider cases where entities have multiple types. Secondly, the existing models use various encoding methods for entity descriptions that may not be suitable for processing textual information in network security knowledge bases. Thirdly, most of the current knowledge representation learning models are based on translation or tensor factorization with overly simple scoring functions and limited representation ability [17]. In response to these challenges, this paper first constructs a network security dataset for training and evaluating models. Compared with the existing datasets, the one built in this study includes more types of entities related to network security [18]. It designs a suitable encoding method based on the features of network security texts. Finally, it optimizes a translation model to adapt it for completing applications in network security knowledge graphs [19,20].

This paper’s primary contributions can be outlined as follows:

1. A network security text encoding model based on vBiLSTM is proposed. It extracts primary word-level features using the word2vec model from entity descriptions and then inputs word2vec-initialized vectors into BiLSTM to obtain advanced long-sentence features. The word vectors output by the two models are concatenated based on certain dimensions to obtain a distributed entity description word vector that contains more semantics. This allows for enhancing the precision of text encoding within the domain of network security.
2. A knowledge graph completion model, referred to as Knowledge Graph Completion with Fusing Neighborhood Information (KGC-N), is proposed. Initially, it learns entity semantic features via a knowledge graph embedding algorithm and then uses a graph attention network for multi-hop secure entity neighborhood features in the security knowledge graph, fusing them with the initial features. Finally, it uses a Transformer decoder to decode and predict the fused feature vectors. The use of text information after feature fusion allows for improving the accuracy of the network security knowledge graph completion and effectively addresses the challenge of information sparsity in the network security knowledge graph.

The structure of the remainder of this paper is organized as follows. Section 2 reviews the recent studies on cybersecurity knowledge graphs. Section 3 details the vBiLSTM model. Section 4 introduces the overall framework and theoretical composition of the KGC-N model. Section 5 presents the training results and comparative tests of the model. Finally, Section 6 presents the conclusion.

2. Related Work

The completion of cybersecurity knowledge graphs is widely studied in the cybersecurity field. Many researchers have studied various information sources, applied several techniques such as natural language processing, graph algorithms, and deep learning, and constructed more comprehensive and precise cybersecurity knowledge graphs.

For instance, Wang et al. [20] developed a knowledge graph completion model, referred to as CSNT, for completing cybersecurity data. This model uses BiLSTM to capture the interactive information between entities and relations. It models the relationships between entities by combining neural networks and tensor decomposition, employing a Pearson mixture network to control the generation of joint vectors. Moreover, it introduces a novel self-distillation strategy to reduce catastrophic forgetting during model training. After learning the relational patterns among entities in cyberspace detection intelligence, the model can be used to uncover the knowledge that is not found in the latter and complete missing or erroneous information in records. Finally, it predicts the future state of cyberspace by perceiving and understanding it.

Liu et al. [21] proposed an automatic Cyber Threat Intelligence Analysis method, referred to as K-CTIAA. This method extracts threat actions from unstructured CTI using pre-trained models and knowledge graphs. It supplements related knowledge in the knowledge graph to corresponding positions in CTI through knowledge querying and insertion. This helps the pre-trained models understand the semantics of cybersecurity terminology and extract threat actions. By introducing a visibility matrix and modifying the self-attention computation equation, K-CTIAA reduces the adverse impact of knowledge insertion, which is commonly referred to as the knowledge noise issue. It can provide suggestions for defense against attacks by mapping corresponding countermeasures with digital tools, which improves the performance of automatic threat intelligence analysis and holds significance for handling cybersecurity threats.

Qi et al. [22] introduced knowledge graphs into the detection of composite cyberattacks and constructed a cybersecurity knowledge graph based on known attack knowledge. They conducted a correlation analysis of real-time data to reconstruct the attack process. They proposed an algorithm for multi-dimensional data correlation analysis based on dynamic clustering mechanisms and an attack chain completion method based on optimal path queries. Consequently, they effectively enriched the cybersecurity knowledge graph, tackling low analysis efficiency due to redundant data and addressing missing and misinterpreted data in the collected ones. The cybersecurity knowledge graph that they constructed can automatically respond to composite attacks discovered through mining. However, their algorithm has a large computational load and high parameter requirements, which may affect the accuracy and efficiency of the attack chain mining.

Piplai et al. [23] extracted entities by building a custom-named entity recognizer for After Action Reports (AARs), which is referred to as the Malware Entity Extractor (MEE). They also constructed a neural network to predict the relationships between malware entities. When predicting entity triples and their relationships, they defined entity-relation sets in the cybersecurity knowledge graph. They then merged similar entities to improve the constructed AAR knowledge graph. This fusion helps complete the intelligence extracted from multiple documents and reports. The merged CKG contains knowledge from various AARs. In addition, because of this fusion completion process, security analysts can enhance the accuracy of a knowledge graph by performing queries and retrieval.

Kaiser et al. [24] built a multi-tiered threat knowledge base using data from multiple threat intelligence sources, which is referred to as AttackDB. It correlates high-level ATT&CK techniques with low-level telemetry existing in behavioral malware reports. They also proposed an Attack Hypothesis Generator that relies on knowledge graph traversal algorithms as well as various link prediction methods to complete missing information and automatically infer ATT&CK techniques in observable network attacks. Their algorithm can generate accurate adversarial technology hypotheses based on AttackDB, with improved precision and efficiency in cybersecurity knowledge graph completion tasks. It can also automate the attack hypothesis generation process. Fang Y et al. [25] proposed a cybersecurity entity recognition model called CyberEyes, which combines a graph convolutional neural network with a BiLSTM model to extract non-local dependencies at both the context and graph levels. The performance of this model on cybersecurity corpus is higher than the traditional CNN-BiLSTM-CRF model. To enhance the capability of AI-based cyber defense systems in capturing, detecting, and preventing known and future attacks, Sills M et al. [26] proposed a system that generates higher quality graphical representations by augmenting graph embedding techniques to generate various medical device vulnerability intelligence resources and known vulnerability threat intelligence resources. Jia Y et al. [27] designed a cybersecurity ontology covering assets, vulnerabilities, and attacks, and constructed a cybersecurity knowledge base based on a quintuple model of inference rules using machine learning methods.

Huaijun Shang [28] constructed a cybersecurity domain ontology based on a vulnerability repository and adopted rule-based and lexicon feature-based approaches for specific entities to improve the effectiveness of cybersecurity entity recognition, respectively, and ultimately realized the updating and visualization of the cybersecurity knowledge base. Tong Wang et al. [29] studied the knowledge graph construction technology of threat intelligence, proposed a model that can automate the extraction of entities and relationships for threat intelligence, and realized the visual display of a threat intelligence knowledge graph. Jiayi Peng et al. [30] proposed a BiLSTM-CRF model using an active learning approach to improve the accuracy of the named entity recognition task in the small sample information security domain. Ruobin Zhang et al. [31] proposed a BiLSTM-CRF model for the problem of security vulnerability entity recognition using a lexicon to correct the recognition results, which significantly reduces the cost of manually selecting features while achieving better performance.

Yingjie Xu [32] designed a naming recognition method applied to the Chinese cybersecurity text corpus by extracting local features, using BiLSTM for contextual feature extraction, obtaining the input feature representation of the model, and, finally, using CRF for sequence annotation, thus completing the task of naming entity recognition more accurately in the Chinese cybersecurity domain.

We selected some models by scholars and their corresponding model characteristics, applied in the Table 1 below, to provide a more concise and clear display.

Table 1. Correlation model characterization table.

Authors	Contribution	Key Features
Liu [21]	K-CTIAA: Cyber Threat Intelligence Analysis	Utilizes pre-trained models and knowledge graphs; reduces knowledge noise.
Qi [22]	Knowledge graph for composite cyber-attacks	Enhances graphs with dynamic clustering and path queries; high computational load.
Kaiser [24]	AttackDB: Multi-tiered threat knowledge base	Correlates ATT&CK techniques; uses knowledge graph traversal for hypothesis generation.
Sills M [26]	AI-based cyber defense system	Augments graph embedding techniques for better graphical representations.
Huaijun Shang [28]	Cybersecurity ontology from vulnerability repository	Enhances entity recognition with rule-based and lexicon feature-based methods.
Tong Wang [29]	Automation in threat intelligence knowledge graphs	Automates extraction of entities and relationships; visual display of knowledge graph.
Ruobin Zhang [31]	BiLSTM-CRF model for vulnerability recognition	Uses lexicon for better accuracy; reduces manual feature selection cost.
Yingjie Xu [32]	Naming recognition for Chinese cybersecurity text	Extracts features using BiLSTM and CRF for precise entity recognition.

It can be deduced from the presented literature review that the existing feature fusion knowledge graph completion methods for information gaps in cybersecurity knowledge graphs are limited. This paper proposes a completion method based on the KGC-N model, which ensures high accuracy in knowledge graph completion while implementing the feature fusion of cybersecurity entities.

3. Text Information Encoding Model Based on vBiLSTM

This paper constructs a comprehensive cybersecurity knowledge graph based on the STIX 2.1 standard. The latter is a framework defining STIX domain objects including attack patterns, campaign activities, reports, threat actors, tools, and threat indicators. By integrating multiple authoritative cybersecurity databases such as CAPEC, CVE, CWE, CPE, and ATT&CK, the knowledge graph involves 65 entity types and 114 relationship types, providing a complete representation of the potential cyber threats. The construction process involves crawling open-source cybersecurity repositories for entity data, forming the knowledge graph, and extracting and linking various entities. Each entity and relationship is assigned a unique identifier denoted by 'Record Identifier' and having the following format: '#<Cluster Number>:<Position in Cluster>', where 'Cluster Number' denotes the data cluster to which the record belongs and 'Position in Cluster' represents its specific location within this cluster. The entity attributes in the knowledge base include descriptions, names, and aliases. They provide detailed information for each entity. This robust construction method lays a solid foundation for the study presented in this paper. To create a cybersecurity dataset, data are extracted from existing open-source cybersecurity repositories as follows:

(1) In the analysis of cyber-attack chains, nodes labeled as 'adversary_entity' are considered as initial exploration points because these activities often originate from malicious actors. By constructing MATCH queries, the system can cross graph structure paths related to cybersecurity breaches and link various node types, such as 'Vulnerability_Exploit', 'Intrusion_Signature', 'Intrusion_Tactic', 'Intrusion_Pattern', and 'Incident_Report'. This aims to reveal the tactics of attackers, the way they exploit system vulnerabilities, and the risks and damages of their successful attacks. During path traversal, besides the initial node, other node types are kept undefined to prevent confusion and ambiguity about the starting point.

(2) The extracted nodes are used as root nodes, and Cypher statements are written to extract some edges connected to them, which results in triples including the root node.

(3) In entity information processing, the 'description_text' attribute of entities is primarily used as the information source. If this attribute is incomplete, the 'entity_name' attribute is used instead. The entity descriptions in the security knowledge base involve multiple languages, including Chinese and English. Although the Chinese and English entries may have the same meaning, their representations in the vector space are different. To address the inconsistency in vector semantics caused by language differences, English descriptions are used for standardizing the processing of entity descriptions. An example is shown in Figure 1, which has an accurate description.

(4) Splitting the dataset into training, validation, and testing sets is a crucial step in machine learning projects. It ensures that the model is trained, fine-tuned, and ultimately evaluated on different subsets of data, utilizing an adequate amount of training data to train the model while also mitigating the risk of overfitting due to insufficient data. Additionally, using a validation set allows for timely validation of the trained model to observe training progress and performance improvements. Finally, the testing set is utilized to assess the model's generalization ability. Therefore, in this paper, the dataset was split into training, validation, and testing sets in an 80:15:5 ratio to ensure reasonable data volumes for each subset and facilitate effective model training.

vector space
Chinese descriptions
vector space
English descriptions

Figure 1. The description information of the entity.

The encoding of the text is the main step in knowledge representation learning, which integrates additional entity information. The text information encoding model aims to transform the characteristic information of entities into vector form. The resultant entity vectors should retain the semantics of the original information as much as possible and encompass additional features. Entity embeddings that have integrated other information can possess richer semantic characteristics, which provides the knowledge representation learning model with enhanced inferential ability.

(1) vBiLSTM Model Design

The encoding of entity descriptions in the cybersecurity domain usually relies on word vector models such as Skip-gram, which is suited for processing shorter sentences by simplifying them into average word vector representations. However, cybersecurity-related texts often incorporate longer sentences or even entire paragraphs. This causes a challenge for the average word vector approach in fully capturing the completeness of the entity information. Therefore, when designing an encoding model, it is essential to consider this characteristic of cybersecurity entity texts to ensure a comprehensive and precise representation of the entity information.

The text information encoding model design involves two main aspects. Firstly, the model employs the word2vec word vector model for converting entity description information into word vectors. The main advantage of word2vec lies in its ability to retain the semantic information in entity descriptions and extract key word-level features. Compared with methods that randomly initialize vectors, the distributed word vectors generated by word2vec provide a deeper and more precise exploration of the complex semantics embedded in entity descriptions. However, it has certain limitations when dealing with long sentences. Although it succeeds in capturing word-level features, it falls short in expressing the overall semantics of longer sentences. In the cybersecurity field, the length and complexity of entity descriptions cannot be overlooked. For instance, entity descriptions in the ATT&CK matrix are typically brief and concise, while those in CAPEC are more detailed and include comprehensive information about attack patterns. This complexity of long sentences indicates that relying solely on the word2vec model may not fully capture their rich semantic content. To address this issue, this paper designs a method combining word2vec and BiLSTM models, where word vectors initialized by word2vec are fed into a BiLSTM network. This combination allows for extracting higher-level features. Therefore, texts of different lengths and complexities can be accurately processed. When dealing with complex semantics in long sentences, it better captures and expresses the deeper meanings. This design optimizes the processing of short sentences and significantly enhances the understanding and expression ability for the complex semantics of longer sentences.

Figure 2 illustrates the framework of the proposed information encoding model, which represents the initial sequence of entity descriptions and denotes the output vector generated by concatenating the forward and backward outputs of the BiLSTM network along the dimension.

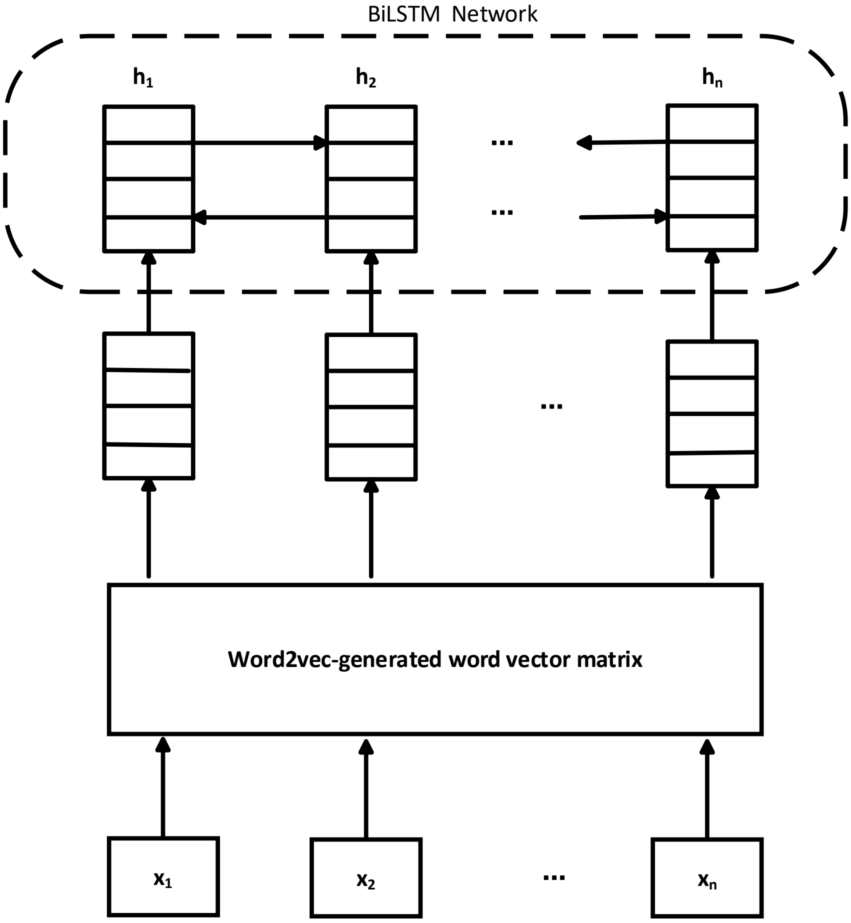


Figure 2. A text encoding model based on the BiLSTM network.

(2) Implementation of the vBiLSTM Model

To enhance the performance of the model, an initial step involves the pre-training of the word2vec model and the subsequent saving of the word vector file. These word vectors are then loaded based on the entity description vocabulary, and the word vector matrix is initialized, which allows for the prevention of overfitting. Finally, these word vectors are fed into the BiLSTM network to learn higher-level features.

Considering the quality and availability of pre-training corpora, although the English version of the Wikipedia corpus is extensive and covers multiple disciplinary fields, it may lack specific professional information in the cybersecurity field. To address this shortcoming, this paper extracts relevant data from the STIX 2.1 standard, the CVE vulnerability database, the CAPEC attack pattern library, and the CNNVD database. These data are integrated into the training corpus to enrich its domain-specific professional content. The process of data extraction is shown in Figure 3.

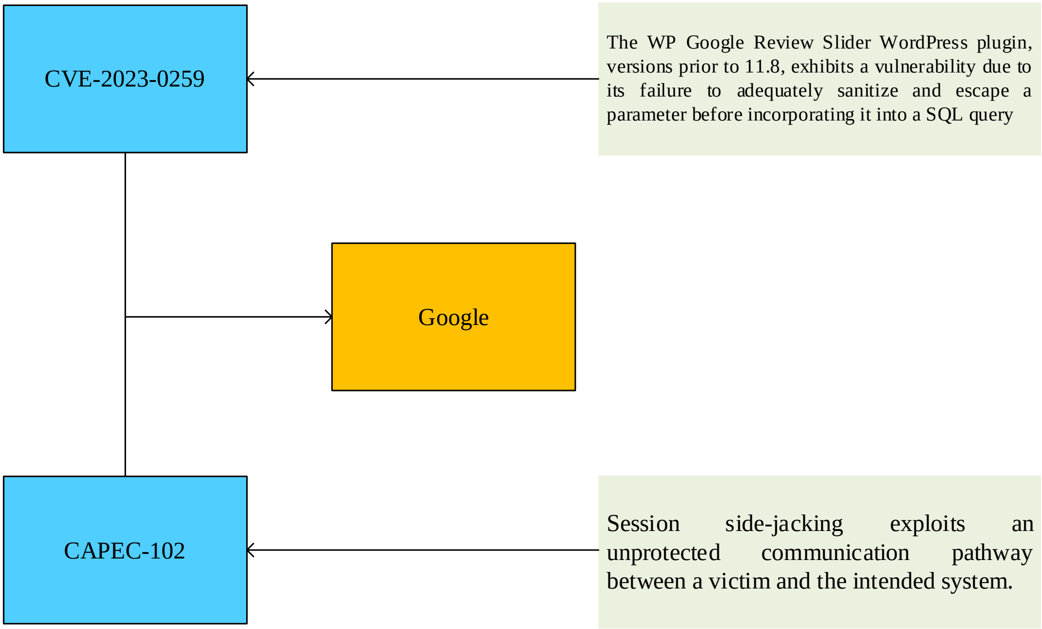

CVE-2023-0259
The WP Google Review Slider WordPress plugin, versions prior to 11.8, exhibits a vulnerability due to its failure to adequately sanitize and escape a parameter before incorporating it into a SQL query
Google
CAPEC-102
Session side-jacking exploits an unprotected communication pathway between a victim and the intended system.

Figure 3. English corpus in the CVE and CAPEC vulnerability library.

The 'word_count' function in Python is used to count all the tokens that appear in entity descriptions. Besides containing all the tokens that appeared in the entity description, the vocabulary also includes Out of Vocabulary (OOV) tokens, which refer to words, tags, and empty characters that are in the model input but not in the model vocabulary. In the establishment of a word vector index, the location information of tokens and their corresponding word vectors are saved in a JSON index file based on the pre-trained word vector file. The word vectors can then be retrieved based on the tokens in the vocabulary. If the retrieval is successful, the word vector is loaded. Otherwise, it is randomly initialized to obtain its corresponding matrix. The random initialization method aims to make the weight initialization more suitable for the ReLU activation function by considering its characteristics. The calculation is performed as:

$$\text{steddev} = \text{gain} \times \sqrt{\frac{2}{\text{fan\_in}}} \tag{1}$$

where fan_in represents the number of neurons in the input and output layers.

The random initialization method can randomly initialize weights from a normal distribution with a mean of 0. This helps solve the problem of gradient vanishing, which allows the ReLU activation functions to learn effective feature representations during network training, and this makes the distribution of the input and output as accurate as possible. Afterward, the word vector sequence is input into the BiLSTM to obtain the sentence representation.

Considering the use of batch processing (mini batch) in the entire process of encoding entity descriptor vectors in the BiLSTM network, the aim is to use the computational resources, accelerate the convergence, and enhance the performance of the model when training deep neural networks more effectively. In a mini batch, the different lengths of the entity description sequences become consistent after zero padding, and the length of the padded sequence depends on the longest one in the mini batch. Since zero padding blurs out the original semantics of the sequence, before inputting the word vector sequence into the network, it is necessary to sort it in descending order according to the sequence length and remove the filled zero vectors. In addition, each entity description should correspond one-to-one with the entity. That is, in a mini batch, the order of the entity descriptions is very important and, therefore, this order should be saved before sorting. After BiLSTM encoding, the original order is restored. Using batch processing and zero padding simultaneously can lead to a new issue, which may cause information loss problems. We conducted a controlled experiment when using batch processing and zero padding. The experimental results indicated that using this combination does indeed result in limited information loss issues; however, it significantly improves the efficiency of model training and enhances the model's fitting speed. Therefore, after weighing the pros and cons, we decided to proceed with this technique for further experiments.

BiLSTM network encoding adds a sorted word vector sequence. The original order of the entity description sequence is first saved. The sentences are sorted in descending order according to the length of the entity description. The 'pack_packed_sequence' method is used to remove the zero vectors filled in the sentences. They are then input into the BiLSTM network for encoding to obtain the output sequence. The 'pad_packed_sequence' method is used to fill the zero vectors in the output sequence. Finally, the order before sorting is restored to obtain the sentence representation.

Traditional BiLSTM mainly focuses on capturing sequence information in the text, using bidirectional processing of text data to understand the context of past and future inputs. This model is very suitable for tasks where understanding context is crucial, such as sentiment analysis of sentences or document classification. However, traditional BiLSTM cannot effectively understand proper nouns and long words in complex sentences, leading to difficulty in handling higher-level features within the sentences and resulting in suboptimal performance in some domains when using BiLSTM. On the other hand, by integrating word2vec-initialized word vectors into the BiLSTM network, vBiLSTM obtains the higher-level features of longer sentences, enhancing its ability to effectively process and encode longer and more complex textual sequences. By integrating word2vec-initialized vectors, vBiLSTM shows significant improvement over traditional BiLSTM, especially when dealing with longer and more intricate texts. Such enhancement is crucial in fields like network security because accurately encoding detailed and high-tech content directly impacts the effectiveness of knowledge graph completion and other AI-driven secure applications. These modifications not only enhance the model's performance but also broaden its applicability to more challenging text-processing tasks.

4. The KGC-N Model

This paper's choice of the KGC-N model over alternatives such as the GCN model or the GNN model is based on specific task requirements and the nature of the data. Firstly, KGC-N is specifically designed for handling knowledge graphs. Knowledge graphs are more than just collections of nodes and edges, they contain complex entity relations and rich semantic information. KGC-N is better equipped to handle these relationships and attributes, which might not be as straightforward or efficient in GCNs or general-purpose GNNs. Secondly, KGC-N can more effectively model various relationships among entities. In knowledge graphs, different types of edges represent different relationships, and KGC-N can assign different weights to these relationship types, which is crucial for certain tasks. Thirdly, data in knowledge graphs are often used for inference and relational mining. KGC-N may perform better on these tasks. Additionally, compared with alternatives like GCNs or GNNs, KGC-N can provide more rich and fine-grained entity representations. Overall, the decision to choose KGC-N over alternatives is typically based on the specific semantic and structural characteristics unique to knowledge graphs. This does not imply that KGC-N is always superior to GCNs or GNNs but rather that they offer more adaptive solutions when dealing with knowledge graph data.

The KGC-N model aims to fill gaps in open-source cybersecurity knowledge bases by creating and correlating information across various open-source cybersecurity knowledge bases using a security knowledge graph. Based on the concept of graph databases, KGC-N is built on the foundation of a cybersecurity knowledge graph. Its overall framework (Figure 4) includes the following key steps: (1) Initial Feature Learning: KGC-N uses knowledge graph embedding techniques to acquire the structural features of security entities. Since each security entity has descriptive textual information, KGC-N applies a pre-trained semantic model to learn the semantic features of the entities. (2) Neighborhood Feature Generation: Given the less pronounced structural features of entities in the security knowledge graph, KGC-N uses graph attention networks to learn the features of entities in the multi-hop neighborhood within the security knowledge graph. (3) Feature Fusion: To comprehensively understand the neighborhood information of security entities, KGC-N uses a feature fusion method that combines forward and reverse neighborhood features. (4) Information Completion: As the core objective of KGC-N, the system uses a decoder to decode and predict feature vectors. This allows for the completion of missing information in open-source cybersecurity knowledge bases. The model framework of KGC-N is shown in Figure 4.

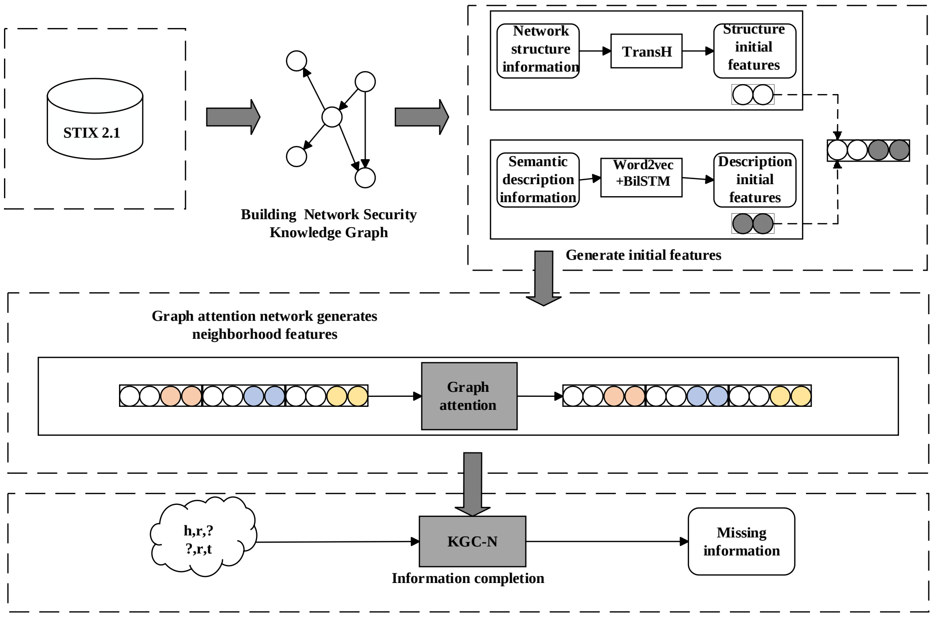

STIX 2.1
Building Network Security Knowledge Graph
Network structure information
TransH
Structure initial features
Semantic description information
Word2vec +BilSTM
Description initial features
Generate initial features
Graph attention network generates neighborhood features
Graph attention
h,r,?
?,r,t
KGC-N
Missing information
Information completion

Figure 4. Framework of KGC-N.

KGC-N obtains initial embedding vectors (E) for each security entity and relationship by learning their initial features in the secure knowledge graph through two dimensions (i.e., network structure and semantic description). $\mathrm{E} = \mathrm{E_s} \parallel \mathrm{E_d}$, where $\mathrm{E_s}$ represents the initial structural embedding vector, $\mathrm{E_d}$ represents the initial semantic embedding vector, and $\parallel$ denotes the concatenation operation. The size of E is equal to $\mathrm{d_1} + \mathrm{d_2}$.

In cybersecurity knowledge graphs, the relationships between security entities include sparse and complex node networks, which cannot be fully captured by relying solely on traditional initial embedding vector representation methods. It is challenging to accurately describe the roles and significance of entities in relation to each other using only initial embedding vectors. The core concept of the KGC-N model is the use of graph attention networks for learning the importance of the security entities within their neighborhood node tasks in the cybersecurity knowledge graph. The latter is reflected through attention values and used to generate richer and more distinct neighborhood feature embedding vectors. Consequently, the KGC-N model can better capture the complex network structural relationships between entities and reflect the unique roles and importance of different security entities in various triple relationships.

To capture information about the neighborhood, directed relationships are used to represent positive and negative neighborhood information. However, this representation is limited to one-to-one relationships within a one-hop neighborhood range. In theory, traversing in reverse any security entity and relationship in the security knowledge graph leads to corresponding multi-hop range inverse neighborhood information. However, in knowledge graphs, the forward traversal efficiency is often higher than that of the reverse traversal. Based on this inspiration, KGC-N exchanges the head and tail entities of triple relationships to generate the reverse graph $G' = \{E, R'\}$, where $R' = \{r_1{}', r'_2, \ldots, r'_m\}$ and $r'_j = \{t, r, h\}$. r'_j represents the inverse relationship between head entity h and tail entity t. G' is then forward traversed to capture bidirectional neighborhood information about security entities and relationships in network security knowledge graphs.

To learn neighborhood features, KGC-N employs an attention mechanism for learning the attention values between security entities and different neighbors in the cybersecurity knowledge graph, generating attention embedding vectors. In the graph attention network, the generation of attention embedding vectors is based on the attention values of the inverse relationship between h and t in a triple. These attention values are computed as follows:

$$T'_{(h,r,t)} = E_{(t)} \parallel E_{(r)} \parallel E_{(h)} \tag{2}$$

$$c'_{(h,r,t)} = W_1 T'_{(h,r,t)} \tag{3}$$

$$b'_{(h,r,t)} = \text{LeakyReLU}\left(W_2 c'_{(h,r,t)}\right) \tag{4}$$

$$a'_{(h,r,t)} = \text{softmax}_{(h,r)}\left(b'_{(h,r,t)}\right) = \frac{\exp\left(b'_{(h,r,t)}\right)}{\sum_{h' \in N'_t} \sum_{r' \in R'_{th'}} \exp\left(b'_{(h',r',t)}\right)} \tag{5}$$

where E stands for the initial embedding of entities or relations connected by initial structural embedding and semantic embedding, W_1 and W_2 represent two different linear transformation matrices, LeakReLU is used to obtain non-linear absolute attention values for triplets, softmax is used to calculate relative attention values for triplets, N'_t represents the set of entities within the k-hop neighborhood range of entity t in the reverse graph G', and R'_{th} represents the set of relations connecting entity t and entity h' in G'.

In the model, considering the structural characteristics of nodes in the cybersecurity knowledge graph, we set the value of k to 2. Firstly, in a knowledge graph, the neighborhood information of an entity (or node) extends beyond its directly connected entities (one-hop neighborhood) to include entities further connected through these direct neighbors (two-hop neighborhood). Setting k = 2 means the model considers not only the direct neighbors of a node (one-hop neighborhood) but also the neighbors of these neighbors (two-hop neighborhood). Secondly, the relationships between an entity and its two-hop neighborhood may provide crucial information about the entity's role and importance. By considering the two-hop neighborhood, the model can gain a deeper understanding of the complex network structure and dynamic relationships between entities. Additionally, while larger values of k can provide more comprehensive neighborhood information, they also significantly increase computational costs and may introduce noise. After multiple experiments, it was also confirmed that this view is correct. Setting the value of k to 2 strikes an effective balance between capturing sufficient neighborhood information and maintaining reasonable computational costs. Lastly, in many graph-structured problems, it has been demonstrated that considering two-hop neighborhood information is often sufficient to capture key structural features and dynamic relationships, particularly in resource-constrained and computationally efficient scenarios.

To fully consider the network structure of nodes, KGC-N sets the value of k to 2 for the calculation of the attention values for reverse neighborhoods.

More precisely, the three sets of relationships represented by initial embeddings in G and G' are input into different graph attention networks. In the network security knowledge graph, forward neighborhood feature vectors E_a and reverse neighborhood feature vectors $E_a{}'$ of secure entities and relationships, having a size of d', are generated.

By computing graph attention networks, KGC-N determines forward neighborhood feature vectors E_a and reverse neighborhood feature vectors $E_a{}'$ for each secure entity and relationship in the security knowledge graph. To fully learn the forward and reverse neighborhood information of secure entities, it uses a feature fusion method to merge the forward neighborhood feature vectors and reverse neighborhood feature vectors of secure entities and relationships. The average fusion is used. It is expressed as follows:

$$\mathrm{E}'' = \mathrm{avg}\left(\mathrm{E_a} + \mathrm{E'_a}\right) \tag{6}$$

For the above generated fused features, based on the encoder–decoder concept, KGC-N uses ConvKB as the decoder. The convolutional kernel of ConvKB is decent for learning the global feature information of triplets. In addition, it uses the ReLU activation function to activate the output values of the convolutional layer. The decoding process of triplets is represented by a scoring function $\mathrm{f_{(h,r,t)}}$, which is given by:

$$\mathrm{f_{(h,r,t)}} = \mathrm{W_3}\left(\|_{i=1}^{k}\ \mathrm{ReLU}\left(\left[\mathrm{E''_{(h)}}, \mathrm{E''_{(r)}}, \mathrm{E''_{(t)}}\right] * \Omega^{i}\right)\right) \tag{7}$$

where K represents the number of convolutional kernels in ConvKB, Ω^{i} represents the filter of the i-th convolutional kernel, $*$ denotes the convolution operation, $\|$ denotes the concatenation operation, and $\mathrm{W_3}$ represents a linear transformation matrix used to compute scores for triplets.

The decoder processes the merged feature embeddings of head entities, relations, and tail entities in triples, and it outputs scores for these triples. In theory, this decoding process allows for the real triples to obtain higher scores, while the false triples receive lower ones. In the information completion process, the decoder generates scores for all the potential completion triples to accurately distinguish between the real and false ones.

In the training process, based on real triples in the set of relations R, KGC-N synthesizes pseudo triples by replacing the head or tail entity of each triple, as shown in Equation (8). Moreover, during the training process, KGC-N adopts a soft margin as the loss function, which is presented in Equation (10).

$$\mathrm{R}' = \left\{\left(\mathrm{h}', \mathrm{r}, \mathrm{t}\right) \middle| \mathrm{h}' \in \mathrm{E}\backslash\mathrm{h}\right\} \cup \left\{\left(\mathrm{h}, \mathrm{r}, \mathrm{t}'\right) \middle| \mathrm{t}' \in \mathrm{E}\backslash\mathrm{t}\right\} \tag{8}$$

$$\mathrm{g_{(h,r,t)}} = \begin{cases} 1, & (\mathrm{h}, \mathrm{r}, \mathrm{t}) \in \mathrm{R} \\ -1, & (\mathrm{h}, \mathrm{r}, \mathrm{t}) \in \mathrm{R}' \end{cases} \tag{9}$$

$$\zeta = \sum_{(\mathrm{h,r,t}) \in \{\mathrm{R} \cup \mathrm{R}'\}} \log\left(1 + \exp\left(\mathrm{g_{(h,r,t)}} \cdot \mathrm{f_{(h,r,t)}}\right)\right) + \frac{\lambda}{2} \parallel \mathrm{W_3} \parallel_2^2 \tag{10}$$

Entity link prediction is a component of completing knowledge graphs. It is considered a sorting task, aiming to fill in missing entity parts in triples. For instance, in a triplet (h, r, t), given the head entity and relationship (h, r), the task consists of predicting the missing tail entity t. On the contrary, if the relationship and tail entity (r, t) are given, the head entity is predicted. The evaluation process is summarized as follows:

For each triplet in the test set, the head or tail entity in the triplet is replaced with other entities in the set to form a corrupted triplet. The rating function of the knowledge representation learning model is then used to rate these damaged triplets. Afterward, they are sorted according to ascending or descending rules. Finally, the correct ranking position of the triplets is recorded.

The entity link prediction task uses three main evaluation metrics (MR, MRR, and Hits@k) to evaluate the effectiveness of the designed KGC-N model for the network security knowledge graph. The mean ranking (MR) calculates the average ranking position of correct triples in test cases. A smaller value indicates that the model can rank the correct triples higher. The calculation process is shown in Equation (11). The mean reciprocal ranking (MRR) calculates the average of the reciprocal of the correct triplet ranking across all the test cases. A larger MRR value indicates that the model ranks the correct triplet higher, as shown in Equation (12). Hits@k denotes the average proportion of true triples ranked less than or equal to k in information completion. In other words, it represents the proportion of test cases with correct triples ranking in the top k. It demonstrates the coding model's quality, completeness, and practicality. Its calculation is given in Equation (13). The larger the Hits@k value, the better the encoding effect of the model. These metrics together provide a comprehensive view of model performance. MR provides the average ranking position, MRR considers the ranking of the first correct answer, and Hits@k tells us how often the correct answer appears in the top k results.

$$\mathrm{MR} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{rank}(\mathrm{r_i}) \tag{11}$$

$$\mathrm{MRR} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{\mathrm{rank}(\mathrm{r_i})} \tag{12}$$

$$\mathrm{Hits@k} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{indicator}(\mathrm{rank}(\mathrm{r_i}) \leq \mathrm{k}) \tag{13}$$

where N represents the number of triples, K represents the first k results, and $\mathrm{indicator}(\mathrm{rank}(\mathrm{r_i}))$ is an indicator function used for each test case (i). If for a certain entity $\mathrm{r_i}$, the predicted ranking rank ($\mathrm{r_i}$) is less than or equal to k, the function value is 1; otherwise, it is 0. In simple terms, if the correct answer is within the top k results predicted by the model, then the score is 1; otherwise, it is 0.

5. Simulation Experiments

The experimental configuration involved in this text is shown in Table 2.

Table 2. Experimental configuration (hyper) parameters.

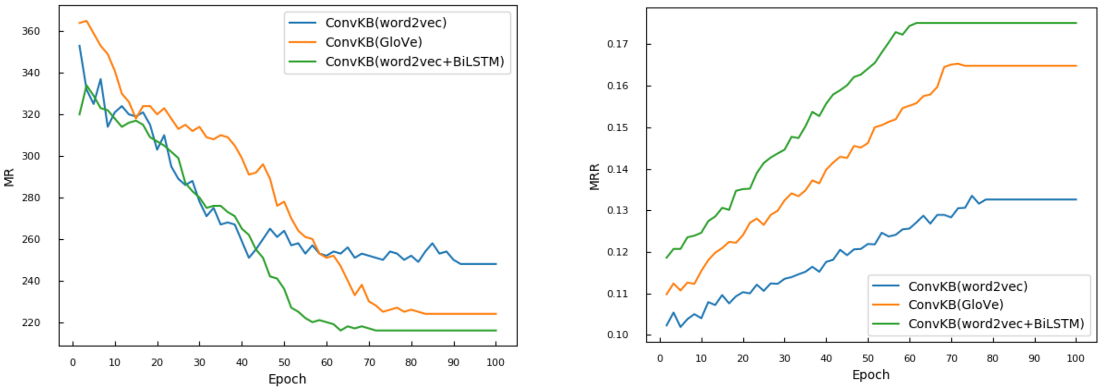
Variable Name	Value
Initial structural embedding vectors for security entities and relationships	100
Semantic description vectors	300
Learning rate	0.001
Decoder's number of epochs	100
Dropout	0.3

In the experiments, vector representations for the descriptions of cybersecurity domain entities are initially generated using word2vec, GloVe, and a word2vec model combined with BiLSTM. These generated vector representations are then used to initialize the entity embedding layer in the knowledge representation learning models. Neural network-based ConvE and ConvKB are used as knowledge representation learning models. Word vectors generated using the three different encoding methods are input to these models. The experiments aim to compare the entity prediction accuracies of these models. Because of the specific characteristics of the cybersecurity dataset (i.e., its high sparsity and the tendency of correct entities to rank lower in prediction sequences), the Hits@N evaluation metric is set in the range of 10–40, which suits the characteristics of the dataset.

In knowledge graph completion tasks, it is crucial to accurately identify and rank relevant entities and connections, and Mean Rank (MR) and Mean Reciprocal Rank (MRR) can effectively reflect the model's performance in this regard. MR measures the average ranking position of correct triples in the test cases. In knowledge graph completion tasks, a lower MR value indicates that the model can rank the correct triples higher more efficiently. MRR calculates the average reciprocal rank of correct triples in all test cases. A higher MRR value indicates better performance of the model in ranking correct triples higher. In summary, MR and MRR, as evaluation metrics, can reasonably reflect the performance of the KGC-N model in knowledge graph completion tasks in the cybersecurity domain. These metrics focus on how effectively the model identifies and ranks associated entities and relationships, which is crucial for knowledge graph completion tasks in the cybersecurity domain.

Given that FB15k and WN18 are widely recognized and used standard datasets in the field of knowledge graph research and contain multiple types of entities and rich relationship types, this makes them suitable for testing the capabilities of knowledge graph complementation models, especially when dealing with the complex entity relationships and inference tasks in this paper. At the same time, the use of these recognized benchmark datasets allows for validation of the model's generalization capabilities and a better understanding of the model's strengths and weaknesses. In addition, since many existing knowledge graph models have also been evaluated on these datasets, the use of FB15k and WN18 allows for a direct comparison of the performance of the new model with the existing models. Finally, since these datasets are maintained by professional organizations, the quality and reliability of the data are assured.

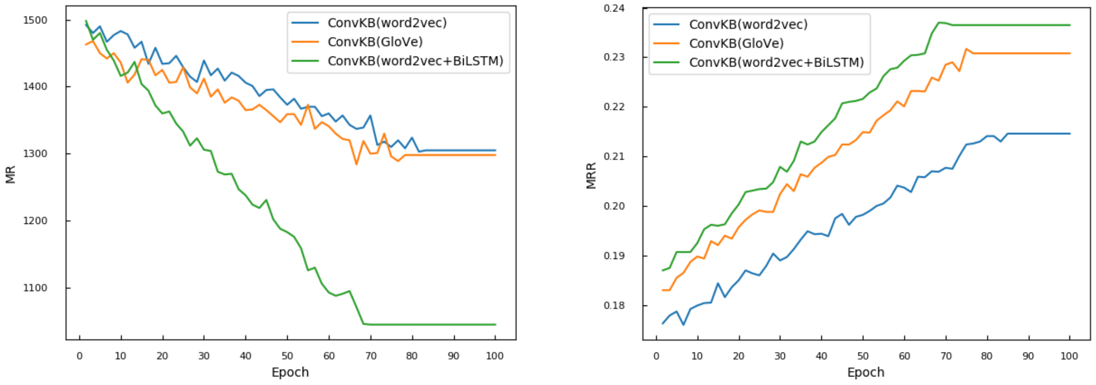
The performance (represented by MR and MRR) of the word2vec, GloVe, and vBiLSTM models in entity link prediction on open-domain general datasets (FB15k and WN18) is shown in Figure 5.



(**a**) MR (**b**) MRR

Figure 5. MR and MRR for entity link prediction using word2vec, GloVe, and vBiLSTM on open-domain datasets.

Figure 6 shows the results of entity link prediction (represented by the MR and MRR) obtained using the word2vec, GloVe, and vBiLSTM models on the cybersecurity dataset constructed based on STIX 2.1.



(**a**) MR (**b**) MRR

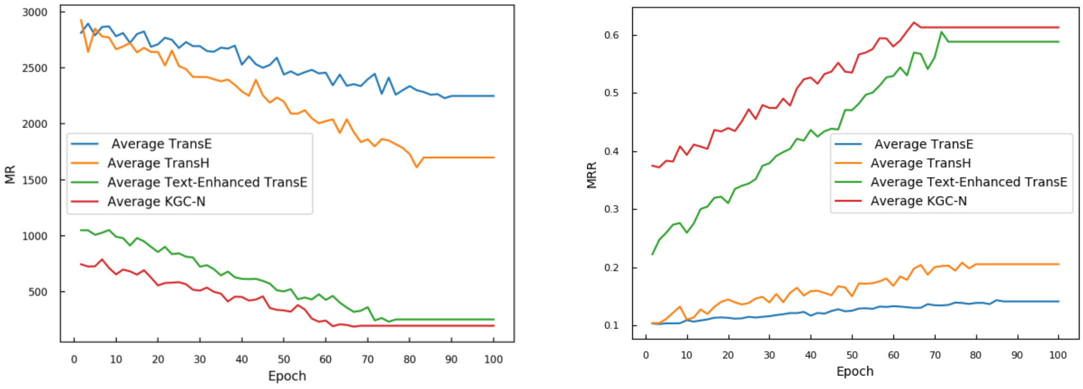
Figure 6. MR and MRR for entity link prediction using word2vec, GloVe, and vBiLSTM on the constructed cybersecurity dataset.

The vBiLSTM model outperforms the word2vec and GloVe models, and its accuracy rates are among the top 10 to 40 positions (Hits@10~40). This is attributed to the entity description vectors generated by the vBiLSTM encoding method, which contain richer and more effective semantic features. The word2vec+BiLSTM encoding method is efficient when processing entity descriptions related to cybersecurity. It effectively combines the word vector generation ability of word2vec with the sequential feature extraction advantages of BiLSTM, more comprehensively capturing and expressing the semantic content of entity descriptions. The richness and effectiveness of these semantic features contribute to an enhancement in the model performance in knowledge representation learning tasks, especially demonstrating significant advantages in the cybersecurity field.

The proposed method is also in contrast to the three traditional knowledge graph-based methods. The experiments related to this study and the comparative ones are all performed on the STIX 2.1 dataset integrated from four major knowledge bases. The methods involved in the comparison are summarized as follows:

In the studies on cybersecurity knowledge graphs, methods such as TransE, TransH, and Text-Enhanced GAT [16] are applied to explore and refine the structure and content of the knowledge graphs. The TransE method focuses on handling one-to-one entity relationships. It uses 100-dimensional word vector matrices as input and effectively models direct one-to-one relationships. The TransH method targets one-to-many relationship types. It uses the same word vector inputs. However, it is adapted to complex relational patterns through the introduction of hyperplane concepts. The Text-Enhanced GAT method emphasizes attention relations between two-hop forward neighborhoods of security entities, combining 100-dimensional structural with semantic vectors into a 200-dimensional composite vector to better understand and represent the model's complex neighborhood structures. In the comparative experiments, these models are trained through parameter tuning until convergence to ensure optimal performance in specific cybersecurity scenarios. The output consists of ranked lists generated after completing the information in the cybersecurity knowledge graph.

Figure 7 shows a comparison between the MR and MRR of the results obtained using KGC-N and those obtained using basic knowledge graph completion methods.



(**a**) MR (**b**) MRR

Figure 7. MR and MRR of the results obtained using KGC-N and basic knowledge graph completion methods.

It is evident that the performance of the KGC-N model in completing missing entities in the knowledge graph is higher than those of the basic methods. When focusing on the completion of head entities, KGC-N achieves an MR of 297 and an MRR of 0.5984, outperforming the basic methods. In terms of tail entity completion, the MR and MRR of KGC-N, respectively, reach 95 and 0.6269, substantially surpassing the basic methods. On average, KGC-N attains an MR of 196 and an MRR of 0.6127. In summary, KGC-N outperforms other basic methods in knowledge graph completion, particularly in the cybersecurity domain.

After completing the KGC-N and vBiLSTM experiments, we developed some ideas for our future work. Next, we will continue to focus on the development of this industry and actively compare and learn from new models in future research to continue the industry's progress. On the other hand, there are still some deficiencies in the current model. In the application of a knowledge graph completion model in the field of network security, its generalization ability has always been a weakness. In our follow-up work, we will introduce more novel models to address this shortcoming and use more evaluation metrics for comparison reference.

6. Conclusions

In the increasingly complex network environment, the situation of network security is becoming more and more severe. The knowledge graph of network security is an important part of the current field of network security. There are problems such as missing information, incompleteness, and sparse overall structure in the current knowledge graphs of the network security field. This article proposes a vBiLSTM network based on the word2vec word encoding method combined with bidirectional long short-term memory networks to solve the problem that traditional text encoding models provide limited information and insufficient reasoning ability. The proposed vBiLSTM can extract key complex information from sentences and preserve semantic information in entities as much as possible. After extracting the correct semantic information, for the knowledge graph completion problem, this article also proposes KGC-N as a knowledge graph completion model. This model uses feature vectors combined with graph structural information, perfectly integrates forward and reverse domain features, and ultimately utilizes a Transformer decoder for decoding prediction to complete efficient model completion tasks. By using high-quality open-domain datasets such as FB15k and WN18 for prediction testing, the proposed models in this article are shown to have lead levels in both MR (Mean Rank) and MRR (Mean Reciprocal Rank), proving that these two models provide perfect solutions to this task while contributing to the development of the entire field of network security knowledge graphs.

Author Contributions: Conceptualization, W.Z. and M.W.; methodology, G.H.; software, G.H.; validation, Y.F., X.T. and W.Z.; formal analysis, X.T.; investigation, M.W.; resources, W.Z.; data curation, X.T.; writing—original draft preparation, M.W.; writing—review and editing, W.Z.; visualization, G.H.; supervision, Y.F.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Guangxuan Project Foundation of Shenyang Ligong University under Grant SYLUGXXZ202204, the General Research Project of Education of Liaoning Province under Grant 2022IT205, and the China University Industry University Research Innovation Foundation under Grant 2021LDA06011.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yin, J.; Tang, M.J.; Cao, J.; You, M.; Wang, H.; Alazab, M. Knowledge-Driven Cybersecurity Intelligence: Software Vulnerability Coexploitation Behavior Discovery. *IEEE Trans. Ind. Inform.* **2023**, *19*, 5593–5601. [CrossRef]
2. Wang, L.; Qi, Y.; Bai, Y.; Sun, Z.; Li, D.; Li, X. MuKGB-CRS: Guarantee privacy and authenticity of cross-domain recommendation via multi-feature knowledge graph integrated blockchain. *Inf. Sci.* **2023**, *638*, 118915. [CrossRef]
3. Bhattarai, M.; Kharat, N.; Boureima, I.; Skau, E.; Nebgen, B.; Djidjev, H.; Rajopadhye, S.; Smith, J.P.; Alexandrov, B. Distributed non-negative RESCAL with automatic model selection for exascale data. *J. Parallel Distrib. Comput.* **2023**, *179*, 104709. [CrossRef]
4. Le, H.; Le, N.; Le, B. Knowledge graph embedding by relational rotation and complex convolution for link prediction. *Expert Syst. Appl.* **2023**, *214*, 119122. [CrossRef]

5. Fang, H.; Wang, Y.; Tian, Z.; Ye, Y. Learning knowledge graph embedding with a dual-attention embedding network. *Expert Syst. Appl.* **2023**, *212*, 118806. [CrossRef]
6. Shen, T.; Zhang, F.; Cheng, J. A comprehensive overview of knowledge graph completion. *Knowl-Based Syst.* **2022**, *255*, 109597. [CrossRef]
7. Xu, X.; Zhang, P.; He, Y.; Chao, C.; Yan, C. Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs. *arXiv* **2022**, arXiv:2208.00850.
8. Ma, X. Knowledge graph construction and application in geosciences: A review. *Comput. Geosci.* **2022**, *161*, 105082. [CrossRef]
9. Govindarajan, S.; Mustafa, M.A.; Kiyosov, S.; Duong, N.D.; Raju, M.N.; Gola, K.K. RETRACTED: An optimization-based feature extraction and machine learning techniques for named entity identification. *Optik* **2023**, *272*, 170348. [CrossRef]
10. Wang, X.; El-Gohary, N. Deep learning-based relation extraction and knowledge graph-based representation of construction safety requirements. *Autom. Constr.* **2023**, *147*, 104696. [CrossRef]
11. Li, Z.; Liu, X.; Wang, X.; Liu, P.; Shen, Y. TransO: A knowledge-driven representation learning method with ontology information constraints. *World Wide Web* **2023**, *26*, 297–319. [CrossRef]
12. Mohamed, H.A.; Pilutti, D.; James, S.; Bue, A.D.; Pelillo, M.; Vascon, S. Locality-aware subgraphs for inductive link prediction in knowledge graphs. *Pattern Recognit. Lett.* **2023**, *167*, 90–97. [CrossRef]
13. Huang, Z.; Guo, X.; Liu, Y.; Zhao, W.; Zhang, K. A smart conflict resolution model using multi-layer knowledge graph for conceptual design. *Adv. Eng. Inform.* **2023**, *55*, 101887. [CrossRef]
14. Lu, X.; Wang, L.; Jiang, Z.; Liu, S.; Lin, J. MRE: A translational knowledge graph completion model based on multiple relation embedding. *Math. Biosci. Eng.* **2023**, *20*, 5881–5900. [CrossRef] [PubMed]
15. Yuan, J.; Gao, N.; Xiang, J. TransGate: Knowledge Graph Embedding with Shared Gate Structure. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January 2019; pp. 3100–3107.
16. Nayyeri, M.; Vahdati, S.; Lehmann, J.; Yazdi, H.S. Soft Marginal TransE for Scholarly Knowledge Graph Completion. *arXiv* **2019**, arXiv:1904.12211.
17. Khan, F.; McCrae, J.P.; Gómez, F.J.M.; González, R.C.; Díaz-Vera, J.E. Some Considerations in the Construction of a Historical Language WordNet. In Proceedings of the 12th Global Wordnet Conference, San Sebastian, Spain, 23–27 January 2023; pp. 101–105.
18. Ebisu, T.; Ichise, R. TorusE: Knowledge Graph Embedding on a Lie Group. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 18–19. [CrossRef]
19. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *arXiv* **2019**, arXiv:1902.10197.
20. Wang, P.; Liu, J.; Zhong, X.; Zhou, S. A Cybersecurity Knowledge Graph Completion Method for Penetration Testing. *Electronics* **2023**, *12*, 1837. [CrossRef]
21. Li, Z.X.; Li, Y.J.; Liu, Y.W.; Liu, C.; Zhou, N.X. K-CTIAA: Automatic Analysis of Cyber Threat Intelligence Based on a Knowledge Graph. *Symmetry* **2023**, *15*, 337. [CrossRef]
22. Qi, Y.; Gu, Z.; Li, A.; Zhang, X.; Shafiq, M.; Mei, Y.; Lin, K. Cybersecurity knowledge graph enabled attack chain detection for cyber-physical systems. *Comput. Electr. Eng.* **2023**, *108*, 108658–108660. [CrossRef]
23. Piplai, A.; Mittal, S.; Joshi, A.; Finin, T.; Holt, J.; Zak, R. Creating Cybersecurity Knowledge Graphs from Malware After Action Reports. *IEEE Access* **2020**, *8*, 211691–211703. [CrossRef]
24. Kaiser, F.K.; Dardik, U.; Elitzur, A.; Zilberman, P.; Daniel, N.; Wiens, M. Attack Hypotheses Generation Based on Threat Intelligence Knowledge Graph. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4793–4809. [CrossRef]
25. Fang, Y.; Zhang, Y.; Huang, C. CyberEyes: Cybersecurity entity recognition model based on graph convolutional network. *Comput. J.* **2021**, *64*, 1215–1225. [CrossRef]
26. Sills, M.; Ranade, P.; Mittal, S. Cybersecurity threat intelligence augmentation and embedding improvement-a healthcare usecase. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020.
27. Jia, Y.; Qi, Y.; Shang, H.; Jiang, R.; Li, A. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering* **2018**, *4*, 53–60. [CrossRef]
28. Huaijun, S. *Research and Implementation of Web Security Knowledge Base Construction Technology Facing Vulnerability Database*; National University of Defense Technology: Changsha, China, 2018.
29. Wang, T.; Ai, Z.; Zhang, X. Construction technology of threat intelligence knowledge graph based on deep learning. *Comput. Mod.* **2018**, *12*, 21–26.
30. Peng, J.; Fang, Y.; Huang, C.; Liu, L.; Jiang, Z. Research on Named Entity Recognition in the Field of Information Security Based on Deep Active Learning. *J. Sichuan Univ. Nat. Sci. Ed.* **2019**, *56*, 457–462.

31. Zhang, R.; Liu, J.; He, X. Named Entity Recognition in the Security Vulnerability Domain Based on BLSTM-CRF Model. *J. Sichuan Univ. Nat. Sci. Ed.* **2019**, *56*, 469–475.
32. Xu, Y.; Tan, X.; Tong, X.; Zhang, W. A Robust Chinese Named Entity Recognition Method Based on Integrating Dual-Layer Features and CSBERT. *Appl. Sci.* **2024**, *14*, 1060. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.