Computers & Security 132 (2023) 103371

Contents lists available at ScienceDirect

Computers & Security

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

A framework for threat intelligence extraction and fusion[☆],[☆☆]

Yongyan Guo, Zhengyu Liu, Cheng Huang[*], Nannan Wang, Hai Min, Wenbo Guo, Jiayong Liu

School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China

ARTICLE INFO

Article history:
Received 24 January 2022
Revised 19 June 2023
Accepted 26 June 2023
Available online 28 June 2023

Keywords:
Threat intelligence
Knowledge graph
Relation extraction
Joint model
Knowledge fusion

ABSTRACT

Cyber-attacks, with various emerging attack techniques, are becoming increasingly sophisticated and difficult to deal with, posing great threats to companies and every individual. Therefore, analyzing attack incidents and tracing the attack groups behind them becomes extremely important. Threat intelligence provides a new technical solution for attack traceability by constructing Cybersecurity Knowledge Graph (CKG). In this paper, we propose a framework for threat intelligence extraction and fusion, which is able to extract, correlate and unify cybersecurity entity-relation triples from structured and unstructured data. However, the existing entity and relation extraction for cybersecurity concepts uses the traditional pipeline model that suffers from error propagation and ignores the connection between the two subtasks. To solve the above problem, we propose a joint entity and relation extraction model for cybersecurity concepts. We model the joint extraction problem as a multiple sequence labeling problem, generating separate label sequences for different relations, which contain information about the involved entities and the subject and object of that relation. Experimental results on Open Source Intelligence (OSINT) data show that the F1 value of the joint model is 81.37%, which is better than the previous pipeline model. For the knowledge fusion, we propose an improved Levenshtein distance to correlate the same entities extracted from different data sources to construct a preliminary CKG, which is demonstrated in the Experiments section.

1. Introduction

Nowadays, the damage and impact caused by malicious behaviors in cyberspace such as hacker attacks, frauds, and rumors have become more serious. Therefore, how to effectively and accurately detect cyber attacks as early as possible, analyze attack incidents, and trace the source of attackers and groups has become a severe problem for enterprises and countries.

The concept of Cyber Threat Intelligence (CTI) was developed supplying new theoretic support for cyber-attack source tracing, making it possible to trace the source of a wide range of attacks. Therefore, many researchers extract and analyze different threat intelligence to generate the Cybersecurity Knowledge Graph (CKG). The CKG has the characteristic of strong timeliness and high accuracy, which can timely and easily detect, respond and defend against specific targets, providing a new measure for attack source tracking, and can even effectively deal with sophisticated cyber-attacks (e.g., zero-day attacks, advanced persistent threat).

The key step in constructing CKG is cyber threat intelligence extraction, which involves subtasks such as entity recognition, relation extraction, and event extraction. Currently, many research groups have conducted research on the automated construction and analysis of CKG (Gao et al., 2021; Husari et al., 2017; Jia et al., 2018; Milajerdi et al., 2019; Piplai et al., 2020a; 2020b; Zhao et al., 2020b). In terms of CTI information extraction, previous studies are dedicated to extracting cybersecurity concepts (Liao et al., 2016; Mittal et al., 2016; Zhu and Dumitras, 2018) and entities (Ghazi et al., 2018; Husari et al., 2018; Zhao et al., 2020a) from unstructured data. This extraction leads to a rich repository of cybersecurity entity-relation triples that precisely delineate the relations within the cybersecurity realm. For instance, consider the triple (“sqlite3 in versionc 3.26.0”, “hasVulnerability”, “CVE-2019-5018”). Here, “sqlite3 in version 3.26.0” is the subject entity of type “software”, “hasVulnerability” is the relation, and “CVE-2019-5018” is the object entity of type “vulnerability”. This triple clearly indicates that the software “sqlite3” in version 3.26.0 has a specific vulnerability referenced as “CVE-2019-5018”.

The construction of CKG is inseparable from a large number of cybersecurity entity-relation triples from different sources. Threat intelligence comes from structured and unstructured data such as

[☆] This paper was presented in part at the Proceeding of the 23rd International Conference on Information and Communications Security (ICICS 2021) (Guo et al.(2021)).

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

[*] Corresponding author.
E-mail address: codesec@scu.edu.cn (C. Huang).

https://doi.org/10.1016/j.cose.2023.103371
0167-4048/© 2023 Elsevier Ltd. All rights reserved.

audit logs, network traffic, security alerts, vulnerability databases, security bulletins, hacker forums, and social media. These data have the characteristics of multisource, heterogeneous, polysemy, and highly dependent on domain knowledge. Therefore it is difficult to effectively integrate data from different sources. More importantly, extracting cybersecurity entity-relation triples from unstructured data is a great challenge, and is the key step to constructing CKG. Existing research on cybersecurity entity and relation extraction (Jones et al., 2015; Pingle et al., 2019) uses the traditional pipeline model, named entity recognition first and then relation extraction, which leads to error propagation and losses sight of the relevance between entity recognition and relation extraction.

To solve the above problem, we propose a framework for threat intelligence extraction and fusion which can extract and correlate cybersecurity entity-relation triples from structured and unstructured data. For unstructured data, we propose a joint entity and relation extraction model for cybersecurity concepts, which extracts both cybersecurity entities and relations and generates cybersecurity triples. Specifically, we use a tagging scheme to convert the joint extraction problem into a multiple sequence labeling problem by generating separate label sequences for different relations containing information about the related entities and the subject and object of that relation. The joint extraction model applies the pre-trained model, BERT, to generate word vectors. After extracting semantic features by BiGRU, the model assigns higher weights to relation-related words in the sentences by an attention mechanism. Finally, BiGRU combined with CRF is used to decode and construct cybersecurity triples. Then, we fuse the entities extracted from different data sources by an improved Levenstein distance to form a preliminary CKG.

In summary, the main contribution of this paper are as follows:

- We propose a framework for threat intelligence extraction and fusion that can extract and fuse cybersecurity entity-relation triples from large-scale structured and unstructured data. These triples can be used to construct the CKG.
- We present a joint entity and relation extraction model for cybersecurity concepts. The model employs deep learning techniques to extract entities and relations in sentences simultaneously, avoiding the error propagation of traditional pipeline models. The experimental results show that the joint model outperforms the traditional pipeline model with an F1 value of 81.37%.
- We design a lightweight cybersecurity entity fusion method that is optimized for the features of the cybersecurity corpus by fusing entities from different sources based on an improved Levenshtein distance.

The rest of the paper is organized as follows: Section 2 discusses related work, and Section 3 presents our framework for threat intelligence extraction and fusion. Section 4 provides the experiments and analysis related to this work. Section 5 summarizes the conclusion and proposes future works.

2. Related work

In this section, we first review the methods for automated construction and analysis of CKG. Secondly, since the pivotal step of CKG construction is threat intelligence extraction, we review the work related to CTI extraction including entity recognition, relation extraction, and event extraction subtasks. Finally, we present the related research on relation extraction and knowledge fusion.

2.1. Cybersecurity knowledge graph

The Knowledge Graph (KG) was originally proposed by Google. It is a knowledge base that integrates information from multiple sources, links real-world entities or concepts, and provides search services through semantic retrieval. In the field of cybersecurity, correlating and fusing threat intelligence data from different sources to generate the CKG can provide new technical means for situational awareness and attack traceability.

In the area of automated construction and analysis of CKG, researchers have also proposed several ideas and approaches in recent years (Gao et al., 2021; Husari et al., 2017; Jia et al., 2018; Milajerdi et al., 2019; Piplai et al., 2020a; 2020b; Zhao et al., 2020b). Jia et al. (2018) introduced a cybersecurity knowledge base and deduction rules based on a quintuple model. Gao et al. (2021) proposed ThreatRaptor, a system that facilitates threat hunting in computer systems using OSINT. The system uses an unsupervised, lightweight, and accurate NLP pipeline to extract structured threat behaviors from unstructured OSINT text. Piplai et al. (2020b) described a system that extracts information from After Action Reports (AARs) and represents the extracted information in a CKG. Zhao et al. (2020b) demonstrated a threat intelligence framework (HINTI). HINTI first recognizes IOCs and models the interdependent relations between IOCs using heterogeneous information networks (HINs), and then proposes a threat intelligence computing framework based on graph convolutional networks to explore complex security knowledge. Although these approaches have made initial attempts and achieved good results in CKG construction, further research is needed in the key steps of knowledge graph construction.

2.2. Threat intelligence extraction

The construction of a knowledge graph can be divided into three steps, including information extraction, knowledge fusion, and knowledge reasoning. Among them, information extraction plays a decisive role in the quality of the generated knowledge graph. Information extraction for threat intelligence is divided into several subtasks, including entity recognition (Ghazi et al., 2018; Husari et al., 2018; Liao et al., 2016; Mittal et al., 2016; Zhao et al., 2020a; Zhu and Dumitras, 2018), relation extraction (Jones et al., 2015; Pingle et al., 2019) and event extraction (Satyapanich et al., 2020).

In terms of cybersecurity entity and concept recognition, Mittal et al. (2016) proposed a framework for extracting threat intelligence from Twitter, CyberTwitter, which automates the extraction of security vulnerability concepts. Liao et al. (2016) introduced iACE for automatically extracting IOCs and their context in the sentences of technical articles. Zhu and Dumitras (2018) designed Chainsmith, an IOC extraction system that collects IOCs from security articles and classifies them according to the stages of the Kill Chain. Ghazi et al. (2018) used natural language processing techniques to extract threat sources from unstructured web threat information sources and provided comprehensive threat reports in the STIX (2017) standard, which is used for the accurate and efficient exchange of cyber threat intelligence.

Due to the lack of a well-labeled corpus for training, relatively few studies have been conducted on cybersecurity relation extraction and event extraction compared to entity recognition. Pingle et al. (2019) proposed RelExt, a deep learning-based cybersecurity relation extraction method for constructing CKGs. The model uses a pipeline approach, first identifying entities in the text by an entity recognizer then classifying the relations by a deep learning model. Jones et al. (2015) implemented a semi-supervised cybersecurity relation extraction method based on a bootstrapping algorithm to extract relations. Satyapanich et al. (2020) proposed CASIE, a security event extraction system that uses deep neural networks and can incorporate rich linguistic features and word embeddings for extracting security events related to cyber-attacks and vulnerabilities.

2.3. Relation extraction

As a subtask of information extraction, relation extraction has a long research history. The main approaches to relation extraction can be broadly divided into three categories, including early rule-based approaches (Iria, 2005; McDonald et al., 2005); traditional machine learning-based approaches (Culotta and Sorensen, 2004; Jiang and Zhai, 2007); and deep learning-based approaches (Bekoulis et al., 2018; Miwa and Bansal, 2016; Wei et al., 2020; Zeng et al., 2014; Zheng et al., 2017). In recent years, the latest research results in the field of relation extraction have focused on deep learning models (Dai et al., 2019; Fu et al., 2019; Sun et al., 2018; Yuan et al., 2020). The advantage of deep learning methods is that they do not require manual extraction of features nor a large amount of domain knowledge.

Currently, there are two main approaches to relation extraction based on deep learning: the pipeline approach and the joint approach. The pipeline approach performs relation classification after extracting all the entities. Zeng et al. (2014) first applied CNN to relation extraction to automatically extract lexical and sentence-level features. Wei et al. (2020) proposed a novel cascaded binary annotation framework (CASREL) that models relations as functions that map subjects to objects in a sentence, which naturally handles the overlapping triple problems. Although these methods achieve promising results, the pipeline architectures suffer from the problem of error propagation. In addition, neglecting the relationship between the two tasks of entity recognition and relation extraction for training can also affect the effectiveness of relation extraction. Therefore, to construct the bridge between the two subtasks, building a joint model that extracts entities together with relations simultaneously has attracted much attention. Miwa and Bansal (2016) proposed a joint relation extraction model based on shared parameters, which captures both word sequences and dependency tree substructure information for end-to-end relation extraction via LSTM. Bekoulis et al. (2018) propose a joint model that uses a CRF layer to model the entity recognition task and the relation extraction task as a multi-headed selection problem. Zheng et al. (2017) proposed a new tagging scheme that can convert the joint extraction task to a sequence labeling problem. Yuan et al. (2020) proposed a relation-based attention network (RSAN) to jointly extract entities and relations using a relation-aware attention mechanism.

2.4. Knowledge fusion

Due to the existence of duplicate and complementary information in data from different sources, knowledge fusion is proposed to study how the same entity or concept from multiple sources can be fused to form a high-quality knowledge base (Zhao et al., 2020c). Its necessity has been explained in the recent relevant studies (Alves et al., 2020; Gonzalez-Granadillo et al., 2021; Liu et al., 2022; Yuan et al., 2021). Knowledge fusion needs to be performed at two levels: at the ontology level, equivalence or similar classes, relations, and attributes between different ontologies need to be found, i.e., ontology matching; at the entity level, the same objects from different sources need to be correlated and combined, i.e., entity alignment.

Ontology matching requires abstracting a myriad of concepts and complex relations in the cybersecurity domain into a semantic network. Iannacone et al. (2015) proposed STUCCO, an ontology for building CKGs, integrating 13 different formats of cybersecurity data sources. Building on this foundation, Syed et al. (2016) proposed a Unified Cybersecurity Ontology (UCO). The UCO ontology provides a general understanding of the cybersecurity domain and, in addition to mapping to STIX, UCO extends several related cybersecurity standards, vocabularies, and ontologies such as CVE, CCE, CVSS, CAPEC, CYBOX, KillChain, and STUCCO.

Existing entity alignment studies fall into two main categories, including similarity-based and embedding-based approaches. Similarity-based approaches use information such as entity names, attributes, and relations to compute the similarity of entity pairs. Lacoste-Julien et al. (2013) proposed a simple greedy matching algorithm that uses the structural information of the knowledge graph and the similarity metric between entity attributes to align entities on a large-scale knowledge base. Azevedo et al. (2019) proposed a method to connect different IoCs based on two similarity measures (the *n*-level correlation) in order to generate threat intelligence of quality in the form of enriched IoCs. The embedding-based approach maps the triples in the knowledge graph to the same vector space and aligns entities that are similar to each other by computing the distance between entity vectors. Chen et al. (2016) improved the TransE (Bordes et al., 2013) model by proposing MTransE, which encodes entities and relations for each language in a separate embedding space to achieve cross-language entity alignment. Since knowledge graphs are in graph structure, some studies also use graph neural networks to embed the information of knowledge graphs, for example, Liu et al. (2020) improved the GNN-based entity alignment method by using an attributed value encoder and partition the KG into subgraphs to model various types of attribute triples. Nie et al. (2021) proposed jointly utilizing the global KG structure and entity-specific relational triples to achieve entity alignment.

In the construction of CKG, a lot of research has been conducted on the extraction of cybersecurity entities and concepts, while research on cybersecurity relation extraction is still in its infancy. Existing approaches use traditional pipeline methods, which leads to error propagation and loses sight of the relevance between entity recognition and relation extraction. Different from these above works, this paper proposes a joint entity and relation extraction model for cybersecurity concepts, which extracts entities and relations simultaneously, effectively avoiding the shortcomings of the traditional pipeline model. In addition, our paper further fuses threat intelligence from different sources through a lightweight knowledge fusion algorithm.

3. Framework design

In this section, we will introduce our proposed framework for threat intelligence extraction and fusion. The ontology of our cybersecurity knowledge graph references UCO 2.0 (Syed et al., 2016) and STIX 2.0 STIX (2017).

- The main entity types include: Indicator, Threat Actor, Attack Pattern, Malware, Tool, Campaign, Course of Action, Vulnerability, and Software.
- The main relation types include: hasProduct, hasVulnerability, uses, attributedTo, mitigates, and indicates.

Our framework is divided into four parts, as shown in Fig. 1. In the data collection, threat intelligence data is collected from structured data and unstructured data on the internet, and the framework processes these two types of data separately. For structured data such as STIX data, we directly obtain the entities or triples by ontology matching, as described in Section 3.1. In the threat intelligence extraction, we propose a joint entity and relation extraction model to extract cybersecurity triples from unstructured data, as described in Section 3.2. In the knowledge fusion, we design an improved Levenshtein distance to fuse entities from different sources based on the characteristics of the cybersecurity corpus, as described in Section 3.3. Finally, our framework deposits the cybersecurity triples extracted and fused from structured and unstructured data into the neo4j database to form a preliminary CKG.

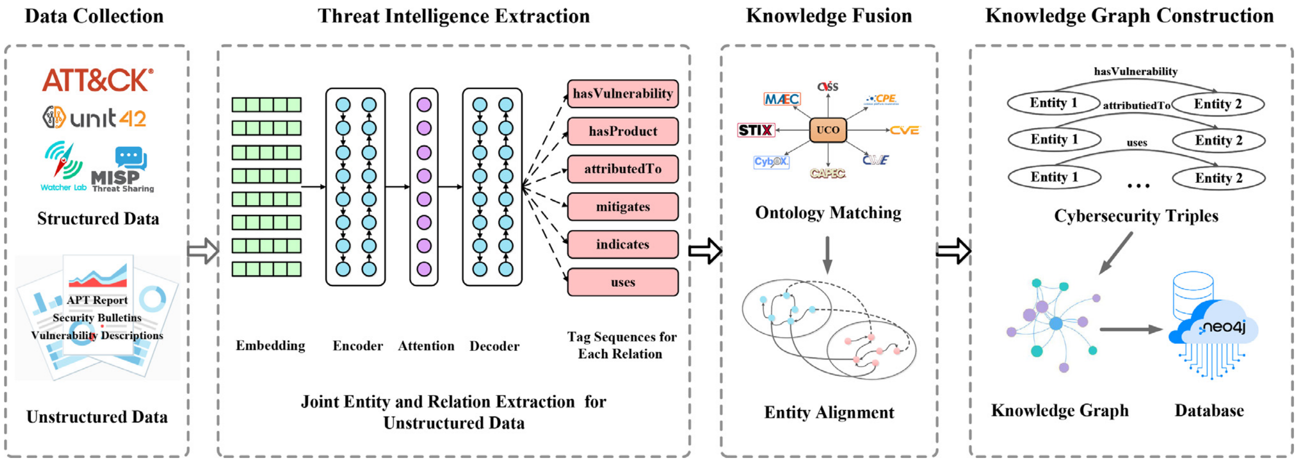


Fig. 1. The framework for threat intelligence information extraction and fusion.

3.1. Data collection and structured data process

The framework regularly collects raw data from various sources such as existing attack and defense knowledge bases, intelligence sharing platforms, vulnerability databases, security bulletins, and APT reports. These data include structured data and unstructured data, which are processed separately by the framework. The structured data only requires simple steps before importing into the knowledge graph, while the unstructured data requires a more complex model to extract the cybersecurity triples from it.

There exist many structured threat intelligence data, including attack and defense knowledge bases (e.g., ATT&CK Strom et al., 2018) and intelligence sharing platforms (e.g., MISP, 2021; Unit 42, 2021; WatcherLab, 2021), which are maintained by professional teams with high quality and rich content, usually in a structured form such as STIX. Collecting these structured data can quickly get a large amount of reliable threat intelligence data. We store the entities and relations in these structured data directly into the neo4j database after ontology matching. In addition, the data sources are not limited to those platforms mentioned above. Structured data from other sources can also be extended into our framework after ontology matching.

3.2. Joint entity and relation extraction for unstructured data

To extract threat intelligence from unstructured data such as vulnerability descriptions, security bulletins, APT reports, technology blogs, and hacker forums, we propose a joint entity and relation extraction model for cybersecurity concepts. Our model can extract cybersecurity triples from unstructured data. We briefly outline the overall strategy here before discussing details in the following subsections. The model takes unstructured threat intelligence data collected from multiple sources as raw input. Then the data undergoes a pre-processing process including data cleaning, sentence segmentation, and tokenization to obtain the training corpus, which will be fed into the joint extraction model subsequently (see Section 3.2.1 for details). We adopt the cybersecurity entities and relations defined in the UCO 2.0 (Syed et al., 2016) ontology and model the joint entity and relation extraction problem as a multiple sequence labeling problem by generating a sequence of labels for each relation through a specific tagging schema (see Section 3.2.2 for details). Each relation label sequence contains information about the entities involved and the subject and object of the relation. Our proposed multiple sequence labeling model is structured into an embedding layer, an encoding layer, an attention layer, and a decoding layer (see Section 3.2.3 for details). Finally, the model constructs cybersecurity triples based on the label sequences predicted by the model, and these triples will eventually be used to construct CKG.

3.2.1. Data preprocess

Unstructured threat intelligence data is usually stored in rich text documents such as PDF, HTML/XML, JSON, and other formats. First, we use various text parsing tools (e.g. HTMLParser, PDFLib) to extract the raw text from these documents. But the extracted raw text is not well-formatted. Therefore, we devised some data pre-processing procedures as follows.

The first step in preprocessing is data cleaning, where we remove non-ASCII characters from the text and whitespace characters at the beginning and end of each sentence. It is worth noting that in some threat intelligence data, special types of entities are often rewritten to prevent readers from clicking on them by mistake. For example, the IP address “136.244.119.85” is rewritten as “136. 244.119[.]85”; the URL http://www.test.com is rewritten to http://www.test.com ; the email address hacker@test.com is rewritten as hacker[at]test.com. We revert this rewritten form to its original form.

The next step in preprocessing is special entity substitution. In the field of cybersecurity, some entities are very different in form from the normal natural language, such as IP, MAC, Hash, URL, Email, domain name, file name, and file path. We build regular expressions to match these entities from text and replace them with natural language strings in the form of “sub type”, where “type” is the type of the special entity. For example, we would replace the IP address “136.244.119.85” with “sub ip”.

The last step in the preprocessing process is text segmentation, which is the process of converting text into sequences. We use the NLTK library for sentence segmentation and WordPiece for word tokenization.

3.2.2. Tagging scheme

In this section, we will introduce the tagging schema for the joint entity and relation extraction. In the field of relation extraction, there has been related work (Dai et al., 2019; Yuan et al., 2020; Zheng et al., 2017) on the joint entity and relation extraction through the construction of a specific tagging schema. For cybersecurity concepts, the extracted relation usually suffers from the entity overlapping problem that different types of relations sharing the same entities, so the tagging scheme has to overcome this issue. Our model generates a sequence of labels for each relation in UCO 2.0 (Syed et al., 2016). In each tag sequence, we use the typical “BIO” signs to locate the entities in the sentence, where “B”

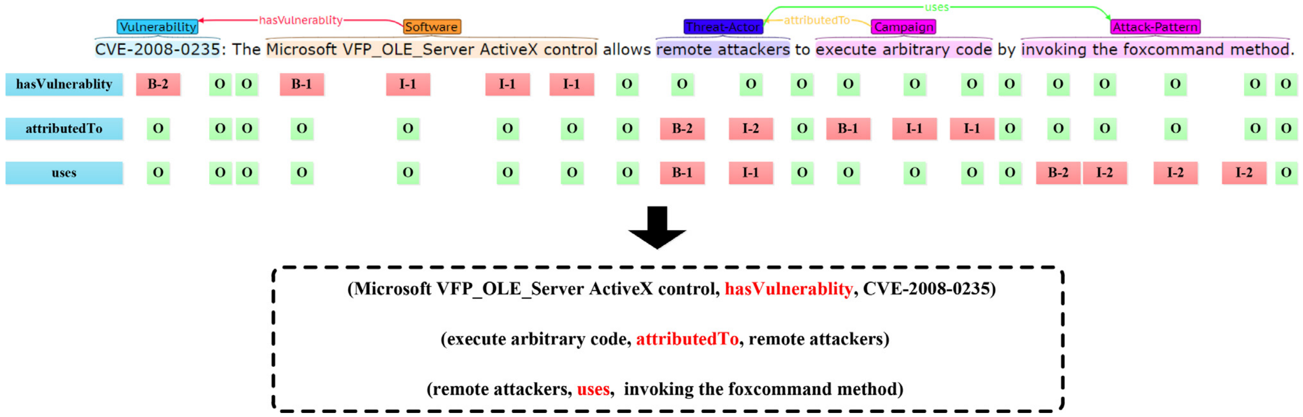

Vulnerability
hasVulnerablity
Software
Threat-Actor
attributedTo
Campaign
uses
Attack-Pattern
CVE-2008-0235: The Microsoft VFP_OLE_Server ActiveX control allows remote attackers to execute arbitrary code by invoking the foxcommand method.
hasVulnerablity
attributedTo
uses
B-2 O O B-1 I-1 I-1 I-1 O O O O O O O O O O O O O O
O O O O O O O O B-2 I-2 O B-1 I-1 I-1 O O O O O O
O O O O O O O O B-1 I-1 O O O O O B-2 I-2 I-2 I-2 O
(Microsoft VFP_OLE_Server ActiveX control, hasVulnerablity, CVE-2008-0235)
(execute arbitrary code, attributedTo, remote attackers)
(remote attackers, uses, invoking the foxcommand method)

Fig. 2. An example for tagging scheme.

represents the starting part of the entity, “I” represents the middle part of the entity, and “O” is the non-entity part. At the same time, we also label the entity as subject or object in the relation, with “1” representing the subject in the triple and “2” representing the object in the triple.

Figure 2 shows an example of the tagging scheme. The first label sequence describes the “*hasVulnerablity*” relation, where “*Microsoft VFP_OLE _Server ActiveX control*” is an entity of type “*Software*”, as the subject of the “*hasVulnerablity*” relation; “*CVE-2008-0235*” is an entity of type “*Vulnerability*”, as the object of the “*hasVulnerablity*” relation. Through the label sequence, we can generate the triple (“*Microsoft VFP_OLE_Server ActiveX control*”, “*hasVulnerability*”, “*CVE-2008-0235*”). Likewise, other label sequences can be used to generate triples of corresponding relations. If a relation does not exist in a sentence, the label sequence for that relation will be all “O”. Besides, as we can see, the “*attributedTo*” and “*uses*” relations have the over-lapped entity “*remote attackers*”, and they can be extracted without conflict based on the separate label sequences.

3.2.3. Multiple sequence labeling model

Based on the tagging scheme above, we propose an end-to-end multiple sequence labeling model to jointly extract cybersecurity entities and relations. We take the sentence and a type of relation as input to the model, and the output sequence holds information about the subject and object entities involved in that relation. Thus, for a sentence, when we traverse all the relation types, the model generates a label sequence for each type of relation, resulting in a joint extraction of entities and relations. Figure 3 gives an overall structure of the model, which is divided into four parts. The embedding layer generates a word vector e_t for each word x_t in sentence X. In the encoding layer, the embedding sentence is fed into the bi-directional Gated Recurrent Units (BiGRU) neural network to generate a hidden state representation h_t. Then we apply the attention mechanism to assign different weights to the context words under different relations and construct a relation-specific sentence representation l_k. Finally, in the decoding layer, we use another BiGRU neural network and joined it with CRF for decoding to obtain the label sequence and extract corresponding entities under the specific relation.

Embedding

Given a sentence as a sequence of tokens, the word embedding layer is responsible to map each token to a word vector. In this paper, we propose to use a pre-trained model to generate word vectors. The pre-trained word embedding model converts words in natural language into dense vectors, and semantically similar words will have similar vector representations. The latest pre-trained model BERT (Devlin et al., 2018) can solve the problem of polysemy, generating different word vectors for the same word according to the context, which can better express the semantic features of the words. This situation often occurs in the cybersecurity corpus. For a piece of software, when describing the vulnerabilities that exist in that software, this entity should then be recognized as a “*Software*” type, and the triple (“*Software*”, “*hasVulnerability*”, “*Vulnerability*”) can be extracted. In another context, the software may be used as a tool by an attacker, at which point the entity should be recognized as a “*Tool*” type, and the triple (“*Threat Actor*”, “*uses*”, “*Tool*”) can be extracted. So, we use the BERT model to generate word embedding vectors in the embedding layer. For the input sentence $X = \{x_1, x_2, x_3, \dots, x_n\}$, where x_t is the tth word in the sentence. After the computation of the BERT pre-trained model, the word embedding vector $E = \{e_1, e_2, e_3, \dots, e_n\}$ of the sentence is generated, where e_t is the word vector of the tth word in the sentence.

Encoder Compared with the traditional recurrent neural network (RNN), GRU consists of an update gate and a reset gate, which can alleviate the gradient disappearance or explosion problem that occurs during training. The GRU hidden state h_t is generated by the previous hidden state h_{t-1} and the input e_t of the current state together. The GRU only calculates the correlation between time step t and the previous time step. However, in the cybersecurity corpus, entities may constitute relations with the entities before or after. So, for the word vectors generated by the embedding layer, we further extract the semantic features of the sentences $H = \{h_1, h_2, h_3, \dots, h_n\}$ using BiGRU and then concatenate the forward and backward GRU hidden states as the contextual word representation. The transformations are as follows:

$$h_t = \left[\overrightarrow{GRU}(e_t), \overleftarrow{GRU}(e_t)\right] \tag{1}$$

Attention mechanism In the cybersecurity corpus, a sentence usually contains many entities and complex relations. As shown in Fig. 2, the sentence contains five different entities (“*Vulnerability*”, “*Software*”, “*Threat Actor*”, “*Campaign*”, “*Attack Pattern*”) and three different relations (“*hasVulnerability*”, “*attributedTo*”, “*uses*”). Therefore, it is necessary to assign different weights to the words in a sentence according to different types of relations. For example, for the “*hasVulnerability*” relation, the words in the sentence indicating a software name or identifying a specific vulnerability should be paid higher attention to. Thus, we have referred to the relation-based attention mechanism proposed by Yuan et al. (2020). The attention mechanism can assign different weights to the words in a sentence under each relation, and the attention score can be calculated as follows:

$$h_g = avg\left\{h_1, h_2, h_3, \dots, h_n\right\} \tag{2}$$

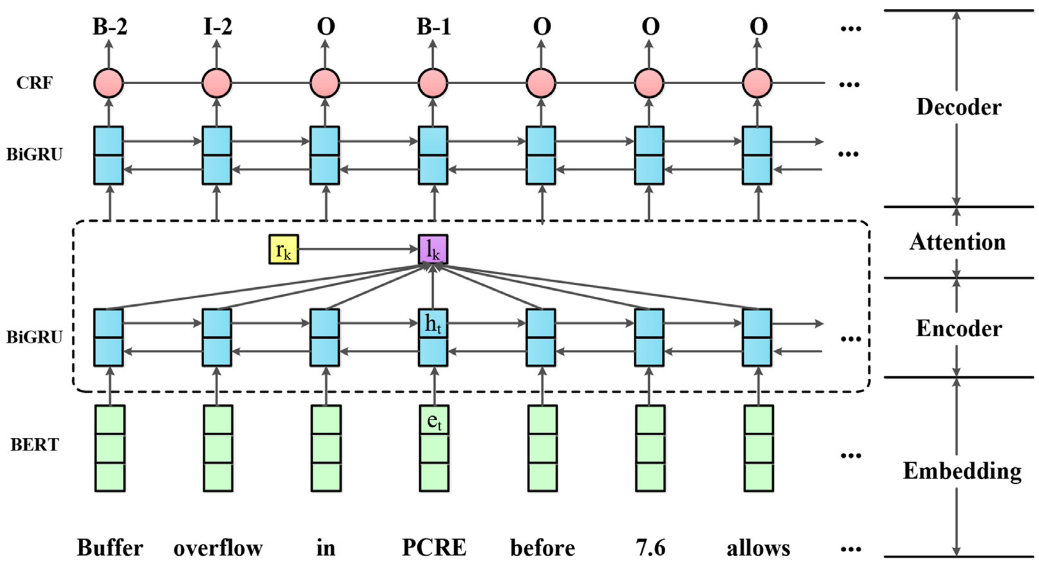


Fig. 3. The multiple sequence labeling model for joint entity and relation extraction. It receives the same sentence input and different relation r_k to extract all triples in the sentence. e_t is the BERT embedding of the word, h_t is the hidden vector of time step t, r_k is the trainable embedding of the kth relation, l_k is the attention weights under relation type r_k. Under the given relation r_k (Take *hasVulnerability* for example), the decoder extracts the corresponding entities of r_k to generate triples (*PCRE, hasVulnerability, Buffer overflow*).

$$e_{tk} = v^T \tanh\left(W_r r_k + W_g h_g + W_h h_t\right) \tag{3}$$

$$a_{tk} = \frac{\exp\left(e_{tk}\right)}{\sum_{j=1}^{n} \exp\left(e_{jk}\right)} \tag{4}$$

where h_g indicates the global representation of the sentence, r_k is the embedding of the k-th relation. v, W_r, W_g, and W_h are all trainable parameters. The attention score generated reflects the importance of the sentence's words in the context as well as relational expression in the current relation. The sentence representation l_k under the r_k relation is generated by the weighted sum of the sentence words, which is calculated as shown in Eq. (5). The attention layer combines the generated l_k and the sentence representations output by the encoding layer as input to the decoding layer, as shown in Eq. (6).

$$l_k = \sum_{t=1}^{n} a_{tk} h_t \tag{5}$$

$$h_t^k = h_t \oplus l_k \tag{6}$$

Decoder The decoding layer generates the label sequences of the sentences under the r_k relation and returns the relational triples through the tagging scheme described in Section 3.2.2. We first used another BiGRU to produce sentence representations $H^o = \{h_1^o, h_2^o, h_3^o, \ldots, h_n^o\}$ and generate sequence scores $Z = \{z_1, z_2, z_3, \ldots, z_n\}$using features from the encoding and attention layers. The calculation process is as follows, where W is the parameter:

$$h_t^o = \left[\overrightarrow{GRU}\left(h_t^k\right), \overleftarrow{GRU}\left(h_t^k\right)\right] \tag{7}$$

$$z_t = W h_t^o \tag{8}$$

Next, the sequence is decoded by the CRF layer, which is able to obtain constrained rules from the training data, to ensure that the predicted cybersecurity entity labels are valid. The decoding process is shown as follows:

$$score(Z, y) = \sum_{t=0}^{n} A_{y_t, y_{t+1}} + \sum_{t=1}^{n} Z_{t, y_t} \tag{9}$$

$$p(y \mid Z) = \frac{\exp(score(Z, y))}{\sum_{y' \in Y_Z} \exp\left(score(Z, y')\right)} \tag{10}$$

$$y^* = \arg\max_{y \in Y_Z} score(Z, y) \tag{11}$$

where A is the transition matrix between labels, $score(Z, y)$ is the position score, and $p(y \mid Z)$ is the normalized probability function. Finally, the label sequence y^* is generated.

3.3. Knowledge fusion based on improved levenshtein distance

Knowledge fusion includes ontology-level fusion and entity-level fusion. The ontology-level is only for structured data, and most structured data exists in STIX format with equivalence classes and relations in the UCO ontology. This section focuses on the entity-level, which aims to determine whether entities from different data sources are the same object in reality.

The knowledge graph constructed from a single data source has problems such as low information coverage and imperfect entity attributes, which are not desirable for further application on downstream tasks. Entity-level fusion can fuse threat intelligence knowledge from different sources to form a more complete CKG. Existing methods determine whether two entities are the same object by the features extracted from the entities, attributes, and relations. In addition to similarity-based approaches (Azevedo et al., 2019; Lacoste-Julien et al., 2013), the mainstream approach in the general domain is embedding-based approaches (Bordes et al., 2013; Chen et al., 2016). However, since our framework aims to construct a preliminary CKG from scratch and mainly addresses threat intelligence extraction, the extracted entities often have incomplete relations. Embedding-based methods generally perform knowledge fusion on the mature knowledge graph, so they are not suitable in the early stage of building CKG. Piplai et al. (2020b) used the Levenshtein distance to calculate the similarity of entity names to perform entity fusion when constructing a knowledge graph for malware after action reports. Although this method is effective for entity fusion with name distortion, misspellings, etc., there is a drawback that can lead to many incorrect results. For example, “APT28” and “APT29” are two different “*Threat Actor*” entities, but their Levenshtein distance is only 1, which causes the entity fusion algorithm to consider them as the same entity mistakenly. There are many other similar cases, for example, two IP addresses with only one digit difference.

In order to perform knowledge fusion as accurately as possible, we propose a knowledge fusion method based on an improved Levenshtein distance for the characteristics of the cybersecurity corpus. Compared to embedding-based knowledge fusion methods, this algorithm does not need to store entity embeddings and model parameters. We find that the above errors are mainly caused by numbers which usually represent deterministic information in entity naming. The Levenshtein distance is the minimum number of edit operations required to turn string a into string b. Editing operations include replace, insert and delete. In our improved Levenshtein distance, a larger penalty weight of w_{num} is set for edit operations on numbers, and edit operations on other characters remain a weight of w_{other}, where w_{num} should be larger than the threshold, which avoids the errors caused by the traditional Levenshtein distance. We use the dynamic programming algorithm to calculate the improved Levenshtein distance as follows. For strings a and b with lengths n and m respectively, i and j are the subscripts indicating the position of the corresponding strings. First, the distance matrix is initialized as shown in Eq. (12). Then the distance of each position of the two strings is calculated iteratively as shown in Eq. (13). In addition, the weight function is shown in Eq. (14).

$$\begin{aligned} &D(0,0)=0 \\ &D(i,0)=D(i-1,0)+W(a_i) \qquad (1 \le i \le n) \\ &D(0,j)=D(0,j-1)+W(b_j) \qquad (1 \le j \le m) \end{aligned} \tag{12}$$

$$D(i,j)=min\begin{cases} D(i-1,j)+W(a_i) \\ D(i,j-1)+W(b_j) \\ D(i-1,j-1)+\max(W(a_i),W(b_j)) & (a_i \neq b_j) \\ D(i-1,j-1) & (a_i = b_j) \end{cases} \tag{13}$$

$$W(char)=\begin{cases} w_{num} & \text{if char is a number} \\ w_{other} & \text{otherwise} \end{cases} \tag{14}$$

With the final result $D(n,m)$, we set a distance threshold and add the "*sameAs*" relation between these two entities if their Levenshtein distance is less than the threshold.

4. Experiments

4.1. Datasets

Our data is collected from publicly available open-source intelligence (OSINT) data. It includes both structured and unstructured data. The unstructured data comes from the CVE vulnerability database (vulnerability descriptions) (MITRE, 2021), security bulletins and APT reports (CyberMonitor, 2021). Structured data comes from STIX format data provided by ATT&CK Strom et al. (2018), Unit 42 (2021), MISP (2021) and WatcherLab (2021), etc. To train the joint extraction model, we used the BRAT annotation platform (Stenetorp et al., 2012) to annotate the unstructured data, and further checked the annotation of the dataset based on our previous work (Guo et al., 2021) to improve the annotation accuracy. We manually annotated 12,680 sentences containing a total of 67,918 cybersecurity triples. Then, we transform the annotated labels into the format described in Section 3.2.2 for futher training. In this paper, we take the "*Threat Actor*" entity as an example to validate the knowledge fusion algorithm. The knowledge fusion dataset we constructed is all used to test the performance of the algorithm. We obtained a total of 698 "*Threat Actor*" entities from different sources of STIX data and manually annotated 560 "*sameAs*" relations between them.

Table 1
Comparison results with the pipeline model.

Models	Precision	Recall	F1-score
Pipeline model (Pingle et al., 2019)	57.04%	67.80%	61.69%
Joint model	**82.28**%	**80.48**%	**81.37**%

Table 2
Comparison results with the joint model.

Models	Precision	Recall	F1-score
NovelTag (Zheng et al., 2017)	54.34%	57.87%	56.05%
GraphRel (Fu et al., 2019)	65.26%	58.30%	61.59%
MultiHead (Bekoulis et al., 2018)	77.76%	65.14%	70.89%
Our Model	**82.28**%	**80.48**%	**81.37**%

4.2. Evaluation metrics

We use standard Precision, Recall, and F1-score to measure the performance. A triple is considered to be correctly extracted if and only if its relation type and both entities are correctly matched.

4.3. Experimental settings

To evaluate the effectiveness of the joint extraction model, we design a set of experiments. In the comparison experiments with the pipeline approach, we compare our model with the existing pipeline model RelExt (Pingle et al., 2019), where parameters that are not mentioned in the paper are set by default. Because an entity may consist of many words, we generate word vectors for each word and average these word vectors to obtain the fixed dimensional embedding described in the RelExt paper. Besides, to further analyze our proposed model, we compare the preprocessing methods, the word embedding models and the choice of neural networks. Also to test whether the model is effective on a smaller training set, we experimented with different training set division ratios. We use a 5-fold cross-validation to train the model. The size of the BERT word vector is 768 dimensions. The size of the BiGRU hidden layer and relational embedding vector are both set to 300. During the training process, the optimizer is RMSprop, the learning rate is 0.0001, and the batch_size is 64. we use the dropout mechanism to avoid overfitting with a rate of 0.5.

4.4. Joint entity and relation extraction experimental result

4.4.1. Comparison with pipeline model

This section shows the results of the comparison between the traditional pipeline approach and the joint model. From Table 1, we can see that our joint model outperforms the pipeline model, significantly improving precision (82.28%), recall (80.48%), and F1-score (81.37%). This indicates that the joint model extracts both entities and relations, which avoids the error propagation between the two subtasks of the pipeline model and effectively improves the performance of entity-relation triples extraction.

4.4.2. Comparison with joint model

To corroborate the efficacy of our proposed model, we benchmark it against established joint models (Bekoulis et al., 2018; Fu et al., 2019; Zheng et al., 2017). As presented in Table 2, our model displays a remarkable improvement over the compared models. This superior performance can be attributed to a meticulously preprocessing procedure, an effective word embedding approach, and a fitting tagging scheme tailored for the cybersecurity corpus. For a more thorough exploration and understanding of its capabilities, we have carried out an array of detailed experiments, the results of which are discussed in Section 4.4.3.

Table 3
Analysis results of the proposed model.

Models	Precision	Recall	F1-score
Our Model	**82.28**%	**80.48**%	**81.37**%
Model-NoSub	79.55%	78.18%	78.86%
Model-W2V	79.41%	71.73%	75.37%
Model-LSTM	82.06%	79.70%	80.86%
Model-NoAtt	81.71%	79.36%	80.52%

Table 4
Experimental results for different dataset splitting.

Train-test split	Precision	Recall	F1-score
80%-20%	82.28%	80.48%	81.37%
75%-25%	82.23%	79.87%	81.03%
66%-34%	81.20%	79.16%	80.17%
50%-50%	81.16%	78.69%	79.90%

4.4.3. Analysis of the proposed model

In the preprocessing approach experiments, as described in Section 3.2.1, we replaced special entities such as IP, MAC, Hash, URL, Email, domain name, file name and file path. We experimented with the effect of not using this preprocessing method. As shown in Table 3 (“Model-NoSub”), the preprocessing approach of special information replacement is more effective, because special information replacement allows these entities with large differences to be replaced with a unified expression form, which is beneficial to the extraction of semantics.

In the word vector experiments, as the word vectors generated by the word embedding model serve as the input to the following model, the quality of the word vectors has an important impact on the model performance. In this section, we experiment with two representative word embedding models, BERT (Devlin et al., 2018) and Word2Vec (Mikolov et al., 2013), where the BERT model is the “cased L-12 H768 A-12” version, and the Word2Vec model is the “GoogleNews- vectors-negative300” version. It can be seen from Table 3 (“Model-W2V”) that using BERT for word embedding has a certain improvement compared to Word2Vec. This is attributed to the fact that BERT can generate different word vectors for the same word depending on the context thus making better use of the contextual information of the text, while Word2Vec can only generate a fixed word vector representation for each word.

In the neural network model experiments, since we use neural networks in our model for the sequence labeling task, we investigate the effect of different neural networks on the model performance. Specifically, we experiment with the performance of LSTM and GRU neural networks in the joint extraction model. As shown in Table 3 (“Model-LSTM”), we found that the GRU performed slightly better than the LSTM, and therefore we take GRU in our model.

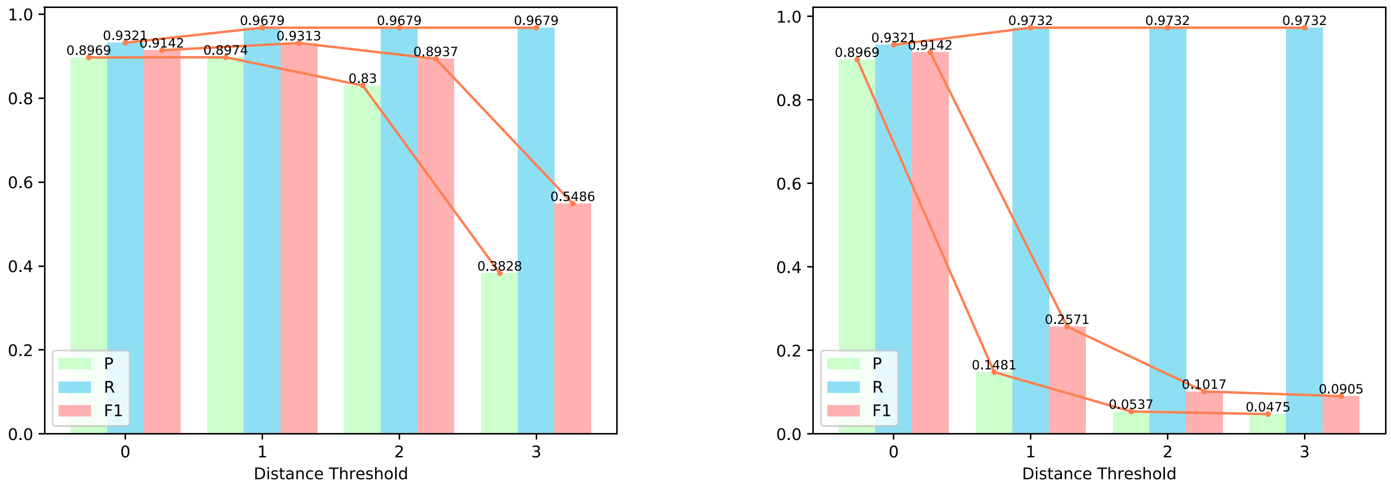
To demonstrate the role played by the attention mechanism in our model, we further explored the effect of not using the attention mechanism. From Table 3 (“Model-NoAtt”), we can tell that the attention mechanism slightly improve the performance by providing a better reflection on the importance of the tokens under the specific relation.

4.4.4. Experimental results with different dataset splitting

In this section, we conduct experiments using different dataset splitting, and the results show that the effectiveness of our model does not drop significantly on a smaller training set ratio, as shown in Table 4. This is because the cybersecurity corpus has some similarities in writing style, syntax, and use of terminology. For example, CVE vulnerability descriptions use similar syntax, and different APT reports may use the same terminology. In addition, we replace special entities with unified forms of expression in the preprocessing stage, which also mitigates the problem of high variability between data from different sources.

4.5. Knowledge fusion experimental results

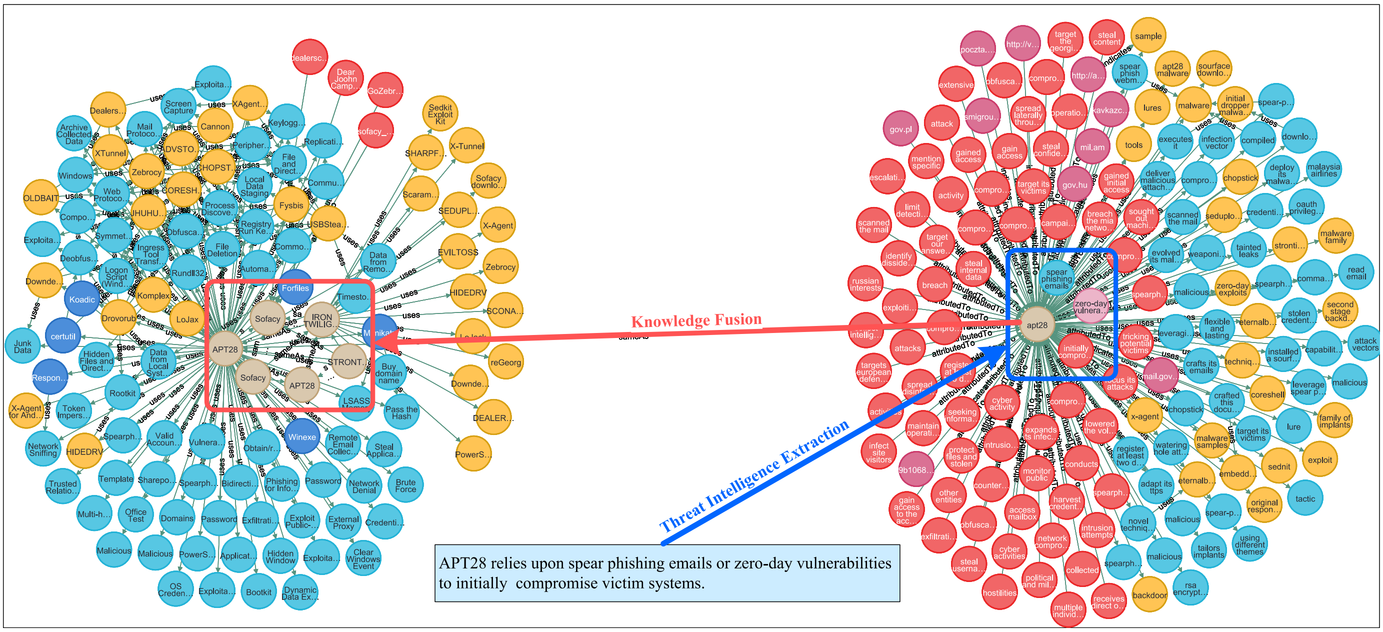
In this section, we take the “*Threat Actor*” entity as an example. We set the distance threshold to 0, 1, 2, and 3 for each experiment. The w_{num} is set to 10 and the w_{other} is set to 1. If the distance between two entities is less than the threshold, we consider these two entities are the same entity. As shown in Fig. 4, the Improved Levenshtein distance (Fig. 4a) outperforms the traditional Levenshtein distance (Fig. 4b). The performance of the traditional Levenshtein distance algorithm decreases significantly as the threshold increases. This is because there are numbers involved in cybersecurity entity names (as described in Section 3.3), causing the traditional Levenshtein distance to produce a large number of erroneous results. In addition, as shown in Fig. 4a, the algorithm is bound to introduce some false alarms as the threshold increases, so the threshold cannot be set too large, and it can be seen that the best results are obtained when the threshold is set to 1. Although our algorithm does not consider more complex knowledge



(a) Improved Levenshtein distance.

(b) Traditional Levenshtein distance.

Fig. 4. Experimental results of knowledge fusion.



APT28 relies upon spear phishing emails or zero-day vulnerabilities to initially compromise victim systems.

Fig. 5. The effect of knowledge fusion.

fusion cases, the algorithm can more accurately fuse entities in the process of constructing the CKG from scratch for naming distortions, misspellings, etc.

4.6. Analysis and discussion

4.6.1. Relation extraction case study

In this section, we illustrate the advantages of the joint model over the pipeline model by two examples, as shown in Appendix Table A1. In both examples, our proposed joint model predicts all the triples in the sentences correctly.

For Case 1, although the pipeline model correctly predicts all "*Software*" entities in the entity recognition task, when predicting the relation between two "*Software*" entities, the model will combine the two "*Software*" entities into two entity pairs in different orders and predict two wrong relations. This indicates that the pipeline model does not take into account the connection between entity recognition and relation extraction tasks, while the joint model is able to predict the "*hasProduct*" relation between the two "*Software*" entities well.

For Case 2, the pipeline model only recognizes the "*zero-day vulnerabilities/Vulnerability*" and "*compromise victim systems/Campaign*" entity but misses the "*apt28/Threat Actor*" and "*spear phishing emails/Attack Pattern*" entities, resulting in a null input to the relation extraction model that fails to predict the relation between them. This indicates that the pipeline model has the defect of error propagation, implying that if an entity is not predicted or is incorrectly predicted, it will affect the subsequent relation extraction task.

4.6.2. Knowledge fusion effect

This section demonstrates the effect of knowledge fusion. As described in Section 3, the framework extracts cybersecurity triples from structured and unstructured data and performs knowledge fusion. In the constructed knowledge graph, we query the data related to the "*APT28/Threat Actor*", as shown in Fig. 5. The left part is constructed from structured data of various existing attack and defense knowledge bases; the right part is a set of cybersecurity triples extracted from unstructured data. Specifically, in the threat intelligence extraction, we use the joint extraction model to extract entities and relations in sentences. For example, the entities and relations in the blue box of the right part of Fig. 5 are extracted from the sentence at the bottom of the figure. In the knowledge fusion, the framework fuses the extracted entities and the existing entities in the knowledge graph using the algorithm described in Section 3.3. For example, the "*APT28/Threat Actor*" entity on the right part of Fig. 5 is the same object as the "*Threat Actor*" entity shown in the red box on the left part of Fig. 5. Above all, our framework can fuse multi-source threat intelligence data to form a more comprehensive CKG.

5. Conclusion

In this paper, we propose a framework for threat intelligence extraction and fusion, which extracts threat intelligence from structured and unstructured data sources, and fuses threat intelligence from different sources to form a preliminary CKG. For the unstructured data, we propose a joint entity and relation extraction model for cybersecurity concepts, which can extract both entities and relations in the cybersecurity corpus. Specifically, we use a tagging scheme to convert the joint extraction problem into a multi-sequence labeling problem by generating separate label sequences for different relations, which contain information about the related entities and the subject and object of that relation. In addition, the model employs the BERT model, BiGRU neural network, and attention mechanism to extract the features of sentences. For the knowledge fusion, we propose an improved Levenshtein distance to fuse entities pointing to the same object from different data sources. In the experimental section, our results on OSINT data demonstrate that the joint model achieves better results compared to the traditional pipeline approach. And the improved knowledge fusion also works better than the traditional Levenshtein distance. The limitation of this study is that our framework is mainly for threat intelligence extraction and simple knowledge fusion to construct a preliminary CKG. Therefore, to further construct a more effective CKG, our future work will focus on entity disambiguation, knowledge embedding, and knowledge inference in cybersecurity.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Yongyan Guo: Conceptualization, Data curation, Methodology, Software, Writing – original draft. **Zhengyu Liu:** Conceptualization, Data curation, Methodology, Software, Writing – original draft. **Cheng Huang:** Conceptualization, Methodology, Validation, Writing – review & editing. **Nannan Wang:** Data curation, Investigation, Software. **Hai Min:** Data curation, Investigation, Software. **Wenbo Guo:** Data curation, Investigation, Software. **Jiayong Liu:** Conceptualization, Investigation, Methodology.

Data availability

Data will be made available on request.

Appendix A

Table A1
The examples of the triples to the given sentences extracted by joint model and pipeline model.

#Case 1	
Raw text	CVE-2021-28967: The unofficial MATLAB extension before 2.0.1 for Visual Studio Code allows attackers to execute arbitrary code via a crafted workspace because of lint configuration settings.
Joint model	(’attackers’, ’crafted workspace’, ’Threat Actor/Attack Pattern/uses’) (’attackers’, ’cve-2021-28967’, ’Threat Actor/Vulnerability/uses’) (’execute arbitrary code’, ’attackers’, ’Campaign/Threat Actor/attributedTo’) (’visual studio code’, ’unofficial matlab extension’, ’Software/Software/hasProduct’) (’unofficial matlab extension’, ’cve-2021-28967’, ’Software/Vulnerability/hasVulnerablity’) (’visual studio code’, ’cve-2021-28967’, ’Software/Vulnerability/hasVulnerablity’)
Pipeline model	(’attackers’, ’crafted workspace’, ’Threat Actor/Attack Pattern/uses’) (’attackers’, ’cve-2021-28967’, ’Threat Actor/Vulnerability/uses’) (’unofficial matlab extension’, ’cve-2021-28967’, ’Software/Vulnerability/hasVulnerablity’) (’visual studio code’, ’cve-2021-28967’, ’Software/Vulnerability/hasVulnerablity’) (’execute arbitrary code’, ’attackers’, ’Campaign/Threat Actor/attributedTo’) (’visual studio code’, ’unofficial matlab extension’, ’Software/Software/uses’) (’unofficial matlab extension’, ’visual studio code’, ’Software/Software/hasVulnerablity’)
#Case 2	
Raw text	APT28 relies upon spear phishing emails or zero-day vulnerabilities to initially compromise victim systems.
Joint model	(’apt28’, ’spear phishing emails’, ’Threat Actor/Attack Pattern/uses’) (’apt28’, ’zero-day vulnerabilities’, ’Threat Actor/Vulnerability/uses’) (’initially compromise victim systems’, ’apt28’, ’Campaign/Threat Actor/attributedTo’)
Pipeline model	Only entities found zero-day vulnerabilities/Vulnerability compromise victim systems/Campaign

References

Alves, F., Andongabo, A., Gashi, I., Ferreira, P.M., Bessani, A., 2020. Follow the blue bird: a study on threat data published on twitter. In: European Symposium on Research in Computer Security. Springer, pp. 217–236.

Azevedo, R., Medeiros, I., Bessani, A., 2019. Pure: generating quality threat intelligence by clustering and correlating OSINT. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, pp. 483–490.

Bekoulis, G., Deleu, J., Demeester, T., Develder, C., 2018. Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst. Appl. 114, 34–45.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, pp. 2787–2795.

Chen, M., Tian, Y., Yang, M., Zaniolo, C., 2016. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. arXiv preprint arXiv:1611.03954

Culotta, A., Sorensen, J., 2004. Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pp. 423–429.

CyberMonitor, 2021. APT & cybercriminals campaign collection. https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections.

Dai, D., Xiao, X., Lyu, Y., Dou, S., She, Q., Wang, H., 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6300–6308.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Fu, T.-J., Li, P.-H., Ma, W.-Y., 2019. Graphrel: modeling text as relational graphs for joint entity and relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1409–1418.

Gao, P., Shao, F., Liu, X., Xiao, X., Qin, Z., Xu, F., Mittal, P., Kulkarni, S.R., Song, D., 2021. Enabling efficient cyber threat hunting with cyber threat intelligence. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, pp. 193–204.

Ghazi, Y., Anwar, Z., Mumtaz, R., Saleem, S., Tahir, A., 2018. A supervised machine learning based approach for automatically extracting high-level threat intelligence from unstructured sources. In: 2018 International Conference on Frontiers of Information Technology (FIT). IEEE, pp. 129–134.

Gonzalez-Granadillo, G., Faiella, M., Medeiros, I., Azevedo, R., Gonzalez-Zarzosa, S., 2021. ETIP: an enriched threat intelligence platform for improving OSINT correlation, analysis, visualization and sharing capabilities. J. Inf. Secur. Appl. 58, 102715.

Guo, Y., Liu, Z., Huang, C., Liu, J., Jing, W., Wang, Z., Wang, Y., 2021. Cyberrel: joint entity and relation extraction for cybersecurity concepts. In: International Conference on Information and Communications Security. Springer, pp. 447–463.

Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., Niu, X., 2017. Ttpdrill: automatic and accurate extraction of threat actions from unstructured text of CTI sources. In: Proceedings of the 33rd Annual Computer Security Applications Conference, pp. 103–115.

Husari, G., Niu, X., Chu, B., Al-Shaer, E., 2018. Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 1–6.

Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., Ferragut, E., Goodall, J., 2015. Developing an ontology for cyber security knowledge graphs. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference, pp. 1–4.

Iria, J., 2005. T-rex: a flexible relation extraction framework. In: Proceedings of the 8th Annual Colloquium for the UK Special Interest Group for Computational Linguistics (CLUK’05), vol. 6. Citeseer, p. 9.

Jia, Y., Qi, Y., Shang, H., Jiang, R., Li, A., 2018. A practical approach to constructing a knowledge graph for cybersecurity. Engineering 4 (1), 53–60.

Jiang, J., Zhai, C., 2007. A systematic exploration of the feature space for relation extraction. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 113–120.

Jones, C.L., Bridges, R.A., Huffer, K.M., Goodall, J.R., 2015. Towards a relation extraction framework for cyber-security concepts. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference, pp. 1–4.

Lacoste-Julien, S., Palla, K., Davies, A., Kasneci, G., Graepel, T., Ghahramani, Z., 2013. Sigma: simple greedy matching for aligning large knowledge bases. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 572–580.

Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R., 2016. Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 755–766.

Liu, Z., Cao, Y., Pan, L., Li, J., Chua, T.-S., 2020. Exploring and evaluating attributes, values, and structures for entity alignment. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6355–6364.

Liu, Z., Su, H., Wang, N., Huang, C., 2022. Coreference resolution for cybersecurity entity: towards explicit, comprehensive cybersecurity knowledge graph with low redundancy. In: International Conference on Security and Privacy in Communication Systems. Springer, pp. 89–108.

McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., White, P., 2005. Simple algorithms for complex relation extraction with applications to biomedical IE. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pp. 491–498.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

Milajerdi, S.M., Gjomemo, R., Eshete, B., Sekar, R., Venkatakrishnan, V., 2019. Holmes: real-time apt detection through correlation of suspicious information flows. In: 2019 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 1137–1152.

MITRE, 2021. Cvelist project. https://github.com/CVEProject/cvelist.

Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T., 2016. Cybertwitter: using twitter to generate alerts for cybersecurity threats and vulnerabilities. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, pp. 860–867.

Miwa, M., Bansal, M., 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1105–1116.

MISP project, 2021. MISP threat sharing platform. https://www.misp-project.org/galaxy.html.

Nie, H., Han, X., Sun, L., Wong, C.M., Chen, Q., Wu, S., Zhang, W., 2021. Global structure and local semantics-preserved embeddings for entity alignment. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 3658–3664.

Pingle, A., Piplai, A., Mittal, S., Joshi, A., Holt, J., Zak, R., 2019. Relext: relation extraction using deep learning approaches for cybersecurity knowledge graph improvement. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 879–886.

Piplai, A., Mittal, S., Abdelsalam, M., Gupta, M., Joshi, A., Finin, T., 2020. Knowledge enrichment by fusing representations for malware threat intelligence and behavior. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 1–6.

Piplai, A., Mittal, S., Joshi, A., Finin, T., Holt, J., Zak, R., 2020. Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access 8, 211691–211703.

Satyapanich, T., Ferraro, F., Finin, T., 2020. CASIE: extracting cybersecurity event information from text. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8749–8757.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J., 2012. Brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102–107.

STIX. Structured threat information expression. https://oasis-open.github.io/cti-documentation/stix/intro.

Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B., 2018. MITRE ATT&CK: Design and Philosophy. Mitre Product MP. 18–0944

Sun, C., Wu, Y., Lan, M., Sun, S., Wang, W., Lee, K.-C., Wu, K., 2018. Extracting entities and relations with joint minimum risk training. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2256–2265.

Syed, Z., Padia, A., Mathews, M.L., Finin, T., Joshi, A., et al., 2016. UCO: a unified cybersecurity ontology. In: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security.

Unit 42. 2021. Unit 42 playbook viewer. https://pan-unit42.github.io/playbook_viewer/.

WatcherLab, 2021. Threat intelligence feed system. https://feed.watcherlab.com.

Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y., 2020. A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1476–1488.

Yuan, L., Bai, Y., Xing, Z., Chen, S., Li, X., Deng, Z., 2021. Predicting entity relations across different security databases by using graph attention network. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, pp. 834–843.

Yuan, Y., Zhou, X., Pan, S., Zhu, Q., Song, Z., Guo, L., 2020. A relation-specific attention network for joint entity and relation extraction. In: International Joint Conference on Artificial Intelligence 2020. Association for the Advancement of Artificial Intelligence (AAAI), pp. 4054–4060.

Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., 2014. Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, pp. 2335–2344.

Zhao, J., Yan, Q., Li, J., Shao, M., He, Z., Li, B., 2020. Timiner: automatically extracting and analyzing categorized cyber threat intelligence from social data. Comput. Secur. 95, 101867.

Zhao, J., Yan, Q., Liu, X., Li, B., Zuo, G., 2020. Cyber threat intelligence modeling based on heterogeneous graph convolutional network. In: 23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020), pp. 241–256.

Zhao, X., Zeng, W., Tang, J., Wang, W., Suchanek, F., 2020. An experimental study of state-of-the-art entity alignment approaches. IEEE Trans. Knowl. Data Eng. 34 (1), 1.

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B., 2017. Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1227–1236.

Zhu, Z., Dumitras, T., 2018. Chainsmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp. 458–472.

Yongyan Guo received the B.Eng. degree in cyberspace security from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the masters degree with the School of Cyber Science and Engineering. His current research interests include Web security and artificial intelligence.

Zhengyu Liu is currently an undergraduate student studying at the School of Cyber Science and Engineering, Sichuan University. His current research interests focus on data-driven security, especially in applying machine learning with empirical data measurements to solve cybersecurity problems, including cyber threat modeling, cyberattacks detection, etc.

Cheng Huang received the Ph.D degree from Sichuan University, Chengdu, China, in 2017. From 2014 to 2015, he was a visiting student at the School of Computer Science, University of California, CA, USA. He is currently an Associate Professor at the School of Cyber Science and Engineering, Sichuan University, Chengdu, China. His current research interests include Web security, attack detection, artificial intelligence.

Nannan Wang is a prospective graduate student in the School of Cyber Science and Engineering, Sichuan University, China. His current main research areas include artificial intelligence, data mining and attack detection.

Hai Min is currently pursuing his bachelor's degree in the School of Cyber Science and Engineering, Sichuan University, China. His current research interests include fuzzing test, firmware security and artificial intelligence.

Wenbo Guo received the B.Eng. degree in cyberspace security from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the masters degree with the School of Cyber Science and Engineering. His current research interests include Web security and artificial intelligence.

Jiayong Liu received his B.Eng. degree in 1982, M. Eng. degree in 1989, and Ph.D. degree in 2008 from Sichuan University, China. He is currently a professor in School of Cyber Science and Engineering, Sichuan University, China. His research interests include network information processing and information security, communications and network information system.