informatics

MDPI

Article

CyberKG: Constructing a Cybersecurity Knowledge Graph Based on SecureBERT_Plus for CTI Reports

Binyong Li [1,2,3], Qiaoxi Yang [1,*], Chuang Deng [4] and Hua Pan [5]

[1] School of Cybersecurity (Xin Gu Industrial College), Chengdu University of Information Technology, Chengdu 610225, China; lby@cuit.edu.cn
[2] Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu 610225, China
[3] SUGON Industrial Control and Security Center, Chengdu 610225, China
[4] Center for Power Emergency Management, State Grid Sichuan Electric Power Corporation, Chengdu 610041, China; eedeng@126.com
[5] Guangxi Beitou Innovation Technology Investment Group Co., Ltd., Nanning 530029, China; panh@bgigc.com
* Correspondence: 3230809017@stu.cuit.edu.cn

Abstract

Cyberattacks, especially Advanced Persistent Threats (APTs), have become more complex. These evolving threats challenge traditional defense systems, which struggle to counter long-lasting and covert attacks. Cybersecurity Knowledge Graphs (CKGs), enabled through the integration of multi-source CTI, introduce novel approaches for proactive defense. However, building CKGs faces challenges such as unclear terminology, overlapping entity relationships in attack chains, and differences in CTI across sources. To tackle these challenges, we propose the CyberKG framework, which improves entity recognition and relation extraction using a SecureBERT_Plus-BiLSTM-Attention-CRF joint architecture. Semantic features are captured using a domain-adapted SecureBERT_Plus model, while temporal dependencies are modeled through BiLSTM. Attention mechanisms highlight key cross-sentence relationships, while CRF incorporates ATT&CK rule constraints. Hierarchical clustering (HAC), based on contextual embeddings, facilitates dynamic entity disambiguation and semantic fusion. Experimental evaluations on the DNRTI and MalwareDB datasets demonstrate strong performance in extraction accuracy, entity normalization, and the resolution of overlapping relations. The constructed knowledge graph supports APT tracking, attack-chain provenance, proactive defense prediction.

Keywords: cybersecurity; knowledge graphs; cyber threat intelligence; named entity recognition; relation extraction

Academic Editor: Jiang Bian

Received: 5 August 2025
Revised: 17 September 2025
Accepted: 18 September 2025
Published: 22 September 2025

Citation: Li, B.; Yang, Q.; Deng, C.; Pan, H. CyberKG: Constructing a Cybersecurity Knowledge Graph Based on SecureBERT_Plus for CTI Reports. *Informatics* **2025**, *12*, 100. https://doi.org/10.3390/informatics12030100

1. Introduction

With the rapid development of network technologies, cyberattacks have become more complex and diverse. Attack methods like zero-day vulnerabilities and Advanced Persistent Threats (APTs) breach traditional defenses and significantly reduce cybersecurity protection effectiveness. According to CNNVD statistics, the annual growth rate of high-risk vulnerabilities has exceeded 40% in recent years, and the design of attack chains has become increasingly sophisticated. This results in two core limitations for rule- and signature-based defense mechanisms: first, they cannot keep up with the rapid changes in vulnerabilities, and second, they struggle to detect and respond to multi-stage attack chains [1,2].

In response, Cyber Threat Intelligence (CTI) has become a key tool in modern network defense. CTI includes important information about network attacks, vulnerabilities, malware, and APT activities, providing security experts with essential insights for threat response and prediction [3]. However, CTI data is primarily unstructured text, facing challenges such as complex semantics, non-unified formats, and data noise, which severely limit the effective utilization of intelligence and, in turn, affect the timeliness and effectiveness of threat response [4].

Knowledge Graphs (KGs) help convert unstructured CTI data into structured information, improving the efficiency of intelligence analysis. While KGs have been used in healthcare and finance, there are still challenges in cybersecurity, such as unclear terminology, limited ability to capture semantics, and difficulties in extracting relationships across documents.

To address these issues, this paper proposes a Cybersecurity Knowledge Graph construction framework based on SecureBERT_Plus—CyberKG. CyberKG's primary contributions are threefold. First, we introduce SecureBERT_Plus, a domain-adaptive encoder tailored to cybersecurity terminology and long-context semantics, improving representation of technical entities. Second, we design an end-to-end joint extraction architecture that combines SecureBERT_Plus, BiLSTM, attention mechanisms, and an ATT&CK-constrained CRF decoder to better model cross-sentence dependencies and reduce pipeline error propagation. Third, CyberKG employs context-aware HAC together with dynamic terminology masking for robust entity fusion and disambiguation across heterogeneous CTI sources.

2. Related Work

2.1. Cyber Threat Intelligence

CTI analyzes APT reports, vulnerability databases (CVE/CNNVD), and attack chain models (e.g., Kill Chain) to identify attackers' TTPs, IoCs, and targets. The core challenge of CTI data analysis arises from its multi-source and heterogeneous nature. Research has evolved from rule-based methods to three main approaches that combine deep learning and knowledge graph technologies.

Rule-based methods rely on manually extracted patterns to identify threat behaviors. While they offer strong interpretability, they struggle with complex sentences and intricate semantics [5]. Satvat et al.'s using BERT(Bidirectional Encoder Representations from Transformers), although progress has been made in basic feature extraction, generic pretrained models still suffer from a semantic gap when applied to specific domains [6].

Knowledge graph technology has introduced a new approach for structured analysis. It enhances the semantic representation ability of intelligence data and supports tasks such as attack chain analysis, APT organization attribution, and threat prediction. For example, the Open-CyKG framework proposed by Sarhan et al. [7] builds threat graphs using open information extraction, which improves retrieval accuracy. However, its ability to generalize to emerging attack scenarios remains limited.

2.2. Information Extraction

With the rapid growth in the scale of CTI data in recent years, efficiently extracting valuable information from APT reports, vulnerability databases, and attack chain models has become a key research direction in cybersecurity. However, due to the unstructured nature of CTI data, extracting entities and their relationships directly from text presents significant challenges. Traditional rule-based methods face limitations such as insufficient performance and poor adaptability when processing these complex texts. However, advancements in deep learning techniques in recent years have provided new approaches for the structured processing of CTI data.

In information extraction tasks, mainstream methods have evolved along two paths: pipeline extraction and joint extraction. The former breaks the task into multiple independent submodules (such as named entity recognition and relation extraction) to improve interpretability, while the latter enhances overall efficiency and avoids error propagation. The following sections will discuss the characteristics, technological evolution, and practical applications of these two approaches in CTI data processing.

2.2.1. Pipeline Approach

The pipeline approach divides the task into sequential sub-tasks, such as entity recognition and relation extraction. Its main advantage is that each module can be independently optimized. Three major technological trends have been identified in this field.

First, deep architectural innovations focused on fundamental model design. Kim's team proposed a BiLSTM-CRF model that captured the semantic features of cybersecurity terminology by incorporating character-level and word-level features. However, it struggled to model long-range dependencies [8]. Ren et al. developed the CSKG4APT platform, which integrated STIX/CYBOX standards for structured APT scenario modeling. It also used deep learning and NLP techniques for entity extraction, enabling an end-to-end workflow from extraction to application [9].

Second, a semantic enhancement pathway has been proposed, emphasizing the integration of domain-specific knowledge. The CTI View system, developed by Zhou et al., combines regular expressions with deep learning to improve IoC and TTP extraction accuracy. However, it struggles to parse complex semantic relations [10].

Third, breakthroughs in pre-trained models have driven advancements in domain adaptation. Srivastava et al. demonstrated that fine-tuned BERT embeddings significantly improve named entity recognition, forming a foundation for model optimization in specific domains [11]. To improve entity recognition accuracy, Wang et al. created the APTNER dataset, which includes key entities and relations, providing a foundation for domain adaptation in cybersecurity intelligence extraction [12].

However, these advances are not enough to resolve two key limitations in pipeline approaches. First, cascading error propagation, where downstream relation extraction is affected by entity recognition failures, as seen in Zhou et al.'s hybrid architecture [10]. Second, semantic fragmentation, which prevents coherent modeling of multi-stage attack scenarios. For example, Ren et al. demonstrated the challenge of maintaining APT attack-chain coherence across modular boundaries [9].

2.2.2. Joint Extraction Approach

Joint extraction paradigms address cascading failures by unifying named entity recognition (NER) and relation extraction (RE) tasks. This integration mitigates sequential error propagation while enhancing semantic consistency across multi-stage threat modeling scenarios.

Joint extraction methods, which integrate named entity recognition (NER) and relation extraction (RE), have been shown to significantly outperform pipeline-based approaches in mitigating error propagation and enhancing semantic coherence. The technical evolution of these methods primarily follows three directions:

In terms of foundational architecture, Bekoulis et al. proposed a BiLSTM-CRF framework enhanced via adversarial training, where small-scale perturbations were injected to improve the model's generalization capability [13]. Zhang et al. introduced the JF-ER framework for joint optimization of entity linking and relation extraction. However, its performance was limited by general knowledge base paradigms [14].

In the area of graph-based modeling, Zhao et al. developed a heterogeneous graph convolutional network (HINTI) for graph-based modeling. It improves information extraction

in complex cybersecurity texts by modeling global entity-relation dependencies [15]. Guo et al. proposed a fusion framework that incorporated an improved Levenshtein algorithm to effectively avoid error propagation inherent in pipeline methods; however, its real-time processing capability on large-scale data remains to be optimized [16]. Wang et al. introduced TIRECO, which employed a dependency graph pruning strategy and the concept of sentence sets to enhance the efficiency of cross-sentence relation extraction. Nevertheless, further validation is needed regarding its scalability and generalization capabilities [17].

Finally, pretrained enhancement approaches have broken through the bottlenecks in semantic modeling. Zuo et al. developed an end-to-end model for joint entity and relation extraction using sequence labeling to enable parallel task processing, achieving better performance than traditional pipeline methods [18]. Ahmed et al. proposed the RoBERTa-BiGRU-CRF model, combining the global representation power of pretrained language models with the local pattern recognition strengths of sequential models. Dynamic gradient modulation alleviates optimization conflicts, improving both entity recognition and relation classification accuracy [19].

Joint methods still exhibit unresolved limitations. Cross-sentence modeling deficiencies hinder overlapping APT relation parsing, as observed in prior frameworks. Domain adaptation constraints also persist when processing unstructured data. CyberKG addresses these gaps through SecureBERT_Plus-powered contextual disambiguation and attention-based attack-chain decoding. This approach demonstrates potential for improving cross-modal fusion accuracy and reducing pipeline errors.

2.3. Summary of Related Research Methods

This paper, based on the relevant work in the aforementioned literature regarding the extraction of named entities and their relationships from unstructured reports, outlines and compares the technical characteristics and application scenarios of various methods. Table 1 presents the main methods and datasets used in these studies for entity-relation extraction in cybersecurity, highlighting the distinctions between each approach.

Table 1. Comparison of Research Methods for Cybersecurity Entity-Relation Extraction.

Ref.	Technique	Dataset	Entities	Relations	Joint Extraction	Knowledge Graph
Niakanlahiji et al. (2018) [5]	SECCMiner	APTnotes	✔	✘	✘	✘
Satvat et al. (2021) [6]	BERT-BiLSTM with SRL	Microsoft, TrendMicro, DARPA	✔	✔	✘	✔
Sarhan and Spruit (2021) [7]	OIE	Microsoft, MalwareDB	✔	✔	✘	✔
Kim et al. (2020) [8]	BiLSTM-CRF (BOC features)	CTI custom dataset (498,000 tags)	✔	✘	✘	✘
Ren et al. (2023) [9]	BERT-BiLSTM-GRU-CRF + Diamond Model	1041 APT reports	✔	✔	✘	✘
Zhou et al. (2022) [10]	BERT-GRU-BiLSTM-CRF	120 English APT reports	✔	✘	✘	✘
Srivastava et al. (2023) [11]	BERT with BiLSTM	Microsoft, Metasploit	✔	✘	✘	✘
Wang et al. (2022) [12]	BERT-BiLSTM-CRF	APTNER	✔	✘	✘	✘

Table 1. *Cont.*

Ref.	Technique	Dataset	Entities	Relations	Joint Extraction	Knowledge Graph
Bekoulis et al. (2019) [13]	BiLSTM-CRF with Adversarial Training	ACE04, CoNLL04, ADE	✔	✔	✔	✘
Zhang et al. (2020) [14]	JF-ER with LSTM, PCNN	NYT-10 variants	✔	✔	✔	✘
Zhao et al. (2022) [15]	HINTI	Custom	✔	✔	✔	✔
Guo et al. (2023) [16]	BERT with BiGRU	OSINT Data	✔	✔	✔	✔
Wang et al. (2020) [17]	C-GCN	Public	✔	✔	✔	✔
Zuo et al. (2022) [18]	BERT-BiLSTM-CRF	Custom	✔	✔	✔	✘
Ahmed et al. (2023) [19]	RoBERTa-BiGRU-CR	Custom (100 annotated CTI reports)	✔	✔	✔	✔
CyberKG	SecureBERT_Plus-BiLSTM-Attention-CRF joint extraction	DNRTI, MalwareDB	✔	✔	✔	✔

Synthesizing these comparative analyses highlights methodological refinements in CyberKG for cybersecurity knowledge graphs. Prior frameworks such as Open-CyKG rely heavily on rule-based or open information extraction, which often fragments cross-sentence attack-chain semantics. CSKG4APT introduces deep models but follows a modular pipeline architecture, leading to error propagation and limited entity disambiguation. In contrast, CyberKG integrates SecureBERT_Plus with BiLSTM layers, attention mechanisms, and CRF decoding in a joint extraction architecture, reducing cascading errors while explicitly modeling cross-sentence dependencies. Dynamic terminology masking further improves recognition of domain-specific entities, and context-aware HAC enables more precise entity fusion across heterogeneous CTI sources. Collectively, these refinements provide more coherent and resilient modeling pathways for complex threat scenarios.

2.4. Transformer-Based Semantic Modeling for Cybersecurity

Transformer, with its multi-head scaled dot-product self-attention (multiplicative) and bidirectional context encoding capabilities, has become a cornerstone technology in natural language processing. BERT, as a typical example based on Transformer's bidirectional encoder representations, achieves knowledge transfer through the pre-training and fine-tuning paradigm, establishing a new path for semantic understanding tasks.

2.4.1. Domain Adaptability Analysis of Generic BERT Models

The general BERT model has demonstrated its potential by being pre-trained on large-scale general corpora such as Wikipedia and BookCorpus. For example, Zuo et al. proposed a joint extraction model using general BERT to extract entities and relations in cyber threat intelligence. This improved entity recognition accuracy in cybersecurity and demonstrated BERT's feasibility for cyber threat analysis [18]. Similarly, Ren et al. applied the BERT model to named entity recognition tasks, successfully enhancing the effectiveness of APT attribution [9].

Although the general BERT model has achieved strong performance across a range of tasks, certain limitations remain when applied to cybersecurity scenarios. First, the pre-training corpora of general BERT lack sufficient coverage of cybersecurity terminology, resulting in shallow understanding of domain-specific terms and leading to a semantic gap. Second, the standard 512-token input window of Transformer-based models limits the modeling of complete attack chains. These chains often span over a thousand tokens in APT reports. This constraint limits the model's ability to capture long-range dependencies and may result in reasoning errors and logical inconsistencies. Therefore, despite its widespread application in various NLP tasks, the performance of general BERT remains inadequate for threat intelligence analysis in the cybersecurity domain.

2.4.2. BERT Models for the Cybersecurity Domain

To address the persistent challenges of domain-specific terminology and long-range contextual in generic BERT architectures for cybersecurity applications, specialized BERT variants have been developed. This section examines the capabilities and limitations of popular domain-adapted models, highlighting domain-specific challenges that led to the innovations in SecureBERT_Plus.

(1) ATTACK-BERT

ATTACK-BERT specializes in the MITRE ATT&CK framework. The model excels in attack behavior vectorization and TTP classification tasks. However, it has two main limitations: poor performance on general cybersecurity NLP tasks (such as named entity recognition) and a heavy reliance on attack description databases [20].

(2) CySecBERT

CySecBERT is optimized for malware analysis and attack classification tasks. Trained on over 4.3 million security texts, it leads in malware classification and short-text analysis, but has significant shortcomings in long-text modeling [21].

(3) CTI-BERT

CTI-BERT, developed by IBM Research, is a BERT model designed for cybersecurity threat intelligence. It was trained from scratch on 1.2 billion security-related tokens using a masked language model (MLM). With a vocabulary of 50,000 tokens and a sequence length of 256, it excels in cybersecurity text classification. However, it is mainly optimized for structured text and offers limited support for unstructured data [22].

(4) SecureBERT

SecureBERT is a cybersecurity-specific BERT variant optimized based on the RoBERTa architecture. This model removes the NSP task and employs larger batch training to enhance long-text modeling capabilities. During pre-training, it incorporates a wide range of cybersecurity terminology (e.g., malware families, APT organizations). It performs excellently in named entity recognition and relation extraction tasks but still exhibits limitations in multi-relationship and long-distance dependency modeling [23].

(5) SecureBERT_Plus

SecureBERT_Plus is an enhanced version of SecureBERT. It scales the training corpus by 8 times and improves MLM task performance by 9%. This model enhances prediction accuracy for cybersecurity terms and context understanding, optimizing multi-task learning for stronger generalization and transferability across tasks such as NER, text classification, and question answering. It is considered one of the most advanced cybersecurity BERT models currently available.

Table 2's comparison shows SecureBERT_Plus' enhanced adaptability to diverse cybersecurity NLP requirements. This model integrates key innovations like DeBERTa-enhanced position encoding and adaptive multi-task coordination. As a result, it covers a wider range of tasks than specialized models such as ATTACK-BERT (limited to TTP classification) and CTI-BERT (optimized for structured data). SecureBERT_Plus also excels at processing cross-sentence relationships, addressing SecureBERT's limitations. These advantages, along with its ability to handle unstructured data, align with CyberKG's operational needs. This makes it an ideal choice for complex threat intelligence scenarios.

Table 2. Comparison of Core Characteristics of BERT Variants in the Cybersecurity Domain.

Model	Task	Key Advantages	Adaptation Boundaries
ATTACK-BERT	TTP Classification	Integrates MITRE ATT&CK framework; semantic alignment	Limited to TTP-related tasks; Requires extensive attack databases
CySecBERT	Malware Detection	Optimized for short-text classification; effective log data parsing	Constrained in long-context modeling; Single-task focused
CTI-BERT	CTI Analysis	Trained on rich datasets; effective for structured text	Limited unstructured data support; No relation extraction capability
SecureBERT	NER + RE	Tailored for cybersecurity domain; enhanced entity recognition	Challenges in cross-sentence relationships; Multi-relationship modeling limitations
SecureBERT_Plus	Multi-task Learning	Multi-task learning (MTL) optimization; enhanced generalization with DeBERTa	Higher computational requirements; Increased inference latency

3. Construction of Cybersecurity Knowledge Graphs

To address the complexity of unstructured threat intelligence in the cybersecurity domain, this study proposes a novel cybersecurity knowledge graph construction framework—CyberKG. The framework primarily tackles core issues such as semantic parsing, joint entity-relation extraction, and knowledge fusion optimization. By integrating domain-specific pre-trained models and knowledge graph optimization techniques, it aims to provide precise support for downstream tasks.

3.1. Overall Framework Design

The CyberKG framework achieves knowledge transformation of threat intelligence through the collaborative processing of three modules. As shown in Figure 1, the architecture consists of three major modules: data preprocessing, semantic information extraction, and knowledge graph construction and optimization. Together, these modules form a complete pipeline from raw data to decision support.

(1) CTI Report Collection: The data preprocessing module standardizes and structures cybersecurity data from multiple sources, such as APT attack reports, vulnerability databases, and malware logs. This module ensures high-quality input data for semantic extraction by performing tasks like format conversion, deduplication, and noise filtering.

(2) Knowledge Extraction: The semantic information extraction module is the core component of the framework, responsible for extracting valuable entities and relations from the standardized data. This module utilizes SecureBERT_Plus for deep semantic encoding, captures temporal dependencies within the context through BiLSTM, and jointly decodes using attention mechanisms and CRF to ensure efficient and

accurate identification of named entities and relations. The entity-relation-entity triples generated by this module serve as the foundation for subsequent knowledge graph construction.

(3) Knowledge Graph Construction: The knowledge graph construction and optimization module is responsible for integrating the extracted entities and relations to build an initial cybersecurity knowledge graph. It enhances the graph's structure and accuracy through optimization methods such as refinement fusion, semantic disambiguation, and hierarchical clustering. The goal of this module is to generate a high-quality knowledge graph that supports downstream tasks, including threat analysis, threat hunting, and defense strategy recommendation. Additionally, the constructed knowledge graph can be used for threat search, threat analysis, threat hunting, and defense strategy recommendation. Specifically, threat search enables precise retrieval of relevant intelligence, threat analysis reveals attack paths and associations, threat hunting proactively detects potential threats, and defense strategy recommendation provides targeted defense advice based on historical attack patterns, contributing to intelligent security defense.

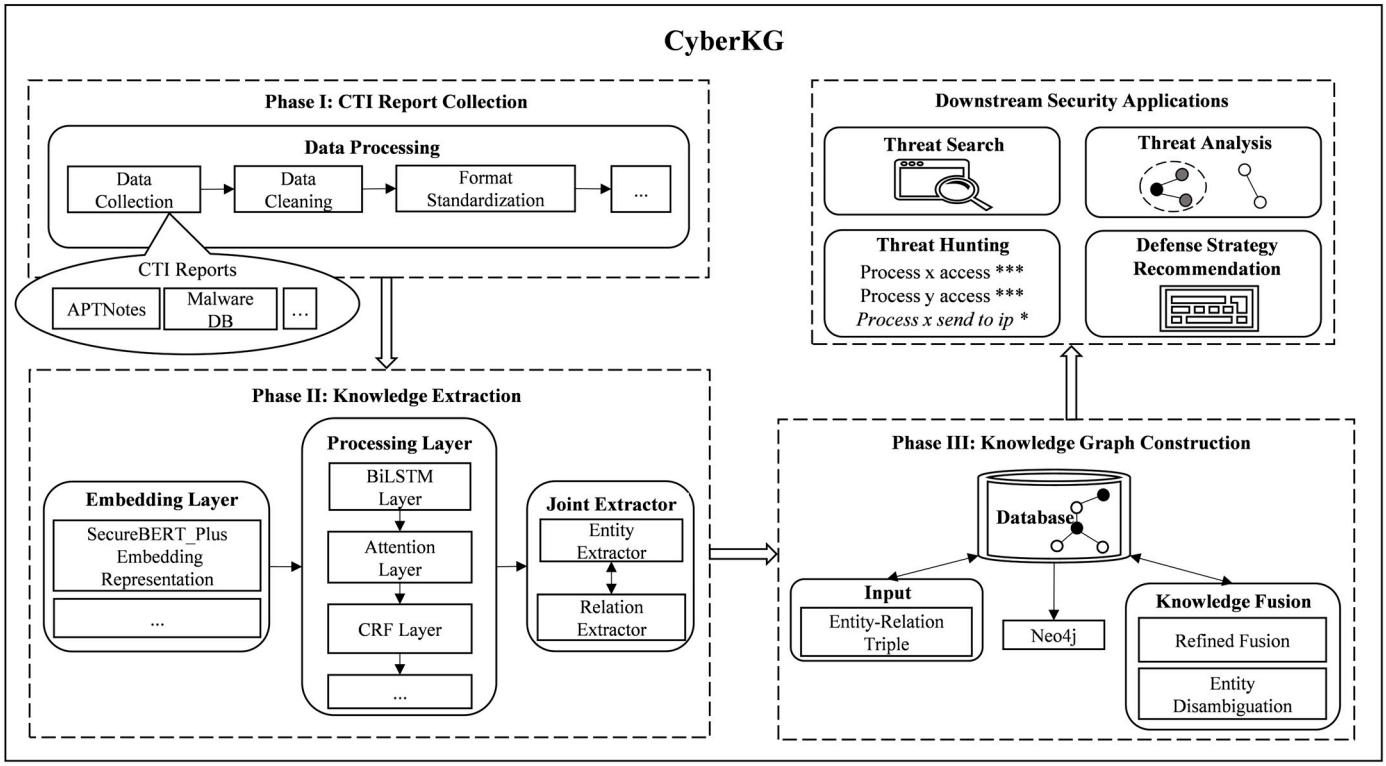

Figure 1. Architecture of CyberKG (Arrows Indicate Data Flow). Asterisks (*) denote anonymized parameters (e.g., IP addresses, file paths) or classes of behavioral patterns.

The design goal of the CyberKG framework is not only to construct a high-quality cybersecurity knowledge graph but also to enhance the intelligence level of threat intelligence analysis. CyberKG optimizes the modules collaboratively to model the full process, from raw unstructured data to structured knowledge graphs. It provides precise, interpretable knowledge support for APT attack tracing, threat awareness, and intelligence attribution.

3.2. Model Construction

3.2.1. Multi-Layer Joint Information Extraction Model

Figure 2 illustrates the architecture of the multi-layer joint information extraction model for network threat intelligence proposed in this paper. Through a cascading structure,

the model integrates SecureBERT_Plus, BiLSTM, Attention, and CRF modules to enable end-to-end attack chain analysis. This approach overcomes the semantic gaps caused by the disjointed nature of entity recognition and relation extraction tasks in traditional pipeline methods, thereby enhancing the performance of information extraction from cybersecurity corpora.

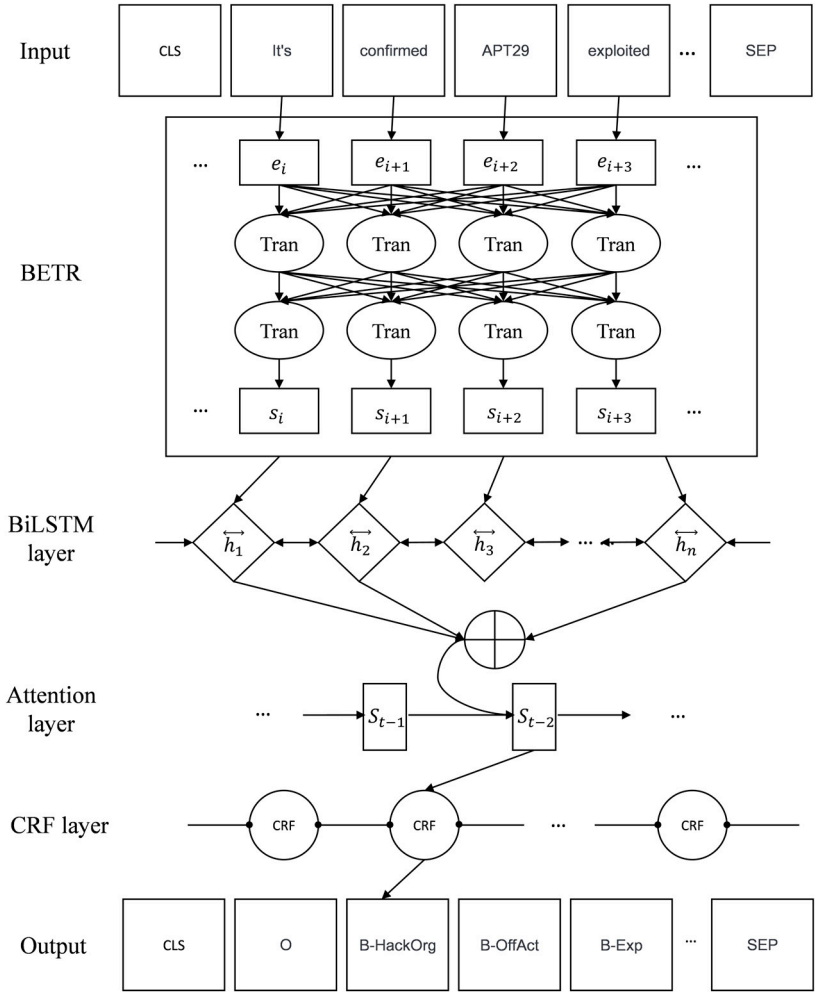

Figure 2. Multi-layer Information Extraction Architecture.

As shown in Figure 2, the proposed model consists of three key stages:

(1) Input Layer and Dynamic Semantic Encoding

The input sentence "It's confirmed APT29 exploited CVE-2023-1234 via phishing emails" is tokenized into a sequence of sub word-level tokens and encoded into contextual embeddings $\{e_1, e_2, \ldots, e_8\}$ using SecureBERT_Plus. Notably, the embedding e_3 for "APT29" is enhanced with attacker-specific features via MITRE ATT&CK semantic mapping, while the embedding e_4 "exploited" is disambiguated from its generic sense ("developed") to the OffAct category, representing an attack action. This dynamic contextualization outperforms static word embeddings in cybersecurity-specific semantic discrimination.

(2) Cross-Sequence Temporal Modeling

The BiLSTM layer further captures temporal dependencies and contextual logic by integrating bidirectional contextual information. For example, in the phrase "APT29 exploited CVE-2023-1234", forward propagation $\overrightarrow{h}_1 \rightarrow \overrightarrow{h}_n$ conveys attack intent from subject to target (APT29 $\rightarrow$ exploited $\longrightarrow$ CVE-2023-1234), while backward propagation

links post-attack tool descriptions (phishing emails ⟶ via ⟶ CVE-2023-1234), forming a complete semantic chain of attack method and target. This enables the construction of a semantic topology such as ⟨Tool, Exploitation Method, Vulnerability⟩.

(3) Attack Chain-Aware Decoding

The additive attention layer dynamically weights tokens to resolve long-range dependencies and accentuate pivotal semantic relationships. For instance, it learns to intensify associations between threat actors and actions (e.g., "APT29" → "exploited"), directly boosting relation extraction precision. Crucially, this mechanism captures vulnerability exploitation patterns spanning multiple sentences (e.g., sequential exploits: CVE-2023-1234 → CVE-2023-4567), thereby explicitly formalizing chained attack behaviors.

The CRF layer models the transition probabilities between labels, leveraging a transition matrix to block illegal tag sequences and optimize label prediction. For example, while a SoftMax layer might mistakenly output [B-Tool, O] for "phishing emails," the CRF layer prevents the invalid B-Tool ⟶ O transition and enforces a valid sequence [B-Tool, I-Tool]. This also ensures logical coherence in the attack chain, such as enforcing that OffAct tags follow Exp or Tool tags.

Ultimately, the model outputs the label sequence [O, B-HackOrg, B-OffAct, B-Exp, O, B-Tool, I-Tool], and extracts the triples ⟨APT29, Exploits, CVE-2023-1234⟩ and ⟨APT29, Uses, Phishing Emails⟩. Compared with traditional pipeline-based methods, the four-stage integration—Encoding ⟶ Propagation ⟶ Focusing ⟶ Constraint—forms a closed-loop process that significantly improves both accuracy and overall performance.

3.2.2. Dynamic Semantic Encoding

SecureBERT_Plus, as the core BERT layer implementation, overcomes the limitations of general pre-trained models in cybersecurity, including poor terminology coverage, semantic ambiguity, and difficulty modeling long texts. Compared with other BERT variants, SecureBERT_Plus introduces several domain-specific innovations. Through vocabulary expansion, it injects cybersecurity terms such as "zero-day" and "lateral movement" to avoid semantic fragmentation caused by sub word splitting. Additionally, noise augmentation with Gaussian perturbations enhances the model's ability to distinguish polysemous terms—for example, mapping "virus" accurately to the malware domain in threat intelligence scenarios [16].

In terms of pre-training tasks, SecureBERT_Plus prioritizes masking entity-level tokens relevant to the security domain (e.g., CVE identifiers and attack techniques), offering more targeted focus on key cybersecurity entities compared to the generic BERT masking strategy. The input text is first processed into a tokenized sequence by the tokenizer, with selected words masked. Following the RoBERTa design, the NSP (Next Sentence Prediction) task is removed, and the [CLS] token is used to aggregate global semantics. Each token's embedding is composed of token embedding, positional embedding, and segment embedding, which are summed to form the final input representation.

The Transformer encoder in SecureBERT_Plus employs a self-attention mechanism (as shown in Figure 3) to effectively capture long-range dependencies in CTI datasets. For example, when processing the sentence "APT29 exploited the CVE-2023-1234 vulnerability and subsequently deployed Cobalt Strike via phishing emails," the self-attention mechanism is able to assign higher attention weights to the semantic relationship between CVE-2023-1234 and Cobalt Strike. Specifically, the self-attention mechanism is computed as shown in Equation (1):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where Q, K, and V are the query, key, and value matrices derived from the input token embeddings. The relevance between tokens is calculated via the dot product of Q and K, and the scores are normalized through a softmax function to generate context-sensitive representations [4]. Equation (1) follows the scaled dot-product (multiplicative) self-attention formulation: queries/keys/values are obtained by linear projections; the dot product is computed via MatMul and scaled by the per-head dimension d_k; SoftMax then produces context-aware weights.

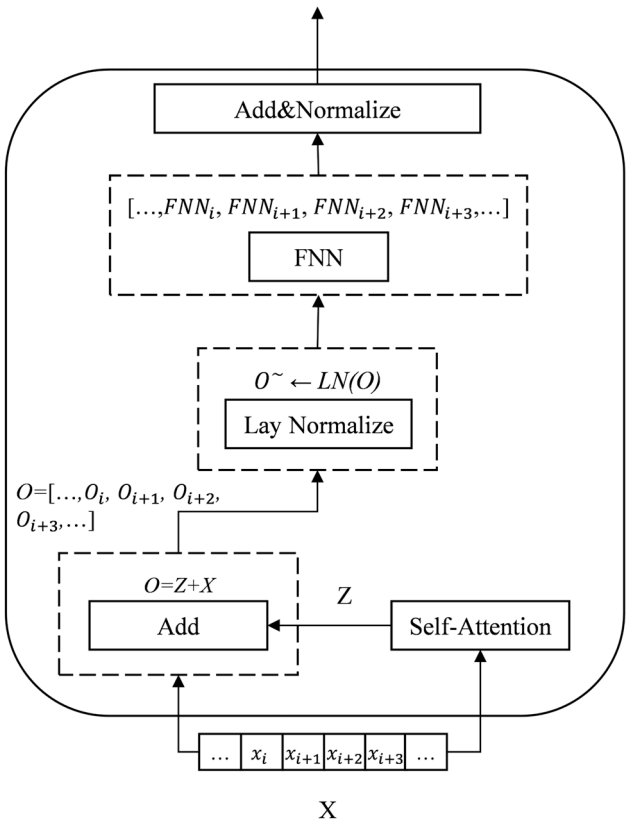

Figure 3. Transformer encoding unit. This module applies multiplicative (scaled dot-product) self-attention in the encoder.

SecureBERT_Plus adopts a multi-head scaled dot-product self-attention (as shown in Figure 4), which employs a parallel subspace modeling strategy to enhance the capture of semantic information while mitigating overfitting in complex tasks. The computation process is as follows:

Assuming the number of heads is n, n sets of Q, K, and V are initialized and projected using separate parameter matrices W_i^Q, W_i^K, and W_i^V, respectively. These are then passed into the self-attention module to compute attention values, and the resulting outputs are concatenated and fed into a fully connected layer. The multi-head scaled dot-product self-attention is defined as

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_n) \tag{2}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{3}$$

Equations (2) and (3) illustrate the computation process of the multi-head scaled dot-product self-attention, and Figure 4 shows its architectural implementation [19]. These equations correspond to the concatenate-then-project step of multi-head attention, preserving the multiplicative form in the encoder.

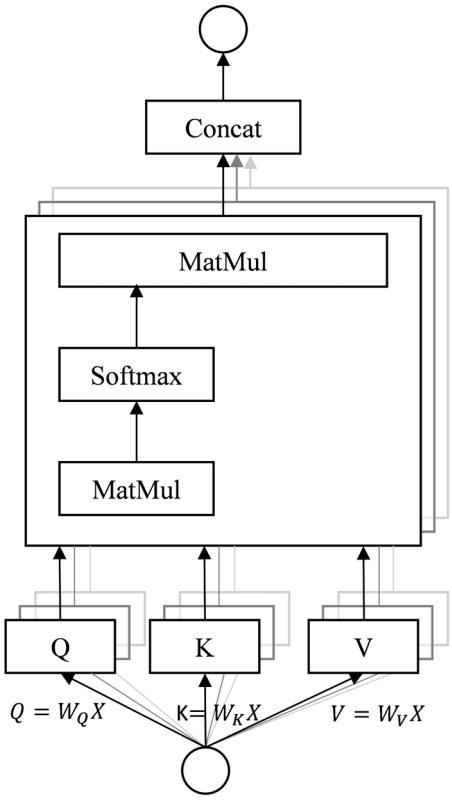

Figure 4. Structural diagram of multi-head scaled dot-product self-attention. Adapted from Vaswani et al. (2017). MatMul denotes matrix multiplication. This module uses multi-head scaled dot-product self-attention, serving the role of global context modeling in SecureBERT_Plus.

To preserve deep feature integrity, the model employs residual connections. After applying multi-layer normalization, the intermediate output Z is transformed toward a standard normal distribution, followed by further processing via a feed-forward neural network. The computation is as follows [24]:

$$O^{\sim} = LN(O) \tag{4}$$

$$O' = FNN(O^{\sim}) \tag{5}$$

$$Final\ Output = LN(O' + O^{\sim}) \tag{6}$$

The final output sequence is defined as

$$O = [\ldots, O_i, O_{i+1}, O_{i+2}, \ldots] \tag{7}$$

Here, Z represents the output from the self-attention mechanism, and X denotes the input sequence. The layer normalization (LN) includes learnable scale and shift parameters γ and β, and the GeLU activation function is used in the feed-forward layer. This design ensures that critical information related to the attack chain is preserved throughout the encoding process and is seamlessly passed to the subsequent BiLSTM layer for further modeling.

3.2.3. Attack Chain Temporal Modeling

Attack events in CTI (Cyber Threat Intelligence) texts typically exhibit multi-stage, chain-like dependencies. Traditional RNNs struggle to model such long-range dependencies due to the vanishing gradient problem. In contrast, BiLSTM enhances the model's ability to capture structured representations of attack chains by modeling sequential depen-

dencies, thereby improving global modeling for both attack chain reconstruction and entity semantic disambiguation.

Given an input sequence $X = \{x_1, x_2, \ldots, x_n\}$, the semantic representations generated by SecureBERT_Plus are

$$X = \{x_1, x_2, \ldots, x_n\}, h_t^b \in \mathbb{R}^{d_b} \tag{8}$$

where $h_t^b \in \mathbb{R}^{d_b}$ denotes the embedding of the t-th token, and d_b is the embedding dimension. The BiLSTM layer computes the hidden states via forward and backward LSTMs as follows [25]:

$$h_t^{\rightarrow} = LSTM_{forward}\left(h_t^b, h_{t-1}^{\rightarrow}\right) \tag{9}$$

$$h_t^{\leftarrow} = LSTM_{backward}\left(h_t^b, h_{t+1}^{\leftarrow}\right) \tag{10}$$

The final BiLSTM output is obtained by concatenating the forward and backward hidden states:

$$h_t^{b_i} = [h_t^{\rightarrow}; h_t^{\leftarrow}] \tag{11}$$

This bidirectional interaction mechanism enables the model to capture global temporal relationships within attack chains, ensuring that dependencies across different stages of an attack are comprehensively modeled. This paper deconstructs the attack lifecycle through a three-stage process—Initial Access → Exploitation → Lateral Movement—as illustrated in Figure 5.

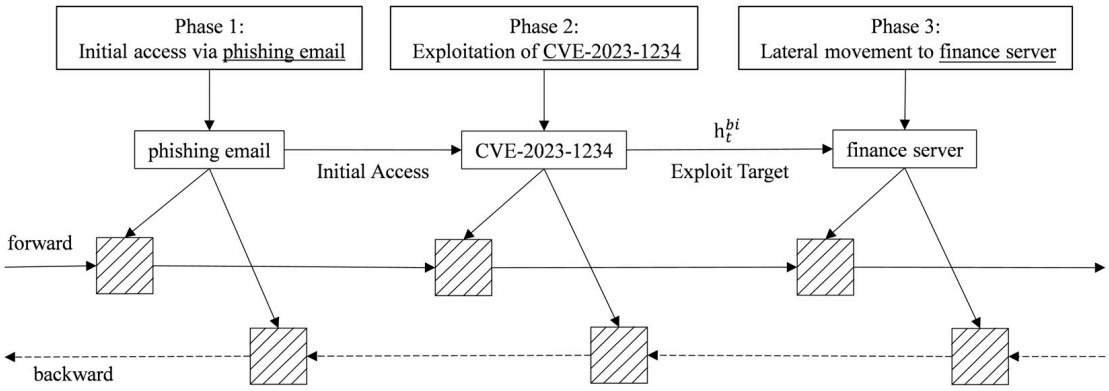

Figure 5. Phase Analysis of the Attack Process.

(1) Forward propagation (Phase 1 → Phase 3):

The model encodes the attack action "lateral movement" while retaining "phishing email" as the initial entry point, thereby constructing the attack chain: phishing email → CVE-2023-1234 → finance server.

(2) Backward propagation (Phase 3 ← Phase 1):

During backward propagation, the model captures the dependency between "finance server" and "CVE-2023-1234", revealing the ultimate target of the vulnerability exploitation. The hidden states $h_t^{b_i}$ generated by the BiLSTM are then passed to the subsequent attention mechanism, further enhancing the aggregation of critical information across attack phases.

In parallel, BiLSTM also effectively addresses the issue of entity semantic disambiguation. In CTI texts, entities often rely on contextual information to be correctly identified. For example, in the sentence "The phishing email contained a malicious attachment targeting the financial department in Chengdu and was sent by Josh," the token "Chengdu" may be incorrectly classified as a geographical entity when viewed in isolation, whereas it actually denotes an organizational attribute of the "financial department." Similarly, "Josh" might

be misidentified as a common personal name, but when contextualized with "phishing email," it can be correctly inferred as the attacker. Through bidirectional modeling, BiLSTM fuses semantic features of entities with surrounding context, thereby producing more accurate recognition results [26]. The CRF imposes ATT&CK-aware transition constraints; the attention-enhanced features reduce illegal transitions and error propagation.

3.2.4. Decoding and Perception of the Attack Chain

To achieve precise localization and relational parsing of key threat elements within lengthy CTI texts, this module introduces a dual mechanism consisting of relation-aware attention and tactic-constrained decoding, enabling accurate attack chain extraction.

First, based on the BiLSTM hidden states $\{h_1, h_2, \ldots, h_n\}$, a relation-aware attention mechanism is constructed, as defined by the following equations [27]:

$$a_i = \frac{\exp(W_a h_i)}{\sum_{j=1}^{n} \exp(W_a h_j)} \tag{12}$$

$$S_t = \sum_{i=1}^{n} \alpha_i h_i \tag{13}$$

Here, α_i is the normalized weight of token i, computed from the dot-product score $W_a h_i$ (with trainable W_a). This multiplicative attention emphasizes key entities and strengthens cross-sentence vulnerability links. The fusion of temporal and attention features is given by

$$O = [S_t; h_t] \tag{14}$$

This fused representation is fed into the tactic-constrained decoding layer, whose CRF scoring function is defined as

$$Score(X, y) = \sum_{i=0}^{n} T[y_i, y_{i+1}] + \sum_{i=1}^{n} P_{i,y_i} \tag{15}$$

Here, the transition matrix T incorporates ATT&CK-based constraints: invalid transitions (e.g., OffAct → Victim) are assigned a value of $-\infty$, while valid paths (e.g., Off Act → Exp) retain log transition probabilities $log\mathcal{P}(next|prev)$. This mechanism corrects labeling errors (e.g., converting [B-ATTACK, I-TOOL] into [B-ATTACK, B-TOOL]), and finally, the optimal tag sequence is obtained using Viterbi decoding as defined in

$$y^* = \arg\max_{y} Score_i(X, y) \tag{16}$$

3.3. Extraction of Entities and Relationships

3.3.1. Ontology Construction

This study constructs a domain-specific ontology tailored to the characteristics of the DNRTI dataset, based on the Unified Cyber Ontology (UCO 2.0) framework proposed by Syed et al. (2016) [28], with significant extensions from Mouiche and Saad's TiKG ontology [29]. To address DNRTI dataset characteristics, we enrich the ontology architecture with four novel entity categories, establishing formal constraints as detailed in Table 3.

(1) Attack Agents: This category includes hacker organizations (HackOrg) and advanced persistent threat groups (APTGroup). These entities represent the initiators of cyberattacks and help trace the origin of malicious activities.
(2) Attack Vectors: This category includes tools (Tool) and malware samples (SamFile). These entities describe the means used during the attack process, such as malicious software and exploitation utilities.

(3) Defensive Entities: This category includes security teams (SecTeam) and protected organizations (Org). These entities represent the defensive side and support the identification and tracking of key participants and protected targets within defense operations.
(4) Spatiotemporal Attributes: This category includes attack time (Time) and geographic region (Area). These entities are used to capture the temporal and spatial dimensions of attacks, which aids in analyzing the distribution and evolution of cyber events.

Table 3. Definition and Constraints of Ontology Relationships in the Cybersecurity Knowledge Graph.

Relationship	Domain	Range	Note
analyses	SecTeam	SamFile	Security team analyzes sample files
associatedWith	HackOrg	HackOrg	Collaborative relationships between hacker organizations
discovers	SecTeam	HackOrg	Security team discovers hacker organizations
discoveredBy	HackOrg	SecTeam	Hacker organization discovered by security team
hasAttackTime	HackOrg/OffAct/Way	Time	Attribute for attack time
hasCharacteristics	HackOrg/OffAct/Tool/ SamFile	Features	Description of entity features
locatedAt	Org	Area	Geographical location of the organization
monitors	SecTeam	Org/Area/Tool/Exp	Objects monitored by the security team
monitoredBy	Org/Area/Tool/Exp	SecTeam	Objects monitored by security teams
motivates	Purp	HackOrg/OffAct/Exp/Way	Motivations driving actions
motivatedBy	HackOrg/OffAct/Exp/Way	Purp	Motivation behind actions
uses	HackOrg/OffAct/Tool/ SamFile	Tool/OffAct/Exp/ SamFile/Way	Tools or methods used by the entity
usedBy	Features/OffAct Tool/SamFile	HackOrg/OffAct/Exp/ Way/Tool/SamFile	Users of tools or methods
targets	HackOrg/OffAct Tool/SamFile	Area/Org/SecTeam	Attack targeting
targetedBy	Area/Org/SecTeam	HackOrg/OffAct/Exp/ Way/Tool/SamFile	Entities being targeted by attacks

The incorporation of these additional entity types enriches the original cybersecurity ontology, enabling a more comprehensive representation of the complex relationships and dynamic behaviors observed in the DNRTI dataset.

The ontology defines 15 semantic relations to model logical associations between entities within threat intelligence. The domain and range constraints of these relations strictly follow cybersecurity operational patterns and are formalized using description logic, ensuring that each relation expresses its semantics precisely. The definitions of these relations are summarized in the following table.

To better understand the interrelationships among various security entities and support in-depth analysis of cybersecurity threat intelligence, this study constructs a domain-specific ontology based on the DNRTI dataset. Building upon this foundation, Figure 6 illustrates a schematic representation of the relationships among different types of security entities. The figure provides an intuitive overview of how entities such as hacker organiza-

tions, vulnerability exploitation, and attack tools are interconnected within the ontology framework [30].

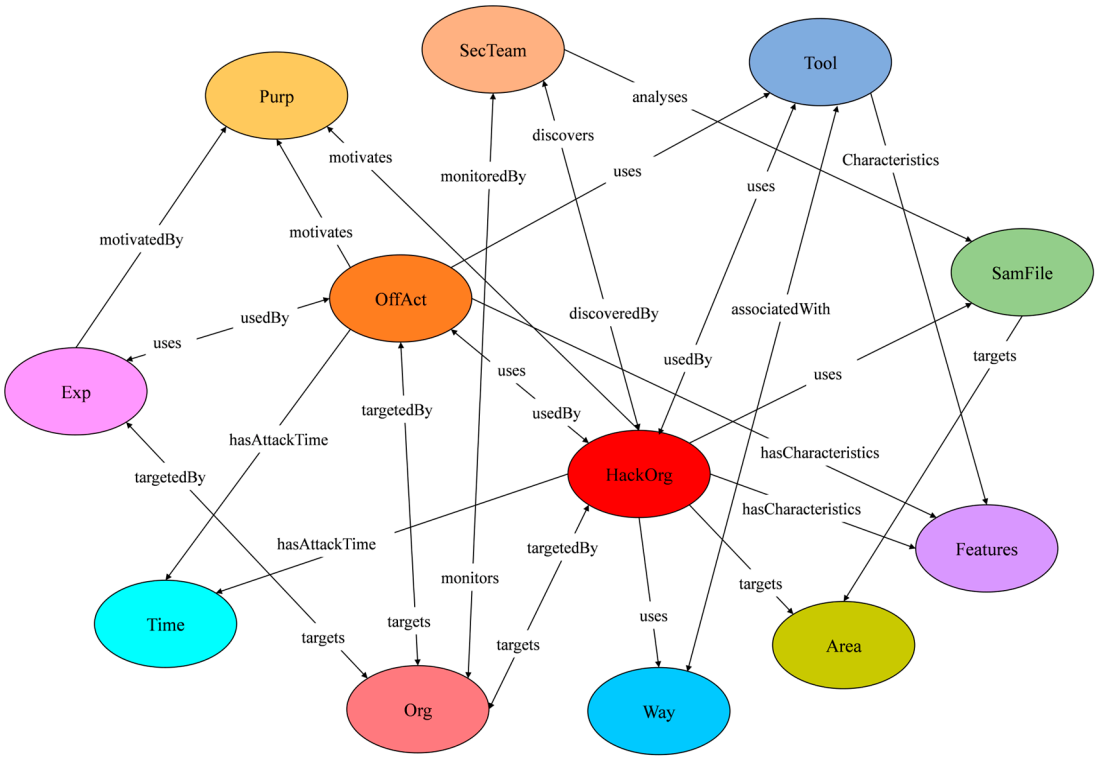

Figure 6. Topological Modeling of the Cybersecurity Ontology.

3.3.2. Relationship Extraction

CTI texts contain complex attack events and inter-entity relationships involving various relational patterns such as attacker–target–attack method. To construct a comprehensive cyber threat intelligence knowledge graph (CyberKG), this study adopts a dynamic context-aware entity pairing strategy. By integrating bidirectional dependency parsing and semantic role labeling (SRL), the output from named entity recognition is mapped into standardized triples (e_i, r, e_j). This improves both entity disambiguation and relation extraction.

The implementation details are outlined in Algorithm 1. For clarity, we specify below the notions of contextual proximity used in the matching step and the assignment of the relation type r. Intra-sentence pairs are processed by normalizing SRL/dependency cues to the ontology relation set R, while cross-sentence pairs are filtered using the task-specific schema and ATT&CK-constrained CRF decoding. This mapping specifies the relation type r by normalizing SRL and dependency cues for intra-sentence pairs and selecting from the task-specific schema for cross-sentence pairs.

The core idea lies in the joint use of bidirectional dependency analysis and semantic role labeling to enhance entity disambiguation and relation inference, particularly under complex attack chain scenarios.

Algorithm 1 Entity-Relation Triple Matching

Input:
x: Sequence with tags
$E_{gen} = \{\ \}$: general entities
$E_{ove} = \{\ \}$: overlapping relational entities
Output:
T: Set of (e_i, r, e_j) triples
Extract all complete entities through the boundary tag;
Assign entities to sets:
$E_{gen} = e_i$ #General entity
$E_{ove} = \{\ \}$ # Overlapping entity relationship entities
While $E_{gen} \neq \{\ \}$ **Do**
For each entity $e_i \in E_{gen}$ **Do**
Find pairable entities $e_j \in E_{gen} \cup E_{ove}$:
Search direction: forward & backward
Match criterion: contextual proximity
Select the nearest valid e_j based on contextual proximity.
Form knowledge triple (e_i, r, e_j).
Remove e_i from E_{gen}
End For
End While
Return all generated triples

In the relation extraction process, this study integrates Semantic Dependency Analysis (SDA) with entity pairing rules to achieve precise identification of inter-entity relationships. The extraction process supports two core patterns:

(1) Intra-sentence general relation triple extraction

Based on Semantic Role Labeling (SRL), core predicates are identified, and subject-object relationships are determined through dependency analysis. As shown in Figure 7, for the sentence "Lazarus Group operates from North Korea," the entities "Lazarus Group" (a hacker organization) and "North Korea" (a geographic location) are detected. SRL identifies the core predicate "operates" along with the modifying phrase "from North Korea," which is then mapped to the normalized relation "belong-to." As a result, the triple [Lazarus Group, operates-from, North Korea] is generated. This mapping specifies the relation type **r** by normalizing SRL/dependency cues to the ontology relation set **R** for intra-sentence pairs. Each entity participates in a pairing only once, and the extraction for this sentence is then complete.

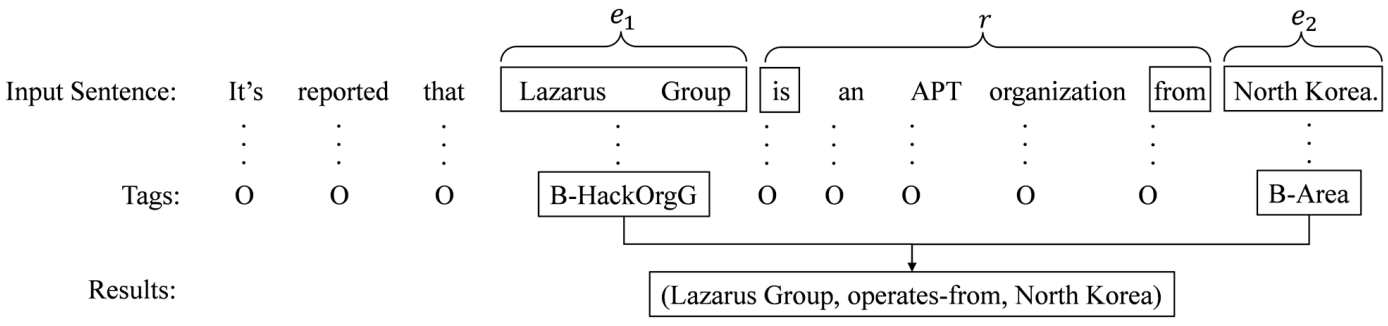

Figure 7. Extraction of General Relationship Triplets.

(2) Cross-sentence overlapping relation triple extraction

Complex attack patterns commonly found in CTI often manifest as cross-sentence attack chains and multi-entity relational structures. As illustrated in Figure 8, consider the following typical attack sequence:

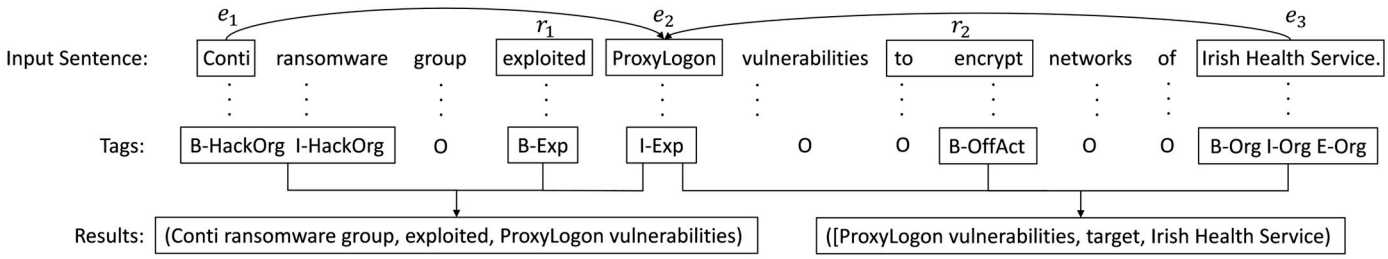

Figure 8. Extraction of Overlapping Relationship Triplets.

Phase 1: Initial access via phishing email sent by "Josh".

Phase 2: Exploitation of CVE-2023-1234 to deploy a backdoor.

Phase 3: Lateral movement to the finance server in the Chengdu branch.

Traditional intra-sentence relation extraction methods are inadequate for handling the temporal logic that spans across phases (e.g., Phase 1 → Phase 3). To address this, the proposed approach employs a sliding-window-based relation search strategy. If two entities appear in adjacent sentences, the model computes a contextual dependency score using BiLSTM outputs. If the score exceeds a predefined threshold, a direct relation between the entities is inferred [31]. In Algorithm 1, contextual proximity is thus defined as a local-window criterion: candidates are considered within the same or adjacent sentences; a BiLSTM-based contextual dependency score is computed for each pair, pairs are retained if the score $\geq \tau$, and intra-sentence ties are resolved by shorter token distance. For cross-sentence pairs that satisfy the proximity rule, the relation type **r** is selected from the task-specific schema (e.g., uses, exploits, targets) based on SRL/dependency patterns, and is further filtered by the ATT&CK-constrained CRF decoding to ensure consistency with domain tactics.

In addition, CTI texts often involve compound attack patterns, where a single entity may establish different relations with multiple objects, resulting in overlapping relation triples. Figure 8 illustrates an example of this process. The entities "Conti ransomware group," "ProxyLogon vulnerabilities," and "Irish Health Service" are extracted from the text. A forward pairing from "Conti ransomware group" to "ProxyLogon vulnerabilities" generates the first triple: [Conti ransomware group, exploited, ProxyLogon vulnerabilities].To prevent redundant pairing, "ProxyLogon vulnerabilities" is skipped as an active subject in subsequent pairing. Instead, a backward search is performed from "Irish Health Service," producing the second triple: [ProxyLogon vulnerabilities, target, Irish Health Service] thus completing the extraction task.

3.4. Construction and Optimization of the Knowledge Graph

Based on the structured triples generated by the previously described SecureBERT_Plus joint extraction model (as illustrated in Figure 9), several core challenges in the threat intelligence network are revealed. These include entity redundancy, such as the coexistence of "data" and "the data" as separate nodes; and relation ambiguity, exemplified by the overlapping semantics of relations like "SEND TO" and "WAS_USED_TO" within data transmission paths [32].

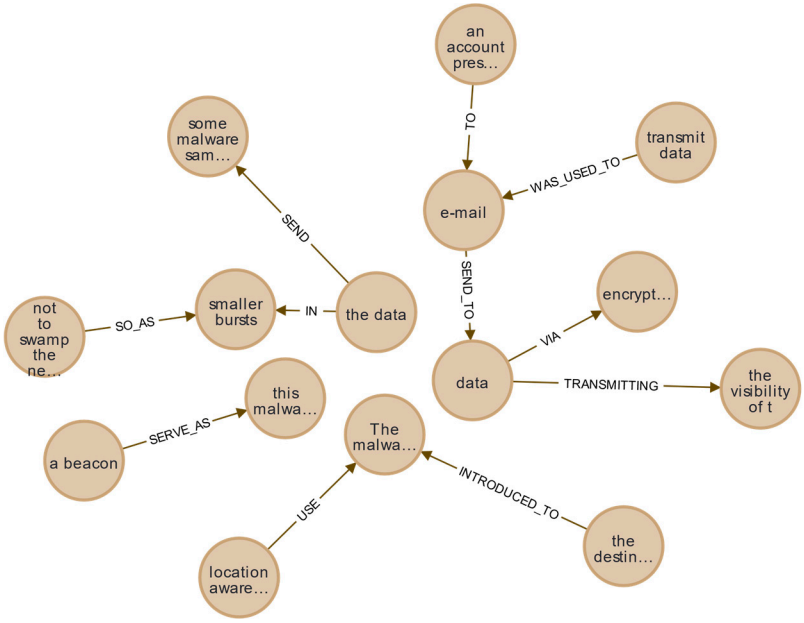

Figure 9. Initial Knowledge Graph Created Using Neo4J.

These challenges stem from the heterogeneous nature of threat intelligence, where varying data sources express the same cybersecurity concepts in diverse ways, leading to both redundancy and semantic ambiguity.

These challenges arise from the heterogeneous nature of threat intelligence—where different data sources employ diverse expressions for the same cybersecurity concepts—leading to issues such as redundant information, ambiguous semantics, and non-standardized nodes within the knowledge graph. To address these problems, this module implements a three-tier optimization strategy:

(1) Triple Refinement: Interfering tokens within NER-labeled entities (e.g., "the", "this") are removed from the graph, retaining only core entities with complete NER annotations.

(2) Entity Disambiguation: To resolve the issue of multiple surface forms for the same entity (e.g., "the malware" vs. "this malware"), context-aware word embeddings are employed. Entity fusion is achieved using hierarchical agglomerative clustering (HAC) with a fully connected strategy. The most frequent core term (e.g., retaining "malware" over "beacon") is selected as the standardized entity name.

(3) Relation Semantic Reconstruction: Semantically overlapping relations in the graph are aligned to the standardized relation set defined by the ontology.

The optimized knowledge network is illustrated in Figure 10. Its structural foundation adheres to the definition in Equation (17):

$$KG = \{(nh, e, nt) | nh, nt \in E, e \in R\} \tag{17}$$

Here, each knowledge triple (nh, e, nt) represents a structured relation between head entity nh and tail entity nt from the entity set E, connected via relation e from the relation set R. Each node is assigned a named entity tag as an attribute.

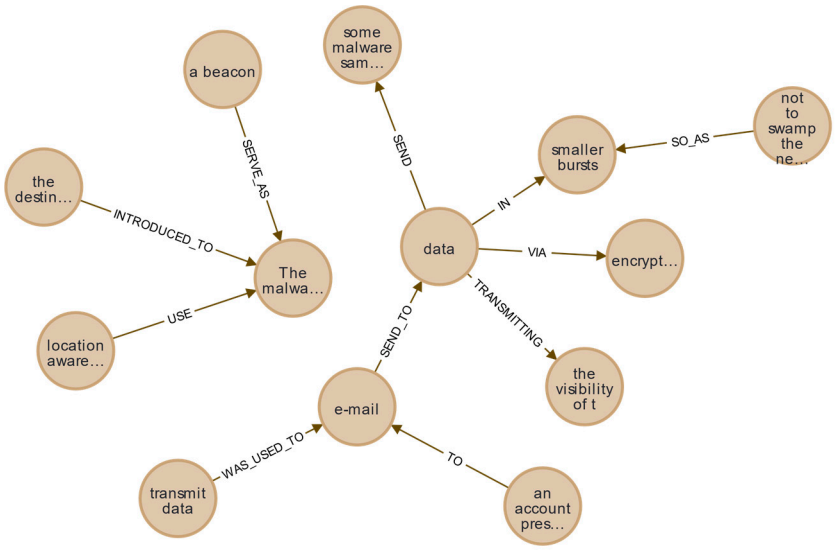

Figure 10. Standardized Knowledge Graph.

4. Experimental Results and Evaluation

4.1. Dataset and Experimental Setup

The training data comprises two cybersecurity datasets: DNRTI for threat intelligence entity recognition and MalwareDB for malware behavior analysis. Both datasets cover APT group activities and malware characteristics, employing the BIO tagging scheme for precise NER/RE boundary annotation.

DNRTI is a cybersecurity-specific NER dataset with over 300 threat reports, 30,000 entities, and 175,000 tokens. Spanning 12 entity categories, it employs preprocessing (cleaning/merging/coreference resolution) to enhance annotation quality. While lacking relation annotations, its specialized coverage aligns with our ontology (Section 3.3.1) and provides critical evaluation benchmarks. Entity distributions are shown in Table 4.

Table 4. Entity Category Distribution Statistics of the DNRTI Dataset.

Entity Types	Meaning	Examples	Count
Area	Location	China	3426
Exp	Exploit	EternalBlue	1991
Features	Characteristics	Encrypted	2439
HackOrg	Hacker group	Fancy Bear	4482
OffAct	Attack pattern	Phishing	2674
Org	Industry	Government	4689
Purp	Motivation	Ransom	2457
SamFile	File	passwd	2346
SecTeam	Security team	Cisco Talos	1918
Time	Time	Q4 2023	2643
Tool	Tool	Mimikatz	4915
Way	Technique	Spear-phishing	2370

As our RE benchmark, MalwareDB uses the MAEC vocabulary for APT malware analysis. From 6819 initial sentences, 1901 were retained after filtering. The annotated subset contains approximately 5200 entity mentions and 1650 relation instances, covering six entity categories (Malware, Vulnerability (CVE), Tool, Exploit, Organization, and Indicator) and four relation types (uses, exploits, targets, and developed_by). Compared with DNRTI, MalwareDB is narrower in report coverage but more relation-oriented and complex: sentences are longer; about 28% of relations span multiple sentences; and 17%

of entities are overlapping or nested. These characteristics increase task difficulty, making MalwareDB suitable for evaluating relation extraction and attack-chain parsing with temporal dependency models (Section 3.2.3).

By using both datasets, we create a multidimensional evaluation framework. DNRTI's cybersecurity-specific entity coverage rigorously validates semantic accuracy in node identification, while MalwareDB's behavioral depth measures relation extraction performance across attack scenarios. Their distinct annotation schemas support our dual-task framework without mixing data sources, creating a synergistic assessment system where each dataset's limitations are balanced by the other's strengths. This pairing provides a comprehensive evaluation framework aligned with our research objectives in attack-chain modeling and threat intelligence structuring.

This pairing ensures the limitations of each dataset are balanced by the other's strengths, providing a comprehensive evaluation framework aligned with our research goals. To maintain evaluation integrity, standard 70–20–10% partitioning is separately applied to each dataset, preserving their unique characteristics while supporting comparative analysis of NER and RE components.

4.2. Parameter Settings

To optimize model performance on entity recognition and relation extraction tasks, this study investigates the potential impact of different parameter configurations. A series of experiments were conducted to examine how variations in key hyperparameters influence model effectiveness.

For instance, epoch number and word embedding dimension were selected as representative parameters. The number of epochs determines the convergence behavior of the model, while the dimensionality of word embeddings governs the semantic representation capacity, which directly affects the model's ability to interpret complex attack chains in long cybersecurity texts.

The experimental results, demonstrating the influence of these parameters on model performance, are summarized in Figure 11.

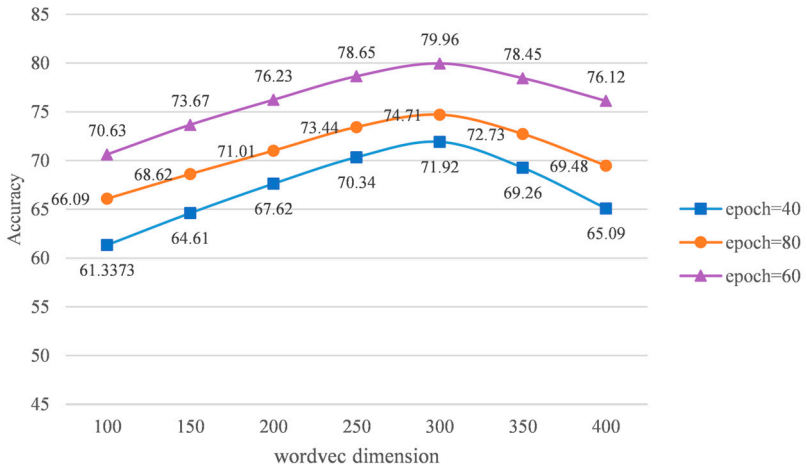

Figure 11. Accuracy Comparison Across Different Dimensions.

Experimental results indicate that the model achieves optimal overall performance when the number of epochs is set to 60. Additionally, the performance of the SecureBERT_Plus-BiLSTM-Attention-CRF model improves with larger word embedding dimensions, but not in a strictly linear way. The highest recognition accuracy occurs at a dimensionality of 300. However, at 400 dimensions, accuracy declines, suggesting a non-linear relationship between embedding dimension and model performance. This highlights the importance

of determining hyperparameter values through comparative experiments tailored to the specific task.

Based on these findings, 60 epochs and 300-dimensional embeddings were set as optimal configurations. Domain-specific constraints further refined additional parameters: a 0.5 dropout rate was used to mitigate overfitting in entity-sparse reports, the BiLSTM architecture was set to 200 units to match median attack chain lengths, and Adam optimizer parameters followed NLP best practices. Complete hyperparameter specifications are detailed in Table 5.

Table 5. Model Parameter Selection.

Parameters	Value
Word dimension	300
Dropout	0.5
Batch-size	64
Epoch	60
Hidden-act	ReLU
Hidden-dim	768
Learning Rate	0.0001
LSTM-size	200
Max_position embedding	512

4.3. Experimental Results and Analysis

4.3.1. Named Entity Recognition

In this study, the accuracy of entity and relation extraction in cybersecurity data depends on two key factors: (1) the model's ability to accurately label entity boundaries and categories, and (2) the model's adaptability to the semantic features of the cybersecurity domain. Only when both conditions are satisfied can high precision be achieved in the final results.

To evaluate the performance of the SecureBERT_Plus-BiLSTM-Attention-CRF model in Named Entity Recognition (NER) tasks, Precision, Recall, and F1 score were used as the evaluation metrics.

Precision refers to the proportion of correctly predicted entities (True Positives, TP) out of all predicted entities (False Positives, FP), measuring the accuracy of the predictions.

Recall indicates the ratio of correctly predicted entities to the total number of true entities (False Negatives, FN), reflecting the comprehensiveness of entity identification.

F1 score is the harmonic mean of Precision and Recall, balancing both metrics and mitigating the evaluation bias in imbalanced datasets.

Moreover, Accuracy was not considered as a performance metric due to its susceptibility to bias in imbalanced data, which can lead to misleading results. In contrast, the F1 score is a more reliable measure of model performance, widely used in information extraction tasks.

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$F1score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{20}$$

To validate the performance of the proposed SecureBERT_Plus-BiLSTM-Attention-CRF model for cybersecurity NER, nine experiments were conducted to compare and assess its effectiveness in Cyber Threat Intelligence (CTI). Experiment 1 evaluated the CNN-BiGRU-CRF model, Experiment 2 the CNN-BiLSTM-CRF model, Experiment 3 the RoBERTa-

BiGRU-CRF model, Experiment 4 the RoBERTa-BiLSTM-CRF model, and Experiment 5 the SecureBERT_Plus-BiLSTM-CRF model, Experiment 6 assessed the performance of the SecureBERT_Plus-BiLSTM-Attention model, Experiment 7 the SecureBERT_Plus- Attention-CRF, Experiment 8 the SecureBERT_Plus-BiLSTM-Attention-CRF model, and Experiment 9 the SecureBERT_Plus-BiLSTM-Attention-CRF to evaluate component contributions.

Table 6 displays the precision (P), recall (R), and F1 scores for the extraction of entity-relation triples, including overlapping relation triples. The results demonstrate clear performance patterns across architectures.

Table 6. Comparison of Experimental Results.

Model	All			Overlap		
	P	**R**	**F1**	**P**	**R**	**F1**
CNN-BiGRU-CRF	0.657	0.635	0.646	0.622	0.634	0.628
CNN-BiLSTM-CRF	0.668	0.647	0.657	0.633	0.646	0.639
RoBERTa-BiGRU-CRF	0.801	0.704	0.749	0.759	0.672	0.713
RoBERTa-BiLSTM-CRF	0.823	0.727	0.772	0.781	0.695	0.735
SecureBERT_Plus-BiLSTM-CRF	0.858	0.754	0.803	0.816	0.722	0.766
SecureBERT_Plus-BiLSTM-Attention	0.868	0.772	0.817	0.826	0.729	0.775
SecureBERT_Plus-Attention-CRF	0.878	0.771	0.823	0.837	0.733	0.784
CySecBERT-BiLSTM-Attention-CRF	0.846	0.745	0.792	0.808	0.731	0.768
SecureBERT_Plus-BiLSTM-Attention-CRF	0.894	0.783	0.839	0.852	0.753	0.803

According to the experimental results, Experiment 9 achieved F1 scores of 0.839 in the ALL scenario and 0.803 in the OVERLAP scenario. This demonstrates that the proposed technique offers a direct and effective solution for entity and relation extraction by fully leveraging the semantic interplay between the two tasks.

Comparing Experiments 1 and 2, as well as 3 and 4, shows that BiLSTM consistently outperforms BiGRU, both in CNN-based architectures and with RoBERTa. This underscores BiLSTM's superior capability in capturing contextual dependencies, validating its effectiveness in NER tasks. Specifically, the CNN-BiLSTM-CRF model achieved an F1 score of 0.657, representing a 1.7% improvement over the CNN-BiGRU-CRF model (0.646). Similarly, the RoBERTa-BiLSTM-CRF model reached an F1 score of 0.772, outperforming RoBERTa-BiGRU-CRF (0.749) by 3.1%.

This gain is mainly due to BiLSTM's memory retention mechanism, which effectively captures long-range dependencies in cybersecurity texts. Compared to BiGRU, BiLSTM excels at modeling relationships between distant tokens, making it more effective and stable when handling long entity boundaries and complex linguistic contexts.

Experiment 4 showed a 17.5% increase in F1 score over Experiment 2, highlighting the superior capabilities of BERT-based embeddings. This improvement reflects the strong generalization enabled by pre-trained language models, confirming their role in semantic modeling and cross-sentence dependency learning.

The SecureBERT_Plus model demonstrates measurable domain adaptation advantages, achieving 0.858 entity recognition precision. This represents a 4.25% improvement over the general-purpose RoBERTa model's 0.823 precision. This improvement comes from the domain-adaptive pretraining strategy, which improves the model's understanding of cybersecurity-specific terms and entities.

The integration of Attention results in steady improvements: Experiment 6 achieved 0.775 F1 for overlapping relations, Experiment 7 reached 0.784 F1, and Experiment 9 achieved the optimal 0.803 F1. These results validate the role of Attention in resolving nested entities, such as polymorphic malware variants.

Experiment 9 outperforms Experiment 8 in both F1 score and Overlap F1 by 5.9% and 4.6%. This improvement is primarily attributed to SecureBERT_Plus's superior domain adaptation and the use of RoBERTa architecture, enhances its ability to handle cross-sentence dependencies and complex entity interactions in CTI tasks. These capabilities are further validated by the integration of Attention and CRF. Attention resolves nested entities, such as polymorphic malware variants, while CRF ensures label sequence coherence, which is essential for maintaining logical attack progression in multi-stage scenarios.

These advancements translate directly into the core inputs for Section 4.3.3 on Knowledge Graph Construction: an entity recognition precision of 0.858 ensures node reliability, while a relation F1 score of 0.799 supports cross-sentence attack chain extraction, collectively laying a solid foundation for semantic integration and topological optimization in the final knowledge graph.

4.3.2. Evaluation of Entity Normalization

Entity normalization is a crucial step in the construction of cybersecurity knowledge graphs, aiming to enhance accuracy and query consistency by merging synonymous entities across data sources. At present, many approaches rely on static word embeddings or rule-based methods for entity alignment. However, these methods have limitations in capturing complex semantic relationships, especially with polysemy and synonymy, leading to low precision in entity aggregation.

To address these issues, the CyberKG framework uses SecureBERT_Plus to generate domain-aware contextual embeddings, combined with a Hierarchical Agglomerative Clustering (HAC)-based method for semantic clustering. This approach improves entity normalization accuracy while overcoming the limitations of traditional methods.

To verify the effectiveness of the proposed entity normalization, this study uses human-annotated entity alignment pairs from the DNRTI and MalwareDB datasets as the gold standard and evaluates performance with multi-granularity metrics. The evaluation system is based on established methods in knowledge base canonicalization, and is defined as follows:

Let $C = \{c_1, c_2, \ldots, m\}$ be the set of entity clusters generated by the normalization system, and $G = \{g_1, g_2, \ldots, g_n\}$ be the set of gold standard clusters. The metrics include macro-level, micro-level, and pairwise evaluations.

First, macro-level metrics are used to evaluate the semantic completeness of clusters, including macro precision (P_{macro}) and macro recall (R_{macro}). P_{macro} is the proportion of fully pure clusters among those generated by the system, i.e., the proportion of entities correctly grouped within each cluster. The formula is defined as

$$P_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{|C_i \cap G_i|}{|C_i|} \tag{21}$$

where $|C_i|$ and G_i represent the number of entities in the system-generated and gold standard clusters, respectively, and $|C_i \cap G_i|$ is the size of their intersection.

R_{macro} is the reciprocal of P_{macro} and reflects the coverage of gold standard clusters by the system, i.e., whether the system can effectively cover all gold clusters:

$$R_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{|C_i \cap G_i|}{|G_i|} \tag{22}$$

Second, micro-level precision (P_{micro}) measures the disambiguation accuracy at the entity level. P_{micro} quantifies the intersection size between generated and gold standard clusters, reflecting the reliability of local merges:

$$P_{micro} = \frac{\sum_{i=1}^{|C|} hits(C_i)}{\sum_{i=1}^{|C|} total(C_i)} \tag{23}$$

where $hits(C_i)$ denotes the number of correctly matched entities in cluster c, and $total(C_i)$ is the total number of entities in C_i.

Finally, pairwise precision ($P_{pairwise}$) evaluates global consistency by verifying coreference relations between entity pairs—addressing cross-source alias mapping issues such as "CVE-2023-1234" and "ProxyLogon". It is calculated as follows:

$$P_{pairwise} = \frac{hits_c}{pairs_c} \tag{24}$$

$$pairs_c = \frac{|c|(|c|-1)}{2} \tag{25}$$

where $hits_c$ is the number of correctly merged entity pairs in cluster c, and $pairs_c$ is the total number of possible entity pairs in that cluster.

This study adopts the F-measure (F1-score) as the core evaluation metric, defined as the harmonic mean of precision and recall. The optimal HAC threshold is determined via grid search on the validation set. Entity similarity is computed using SecureBERT_Plus-generated context-aware embeddings, obtained by encoding full threat intelligence sentences to capture fine-grained cybersecurity semantics. The results are presented in Table 7.

Table 7. Regularization Results.

Metric	**Result**		
	Recall	**Precision**	**F-Measure**
Macro	87.6%	80.9%	84.1%
Micro	91.7%	75.6%	82.9%
Psirwise	80.6%	56.8%	66.6%

The experimental results demonstrate that the generated entity clusters effectively capture the semantic meaning of entities, and cover a large portion of the gold standard entities with high recall. At the macro level, the semantic completeness of the entity clusters achieves 87.6% recall and 80.9% precision (F1-score = 84.1%). This indicates that over 84% of the gold standard entities are covered by the generated clusters, with 80% of the entities within the clusters being semantically pure (e.g., "Fancy Bear" and "APT28" are 100% merged as the same attack group).

Although micro-level precision is slightly lower, it still demonstrates the method's strong performance in entity disambiguation, particularly when handling cross-source entities. The disambiguation of coreferential entity pairs shows clustering consistency with 80.6% recall and 56.8% precision. While the pairwise precision is slightly lower, these results provide valuable directions for future optimizations, particularly in enhancing the accuracy of cross-source alias mappings.

Thus, it can be concluded that the proposed method meets the expected goals in improving entity disambiguation accuracy. Future work could focus on further optimizing the HAC strategy or integrating more advanced deep learning methods to improve cross-source alias handling and enhance global consistency.

4.3.3. Knowledge Graph Generation

The refined knowledge triples are imported into the Neo4j database. As an open-source graph database system, Neo4j efficiently stores and queries structured data, supports transactional operations, and ensures data consistency and reliability [33]. Compared to other graph databases, Neo4j offers superior query performance for entity relationships, making it ideal for complex environments and unstable networks. Therefore, it is chosen as the backend for storing BAKG's data and knowledge [34].

During import, entities are stored as nodes with attributes like type and name, while relationships are stored as edges annotated with relation types. Figure 12 presents a visualization of the cybersecurity knowledge graph constructed in this study.

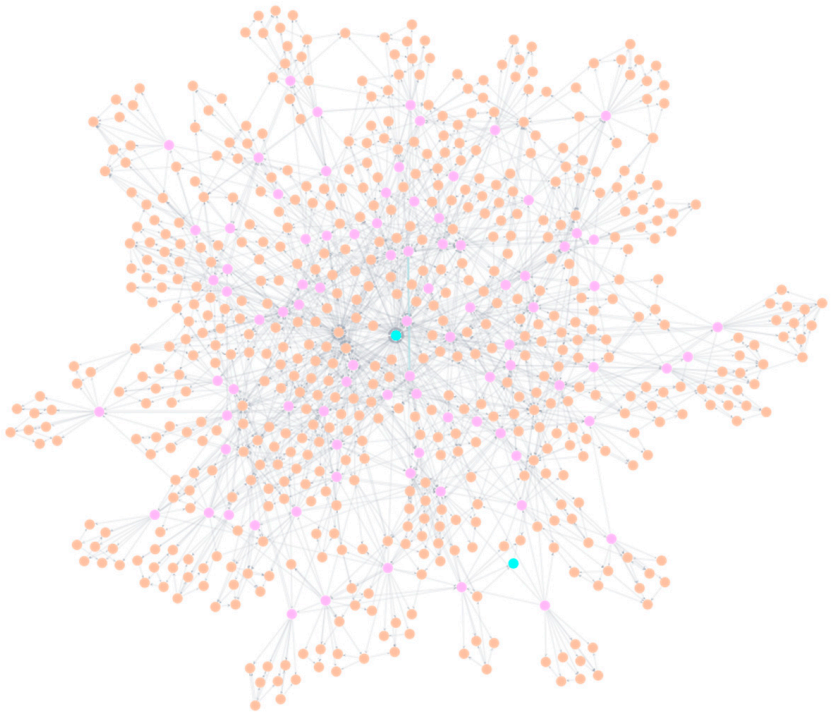

Figure 12. Overview of the Cybersecurity Knowledge Graph.

In the topological analysis of the cybersecurity knowledge graph, the network of inter-entity relationships reveals key insights into complex attack patterns. Figure 13 illustrates a detailed view of the node "An Online E-mail", which represents an online email entity. This node connects to entities like 'Malicious Email', 'Exploit Payload', and 'Keylogger' through various relationships. These links highlight the role of online emails in cyberattacks, such as delivering malware and using social engineering tactics.

Additionally, the node "Custom Malware" is linked to several malicious software components, such as "Rootkit" and "Malicious Script", demonstrating how attackers leverage custom malware for data exfiltration, persistent access, and payload distribution. By examining these details, we can better understand the tactics, techniques, and procedures (TTPs) used in cyberattacks, helping in the analysis and defense against sophisticated threats.

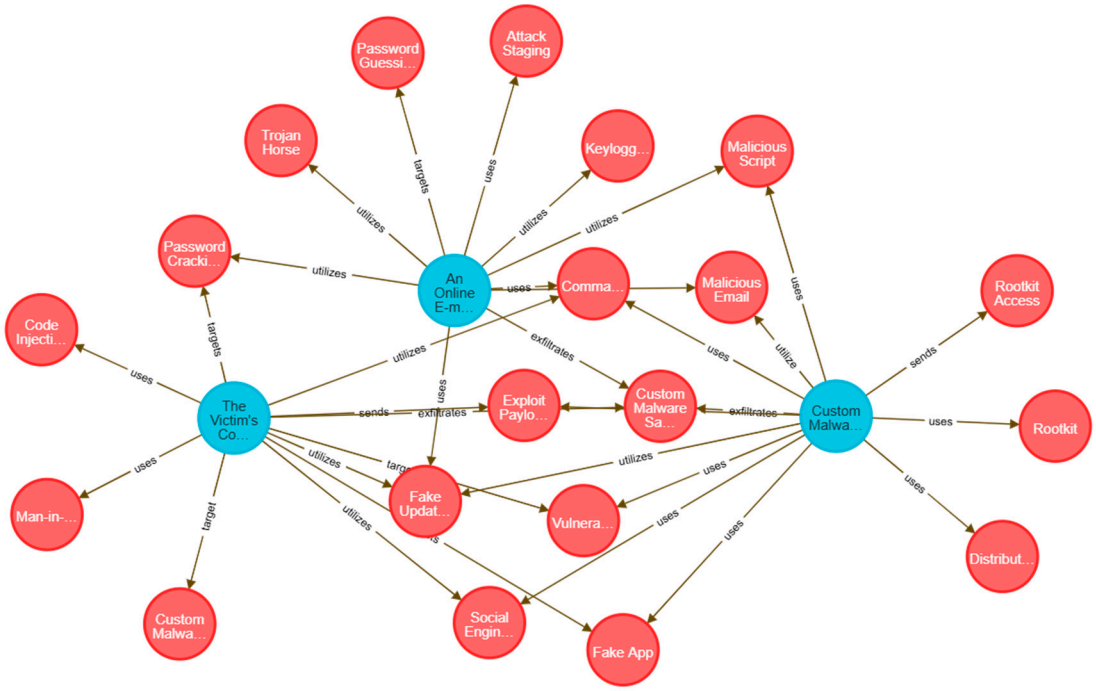

Figure 13. Local Analysis Based on Figure 12.

5. Discussion

This paper presents CyberKG, a framework for constructing interpretable cybersecurity knowledge graphs from unstructured CTI reports. CyberKG improves upon prior pipeline approaches with three innovations: domain-adaptive encoding using SecureBERT_Plus, a joint extraction design combining BiLSTM, attention, and CRF decoding, and context-aware clustering for dynamic entity fusion.

Experiments on DNRTI and MalwareDB indicate that CyberKG achieves good performance. The model shows stable results in NER, overlapping-relation F1, and semantic consistency, demonstrating its ability to handle cross-sentence reasoning and complex entity interactions in CTI. Although DNRTI and MalwareDB are domain-specific rather than widely shared benchmarks, they are widely recognized in cybersecurity research and provide a practical setting for evaluation.

While CyberKG shows good potential, it also has several limitations. The 512-token input window may fragment lengthy APT reports. Current processing is limited to English CTI, excluding non-English intelligence. Knowledge graph updates also require full regeneration, which restricts real-time use.

As future work, we will extend evaluation to widely used shared datasets once suitable cybersecurity-oriented annotations and reproducible implementations become available. We also intend to explore larger context windows, multilingual support, and incremental updating to make the framework more practical for real-time threat monitoring.

Author Contributions: Conceptualization, B.L. and Q.Y.; methodology, B.L. and Q.Y.; software, B.L.; validation, B.L. and C.D.; formal analysis, B.L.; investigation, Q.Y. and H.P.; resources, B.L.; writing—original draft preparation, Q.Y.; writing—review and editing, B.L., C.D. and H.P.; visualization, C.D.; supervision, B.L. and H.P.; funding acquisition, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Sichuan Science and Technology Program, Grant No. 2024NSFSC0515.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The DNRTI dataset can be accessed via the GitHub repository https://github.com/SCreaMxp/DNRTI-A-Large-scale-Dataset-for-Named-Entity-Recognition-in-Threat-Intelligence (accessed on 3 August 2024) or the Hugging Face dataset page https://huggingface.co/datasets/pritamdeka/dataset_dnrti_valid (accessed on 3 August 2024). The MalwareDB dataset is available at https://malwaredb.net/ (accessed on 5 August 2024). If access to these datasets is restricted or requires specific permissions, please contact the corresponding author for more information.

Conflicts of Interest: Author Hua Pan was employed by Guangxi Beitou Innovation Technology Investment Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yan, Z.; Liu, J. A Review on Application of Knowledge Graph in Cybersecurity. In Proceedings of the 2020 International Signal Processing, Communications and Engineering Management Conference (ISPCEM), Montreal, QC, Canada, 27–29 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 240–243. [CrossRef]
2. Zheng, S.; Hao, Y.; Lu, D.; Bao, H.; Xu, J.; Hao, H.; Xu, B. Joint Entity and Relation Extraction Based on a Hybrid Neural Network. *Neurocomputing* **2017**, *257*, 59–66. [CrossRef]
3. Li, J.; Li, J.; Xie, C.; Liang, Y.; Qu, K.; Cheng, L.; Zhao, Z. PipCKG-BS: A Method to Build Cybersecurity Knowledge Graph for Blockchain Systems via the Pipeline Approach. *J. Circuits Syst. Comput.* **2023**, *32*, 2350274. [CrossRef]
4. Guo, L.; Li, X.; Yan, F.; Lu, Y.; Shen, W. A Method for Constructing a Machining Knowledge Graph Using an Improved Transformer. *Expert Syst. Appl.* **2024**, *237*, 121448. [CrossRef]
5. Niakanlahiji, A.; Wei, J.; Chu, B.T. A Natural Language Processing Based Trend Analysis of Advanced Persistent Threat Techniques. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2995–3000. [CrossRef]
6. Satvat, K.; Gjomemo, R.; Venkatakrishnan, V. Extractor: Extracting Attack Behavior from Threat Reports. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 598–615. [CrossRef]
7. Sarhan, I.; Spruit, M. Open-CyKG: An Open Cyber Threat Intelligence Knowledge Graph. *Knowl.-Based Syst.* **2021**, *233*, 107524. [CrossRef]
8. Kim, G.; Lee, C.; Jo, J.; Lim, H. Automatic Extraction of Named Entities of Cyber Threats Using a Deep Bi-LSTM-CRF Network. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 2341–2355. [CrossRef]
9. Ren, Y.; Xiao, Y.; Zhou, Y.; Zhang, Z.; Tian, Z. CSKG4APT: A Cybersecurity Knowledge Graph for Advanced Persistent Threat Organization Attribution. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 5695–5709. [CrossRef]
10. Zhou, Y.; Tang, Y.; Yi, M.; Xi, C.; Lu, H. CTI View: APT Threat Intelligence Analysis System. *Secur. Commun. Netw.* **2022**, *2022*, 9875199. [CrossRef]
11. Srivastava, S.; Paul, B.; Gupta, D. Study of Word Embeddings for Enhanced Cyber Security Named Entity Recognition. *Procedia Comput. Sci.* **2023**, *218*, 449–460. [CrossRef]
12. Wang, X.; He, S.; Xiong, Z.; Wei, X.; Jiang, Z.; Chen, S.; Jiang, J. APTNER: A Specific Dataset for NER Missions in Cyber Threat Intelligence Field. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1233–1238. [CrossRef]
13. Bekoulis, G.; Deleu, J.; Demeester, T.; Develder, C. Adversarial Training for Multi-Context Joint Entity and Relation Extraction. *arXiv* **2018**, arXiv:1808.06876.
14. Zhang, Z.; Sind, X.; Liu, T.; Fang, Z.; Li, Q. Joint Entity Linking and Relation Extraction with Neural Networks for Knowledge Base Population. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8. [CrossRef]
15. Zhao, J.; Yan, Q.; Liu, X.; Li, B.; Zuo, G. Cyber Threat Intelligence Modeling Based on Heterogeneous Graph Convolutional Network. In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), Taipei, China, 14–16 October 2020; USENIX Association: Berkeley, CA, USA, 2020; pp. 1–16.
16. Guo, Y.; Liu, Z.; Huang, C.; Wang, N.; Min, H.; Guo, W.; Liu, J. A Framework for Threat Intelligence Extraction and Fusion. *Comput. Secur.* **2023**, *132*, 103371. [CrossRef]

17. Wang, X.; Xiong, M.; Luo, Y.; Li, N.; Jiang, Z.; Xiong, Z. Joint Learning for Document-Level Threat Intelligence Relation Extraction and Coreference Resolution Based on GCN. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8. [CrossRef]
18. Zuo, J.; Gao, Y.; Li, X.; Yuan, J. An End-to-End Entity and Relation Joint Extraction Model for Cyber Threat Intelligence. In Proceedings of the 2022 7th International Conference on Big Data Analytics (ICBDA), Guangzhou, China, 4 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 204–209. [CrossRef]
19. Ahmed, K.; Khurshid, S.K.; Hina, S. CyberEntRel: Joint Extraction of Cyber Entities and Relations Using Deep Learning. *Comput. Secur.* **2024**, *136*, 103579. [CrossRef]
20. Dey, R.; Debnath, A.; Dutta, S.K.; Ghosh, K.; Mitra, A.; Roychowdhury, A.; Sen, J. Semantic Stealth: Adversarial Text Attacks on NLP Using Several Methods. *arXiv* **2024**, arXiv:2404.05159. [CrossRef]
21. Bayer, M.; Kuehn, P.; Shanehsaz, R.; Reuter, C. CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain. *ACM Trans. Priv. Secur.* **2024**, *27*, 18. [CrossRef]
22. Park, Y.; You, W. A Pretrained Language Model for Cyber Threat Intelligence. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, Singapore, 6–10 December 2023; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 113–122. [CrossRef]
23. Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E. SecureBERT: A Domain-Specific Language Model for Cybersecurity. In Proceedings of the Security and Privacy in Communication Networks, Virtual, 17–19 October 2022; Li, F., Liang, K., Lin, Z., Katsikas, S.K., Eds.; Springer Nature: Cham, Switzerland, 2023; Volume 462, pp. 39–56. [CrossRef]
24. Li, W.; Du, Y.; Li, X.; Chen, X.; Xie, C.; Li, H.; Li, X. UD_BBC: Named Entity Recognition in Social Network Combined BERT-BiLSTM-CRF with Active Learning. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105460. [CrossRef]
25. Xu, H.; Fan, G.; Kuang, G.; Wang, C. Exploring the Potential of BERT-BiLSTM-CRF and the Attention Mechanism in Building a Tourism Knowledge Graph. *Electronics* **2023**, *12*, 1010. [CrossRef]
26. Dai, S.; Ding, Y.; Zhang, Z.; Zuo, W.; Huang, X.; Zhu, S. GrantExtractor: Accurate Grant Support Information Extraction from Biomedical Fulltext Based on Bi-LSTM-CRF. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *18*, 205–215. [CrossRef]
27. Xu, Y.; Tan, X.; Tong, X.; Zhang, W. Robust Chinese Named Entity Recognition Method Based on Integrating Dual-Layer Features and CSBERT. *Appl. Sci.* **2024**, *14*, 1060. [CrossRef]
28. Syed, Z.; Padia, A.; Finin, T.; Mathews, L.; Joshi, A. UCO: A Unified Cybersecurity Ontology. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Palo Alto, CA, USA, 12 February 2016; pp. 1–8.
29. Mouiche, I.; Saad, S. Entity and Relation Extractions for Threat Intelligence Knowledge Graphs. *Comput. Secur.* **2025**, *148*, 104120. [CrossRef]
30. Wang, G.; Liu, P.; Huang, J.; Bin, H.; Wang, X.; Zhu, H. KnowCTI: Knowledge-Based Cyber Threat Intelligence Entity and Relation Extraction. *Comput. Secur.* **2024**, *141*, 103824. [CrossRef]
31. 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]
32. Alshahrani, M.; Thafar, M.A.; Essack, M. Application and Evaluation of Knowledge Graph Embeddings in Biomedical Data. *PeerJ Comput. Sci.* **2021**, *7*, e341. [CrossRef] [PubMed]
33. Li, L.; Wang, P.; Yan, J.; Wang, Y.; Li, S.; Jiang, J.; Sun, Z.; Tang, B.; Chang, T.-H.; Wang, S.; et al. Real-World Data Medical Knowledge Graph: Construction and Applications. *Artif. Intell. Med.* **2020**, *103*, 101817. [CrossRef] [PubMed]
34. Zeng, X.; Tu, X.; Liu, Y.; Fu, X.; Su, Y. Toward Better Drug Discovery with Knowledge Graph. *Curr. Opin. Struct. Biol.* **2022**, *72*, 114–126. [CrossRef] [PubMed]