Gao *et al. Cybersecurity* (2021) 4:9
https://doi.org/10.1186/s42400-021-00072-y

Cybersecurity

RESEARCH Open Access

Data and knowledge-driven named entity recognition for cyber security

Chen Gao[1], Xuan Zhang[1,2,3*] and Hui Liu[1]

Abstract

Named Entity Recognition (NER) for cyber security aims to identify and classify cyber security terms from a large number of heterogeneous multisource cyber security texts. In the field of machine learning, deep neural networks automatically learn text features from a large number of datasets, but this data-driven method usually lacks the ability to deal with rare entities. Gasmi et al. proposed a deep learning method for named entity recognition in the field of cyber security, and achieved good results, reaching an F1 value of 82.8%. But it is difficult to accurately identify rare entities and complex words in the text.To cope with this challenge, this paper proposes a new model that combines data-driven deep learning methods with knowledge-driven dictionary methods to build dictionary features to assist in rare entity recognition. In addition, based on the data-driven deep learning model, an attention mechanism is adopted to enrich the local features of the text, better models the context, and improves the recognition effect of complex entities. Experimental results show that our method is better than the baseline model. Our model is more effective in identifying cyber security entities. The Precision, Recall and F1 value reached 90.19%, 86.60% and 88.36% respectively.

Keywords: Cyber security, Named entity recognition, Attention mechanism, Dictionary, Deep learning

Introduction

There is a large amount of unstructured cyber security data on the Internet, which is difficult to be directly identified and utilized by the cyber security system. These data usually come from cyber security blogs, company communities, and related databases, such as Common Vulnerabilities and Exposures (CVE) and National Vulnerability Database (NVD). Automatically mining cyber security information from these data and building a cyber security knowledge base can provide basic support for resisting cyber threats. Therefore, a method for cyber security NER is proposed.

At present, the model based on deep learning (LeCun et al. 2015) has become the main method in the field of NER. Compared with the traditional rule-based method or statistical learning method, it trains the model by automatically mining the features in the text through the neural network, which is independent of the field and does not require expert knowledge for feature engineering. Although great success has been achieved through deep learning methods, due to the particularity of the cyber security field, it is difficult to accurately identify cyber security entities only relying on data-driven deep learning methods. First of all, deep neural networks usually adopt an end-to-end method and try to learn features directly from large-scale labelled data. This method rarely combines with human knowledge. In cyber security texts, there are a large number of non-standard abbreviations or acronyms, and these entities appear little or not at all in the training set. Data-driven deep learning methods are usually difficult to handle this situation well. Besides, cyber security text contains a large number of long sentences. The structure of these sentences is often complicated. It is difficult to accurately extract its features by relying solely on neural networks.

In response to the above problems, we propose a data and knowledge-driven cyber security NER method. This method combines the input text with pre-trained word

*Correspondence: zhxuan@ynu.edu.cn
[1]School of Software, Yunnan University, 650091 Yunnan, China
[2]Key Laboratory of Software Engineering of Yunnan Province, 650091 Yunnan, China
Full list of author information is available at the end of the article

embedding vectors in the field of cyber security. Based on the BiLSTM-CRF sequence annotation model, a multi-head self-attention mechanism is employed to capture contextual information from multiple different subspaces, which can better understand sentence structure. At the same time, combined with an external dictionary knowledge and a neural network. The dictionary contains public entities and rare entities, which can better handle rare named entities. We evaluate the performance of our model on a public dataset. Experimental results show that the proposed model is better than the baseline model, with an F1 value of 88.36%.

Related work

Cyber security entities mainly refer to security terms, systems and software names etc. Early cyber security NER mainly used statistical learning methods. These methods treated cyber security entities as a sequence labelling problem, that is, to find the most suitable label sequence for a given input sentence. Based on support vector machines, (Mulwad et al. 2011) described a prototype framework to detect and extract information about vulnerabilities and attacks from Web texts. Lal (2013) used the Stanford NER conditional random fields model to develop a system that automatically extracted terms from cyber security blogs and bulletins. Joshi et al. (2013) also developed an information extraction conditional random fields framework to extract cyber security entities, terms and concepts from the NVD database and unstructured texts. Based on (Joshi et al. 2013) work, (Weerawardhana et al. 2014) used statistical machine learning methods and rule dictionaries to conduct NER experiments to identify entities which were embedded in vulnerability description texts.

Most of the above systems rely on statistical machine learning methods. On the basis of a large number of labelled corpora, constructing feature engineering, mining and analysing the language information in the training corpus not only requires a lot of manpower and material resources, but also is low efficiency. After the emergence of deep learning, lots of researchers have begun to use deep learning-based methods for named entity recognition research to avoid tedious feature engineering. Collobert et al. (2011) proposed an effective neural network model that can learn word vectors from a large amount of unlabelled texts. Huang et al. (2015) proposed the BiLSTM-CRF model, which is a combination of neural network and statistical machine learning methods.

In their model, BiLSTM can effectively use the context of words and the CRF can use sentence-level tag sequence information. Subsequently, BiLSTM-CRF model became the mainstream model (Gasmi et al. 2018; Mazharov and Dobrov 2018). Simran et al. (2019) evaluated several deep learning architectures using gated recurrent unit as the basic model. However, due to the complexity of text in cyber security, it is difficult to accurately identify entities in cyber security by using neural network alone. Qin et al. (2019) combined deep learning methods and feature templates to improve the mixed Chinese and English identification of cyber security entities. Qin et al. (2019) and (Liu 2020) focused on improving the connection between word vector and character vector to identify key terms in network security documents. Li et al. (2019) and (Zhang et al. 2019) employed the attention mechanism and antagonistic learning mechanism in network security text recognition. Zhou et al. (2018) also employed the attention mechanism and feature templates to strengthen the recognition of low-frequency vocabulary in cyber security texts. Wu et al. (2020) combined the neural network model with the domain dictionary to improve cyber security entity recognition by constructing the domain dictionary for correction. Dionísio et al. (2019) and (Gu et al. 2018) used supervised learning to propose entity recognition methods for cyber security essay. Xiao (2017) proposed an information extraction system based on unsupervised learning to locate and classify the concept of network security in unstructured texts. (Wang et al. 2020) proposed a new loss function TSFL, a triple loss function based on metric learning and classification, to solve the problem of unbalanced data label distribution.To better encode the context, (Tikhomirov et al. 2020) proposed a new model called RuBERT, which uses BERT coding as the basic structure, pre-trained a large number of Russian-related corpora, and significantly improved the performance of the three NLP tasks in Russian.

All above models have improved the effect of cyber security entity recognition to a certain extent, but they did not combine domain knowledge and data-driven deep learning methods well. Moreover, different words in text sentences have varying degrees of influence on the results of NER. It is difficult to capture long-distance interdependent features in sentences with a single neural network structure. Therefore, the recognition effect needs to be further improved. Aiming at the shortcomings of current methods, we propose a deep learning model that combines attention mechanism and domain dictionary features. In summary, the contributions of our work are as follows:

- The attention mechanism is employed in our model to strengthen the neural network's ability to process long-distance information.
- We combine domain dictionaries containing human knowledge and data-driven deep learning methods to improve the recognition of rare entities.
- To evaluate the model, an open source cyber security NER dataset was used to meet the experimental requirements. The better experimental results were

obtained that the precision, recall and F1 value are 7.00%, 3.98% and 5.45% higher than the baseline model.

Cyber security Named Entity Recognition Framework

In this paper, referring to the universal NER method, the task of cyber security NER is solved as a sequence labelling problem. The beginning vocabulary of the entity is marked as "B-type", the internal and ending vocabulary of the entity is marked as "I-type", and the vocabulary of non-entity is marked as "O". Figure 1 shows the overall structure of our model. The entire model consists of 5 parts: 1) Input layer; 2) Embedding layer; 3) BiLSTM layer; 4) Attention layer; 5) CRF layer.

The input layer contains text and dictionary feature vectors. The embedding layer obtains the vocabulary semantic information of the text by loading the pre-trained cyber security pre-training word embedding vector. The BiLSTM layer performs global feature extraction and outputs vector of the BiLSTM to the attention layer. The attention layer assigns different weights to different feature vectors in the global features to extract local features. After processing text embedding, external dictionary feature embedding is then loaded. The BiLSTM network is also used to encode features to obtain phrase boundary information. Finally, the joint feature vector sequence including global features, local features and domain dictionary features are inputted into the CRF decoding layer. In the CRF layer, CRF is the named entity recognition model, which is used to predict the global optimal labelling sequence.

Embedding layer

The embedding layer includes word embedding and domain dictionary embedding.

Word embedding

Word embedding takes sentences in the text as input to get its vector representation. Word embedding is a distributed word representation method, which learns the semantic and grammatical information of words from a large amount of unlabelled data. In order to better represent the domain semantic relevance of cyber security,

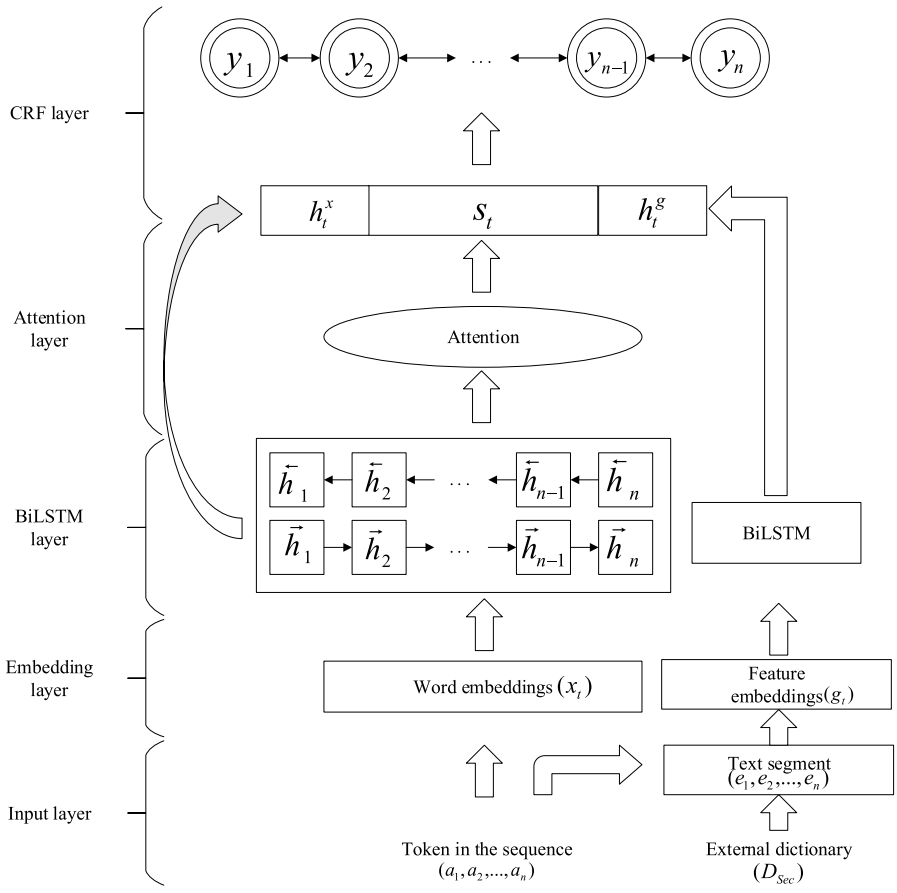

Fig. 1 Cyber security NER Model

we adopt the word embedding vector specific to the cyber security domain of (Roy et al. 2017). Their word embedding vector contains 296,340 and 300,074 unique words for the Malware and CVE datasets respectively. As shown in Fig. 1, given a set of sentence input sequences $a_t(t = 1, 2, \ldots, n)$, the word vector representation is generated through the preloaded token embedding matrix mapping, denoted as $x_t(t = 1, 2, \ldots, n)$, as the input of the BiLSTM coding layer.

Domain dictionary embedding

The domain dictionary embedding is a binary-valued feature vector, which represents the candidate label of a word based on a given dictionary. The value in the feature vector depends on the context and dictionary and is not affected by other sentences or statistical information. Therefore, dictionary feature vectors provide completely different information from statistical methods. We first construct feature vectors for the words in each sentence according to the dictionary and context. The dictionary is constructed at the entity level, but the sequence is marked at the word level. Since an entity often contains multiple words, the entity features need to be represented in word-level tags. We use the N-gram feature representation scheme to indicate whether a word and its surrounding vocabulary group is cyber security named entity.

Given an input sequence $a_t(t = 1, 2, \ldots, n)$ and an external domain dictionary D_{Sec}, we segment the context based on the predefined N-gram feature template to construct a text segment sequence $e_t(t = 1, 2, \ldots, n)$. The feature template we used is shown in Table 1.

For each sentence, a sliding window operation of size N is performed according to the word which forms a text segment sequence of length N. Each text segment that appears in the N-gram feature template can generate a binary value by matching the dictionary to indicate whether the text segment is a cyber security entity in the dictionary D_{Sec}.

However, the N-grams feature template has the problem of out-of-bounds phrase. For the out-of-bounds text segment, we use special tags to replace it. Finally, according to the text segment sequence $e_t(t = 1, 2, \ldots, n)$ of each input sentence, we will get the feature vector containing the entity domain and boundary information, denoted as $g_t(t = 1, 2, \ldots, n)$. Among them, each word corresponds to an 8-dimensional binary value vector representation.

For each sentence in the text, the process of generating feature templates is shown in Fig. 2.

Table 1 Word-based N-gram feature template

Type	Feature template
2-gram	$x_{i-1}x_i, x_ix_{i+1}$
3-gram	$x_{i-2}x_{i-1}x_i, x_ix_{i+1}x_{i+2}$
4-gram	$x_{i-3}x_{i-2}x_{i-1}x_i, x_ix_{i+1}x_{i+2}x_{i+3}$
5-gram	$x_{i-4}x_{i-3}x_{i-2}x_{i-1}x_i, x_ix_{i+1}x_{i+2}x_{i+3}x_{i+4}$

BiLSTM layer

Long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) is a recurrent neural network. It solves the problems of gradient disappearance and gradient explosion in the training process of traditional recurrent neural network (RNN) network structure and better captures long-distance dependencies. Figure 3 shows a typical LSTM unit structure.

The input unit of the LSTM network at time t is composed of the current input x_t and the previous output unit h_{t-1}. The entire structure is regulated by a cell state C. The information in the cell state C is saved and updated according to the three-gate structure.

The forgotten gate f_t controls whether the current content is memorized, and with a certain probability, whether to forget the state of the previous layer of cells.

$$f_t = \sigma\left(W_f \cdot [x_t, h_{t-1}] + b_f\right) \tag{1}$$

Where W_f, b_f are the weight matrix and bias vector of the forgetting gate respectively, represents the *sigmoid* function.

The input gate it is responsible for processing the input of the current position and updating the information in C. The new cell state C_t is obtained by combining the information from the forgotten gate and the input gate.

$$i_t = \sigma\left(W_i \cdot [x_t, h_{t-1}] + b_i\right) \tag{2}$$

$$\widetilde{C}_t = \tanh\left(W_C \cdot [x_t, h_{t-1}] + b_C\right) \tag{3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{4}$$

Where W_i, b_i, W_C , b_C are the weight matrix and bias vector of forgotten gate and cell state respectively. C_{t-1} is the cell state of the previous layer. The output gate o_t controls the information input to the next hidden unit. Performing the *tanh* function operation on the updated cell state C_t, and then multiplying it with the output gate o_t to obtain the final output information h_t.

$$o_t = \sigma\left(W_o \cdot [x_t, h_{t-1}] + b_o\right) \tag{5}$$

$$h_t = o_t \cdot \tanh\left(C_t\right) \tag{6}$$

Where W_o, b_o are the weight matrix and bias vector of the forgotten gate and cell state respectively, h_t is the output vector of the unit at time t.

LSTM can effectively memorize and access the information above the text, but in the task of named entity recognition, the information contained in the text is also important. Therefore, we use two parallel BiLSTM

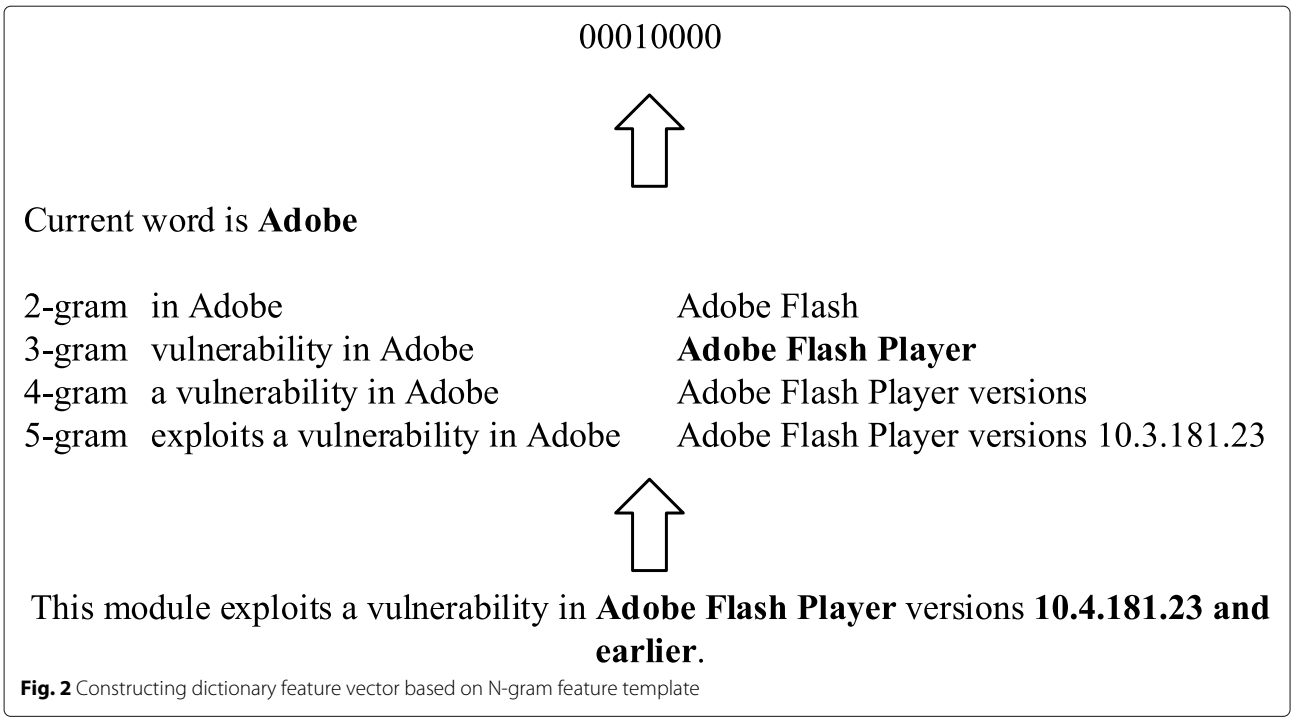

Fig. 2 Constructing dictionary feature vector based on N-gram feature template

networks to extract contextual information and potential entity boundary information. For the sentence $a_t(t = 1, 2, \ldots, n)$, according to the word embedding $x_t(t = 1, 2, \ldots, n)$, the feature vector h_t^x containing context information will be obtained. $g_t(t = 1, 2, \ldots, n)$ will get the feature vector h_t^g, which containing boundary information. The two BiLSTM networks are independent of each other and do not share any parameters. Their hidden state can be defined as follows:

$$h_t^x = \text{BiLSTM}\left(\overrightarrow{h}_{t+1}^{x}, \overleftarrow{h}_{t-1}^{x}, x_t\right) \tag{7}$$

$$h_t^g = \text{BiLSTM}\left(\overrightarrow{h}_{t+1}^{g}, \overleftarrow{h}_{t-1}^{g}, g_t\right) \tag{8}$$

Attention layer

The attention mechanism is a selection mechanism which is used to allocate limited information processing capabilities. It selectively focuses on certain important

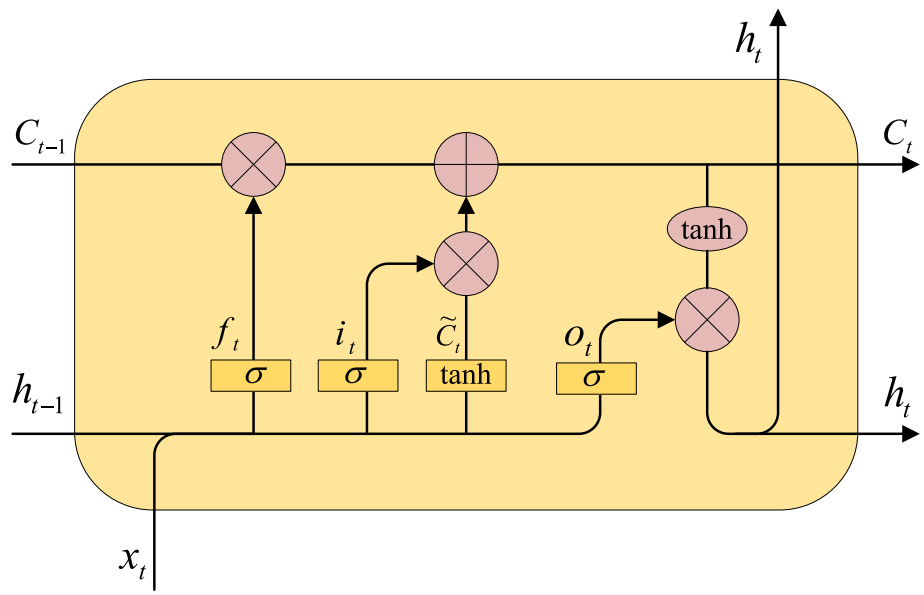

Fig. 3 LSTM unit structure

information and accordingly ignores other information received at the same time. The BiLSTM considers context information in text processing, but does not highlight the role of key information in the context. The attention mechanism assigns higher weights to key information according to the importance of the information in the text, and smaller weights to other information. In addition, the employment of the attention mechanism will make it easier to capture long-distance interdependent features in sentences. Thereby, it will effectively improve the accuracy of entity recognition for cyber security.

The attention mechanism mainly contains three elements: queries Q, keys K and values V, which can be described as a mapping from Q to a series of K-V pairs. As an improved model of the attention mechanism, the self-attention mechanism is mainly used for attention within the sequence to find the connections within the sequence, namely: *Attention (X, X, X)*. In *Attention (X, X, X)*, X refers to the input sequence and the weighted text feature vector is output. In our model, we adopt the scaled dot product attention mechanism. It is essentially an attention mechanism that uses dot product for similarity calculation. Figure 4 shows the calculation method of scaling dot product attention.

As shown in Fig. 4, first performing a MatMul on Q and K, that is, a dot multiplication operation. Scale represents a scaling operation to prevent the inner product of Q and K from being too large. There is an optional Mask process for the operation of Q and K. Then feed the value of the previous step to the softmax function to obtain the weight corresponding to Value. Finally, the weight and V are dotted to get the final output.

$$\text{Attention (Q, K, V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right) \cdot \text{V} \tag{9}$$

Where Q, K, and V are in vector form, and $Q \in R^{n\times d_K}$, $V \in R^{m\times d_V}$, $V \in R^{m\times d_V}$. The dimensions of query Q and key K are d_K, the dimension of the value V is d_V. $\frac{1}{\sqrt{d_K}}$ refers to the scaling factor to adjust the inner product of Q and K.

When employing the attention mechanism, a single attention mechanism is difficult to capture the features in the text from multiple angles and layers. Therefore, we use a multi-headed attention mechanism (Vaswani et al. 2017). Under the premise of not sharing parameters, each layer in the multi-head attention mechanism maps V, K, and Q through the parameter matrix, and then performs the scaled dot product attention calculation. According to the number of layers of multi-head attention h, the same operation is executed h times. Finally, the results of each layer are spliced to obtain feature information of different angles and different levels. The calculation formula is shown below:

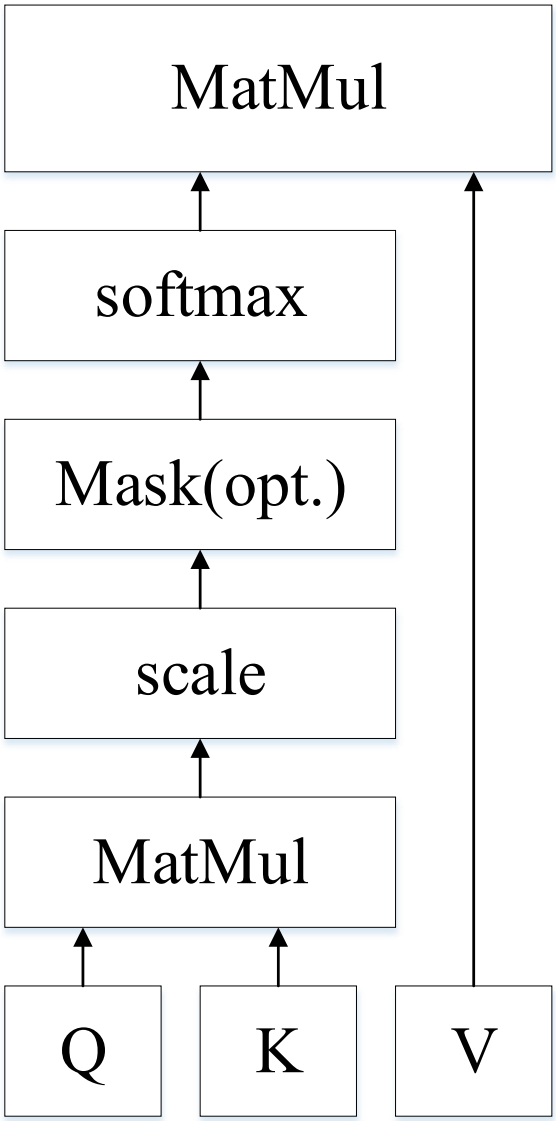

Fig. 4 Scaled Dot-Product Attention

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{10}$$

$$\text{MutiHead (Q, K, V)} = \text{Concat (head}_i, \ldots, \text{head}_h) \tag{11}$$

Where $W_i^Q, W_i^K \in R^{d_K\times d_K}$, $W_i^V \in R^{d_V\times d_V}$, the concat operation means to concatenate the results of each layer after the scaled dot product attention.

CRF layer

In the CRF layer, the previous layer outputs the feature vector h_t^x after passing through the BiLSTM network layer.

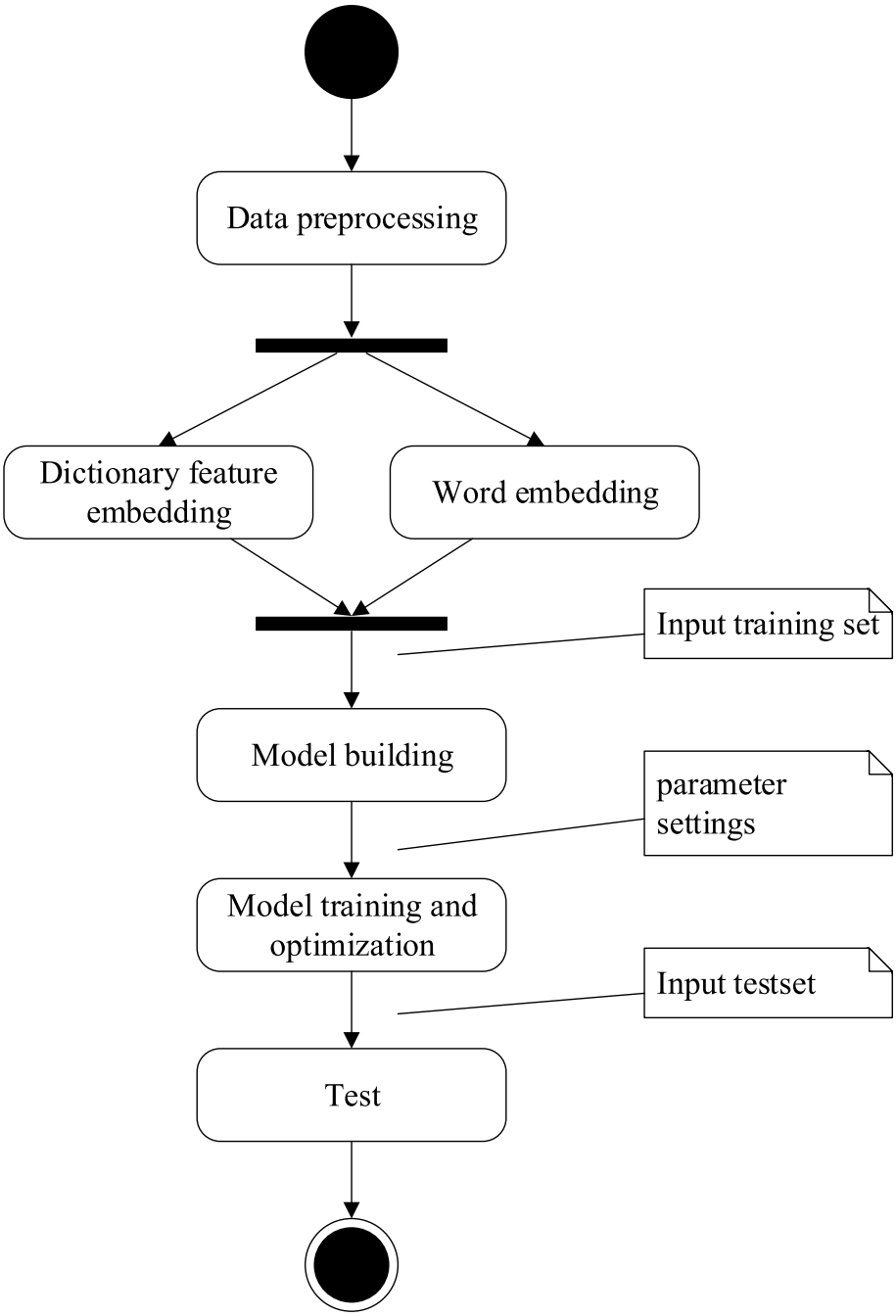

Fig. 5 Model flow chart

By inputting three sets of the same feature vector h_t^x, the information vector s_t is obtained through the attention layer. Then we concatenate h_t^x and s_t to obtain the information vector o_t that combines the context feature and its own feature. The specific formula is as follows:

$$s_t = \text{MutiHead}\left(h_t^x, h_t^x, h_t^x\right) \tag{12}$$

$$o_t = \text{Concat}\left(s_t, h_t^x\right) \tag{13}$$

Combine the information vector o_t and the domain dictionary feature vector h_t^g containing boundary information to obtain the final information vector representation p_t, which is used as the input of the CRF layer.

The traditional neural network layer uses the softmax classifier to independently make labelling decisions. It selects the label with the highest score as the label of the word according to the score of each word corresponding

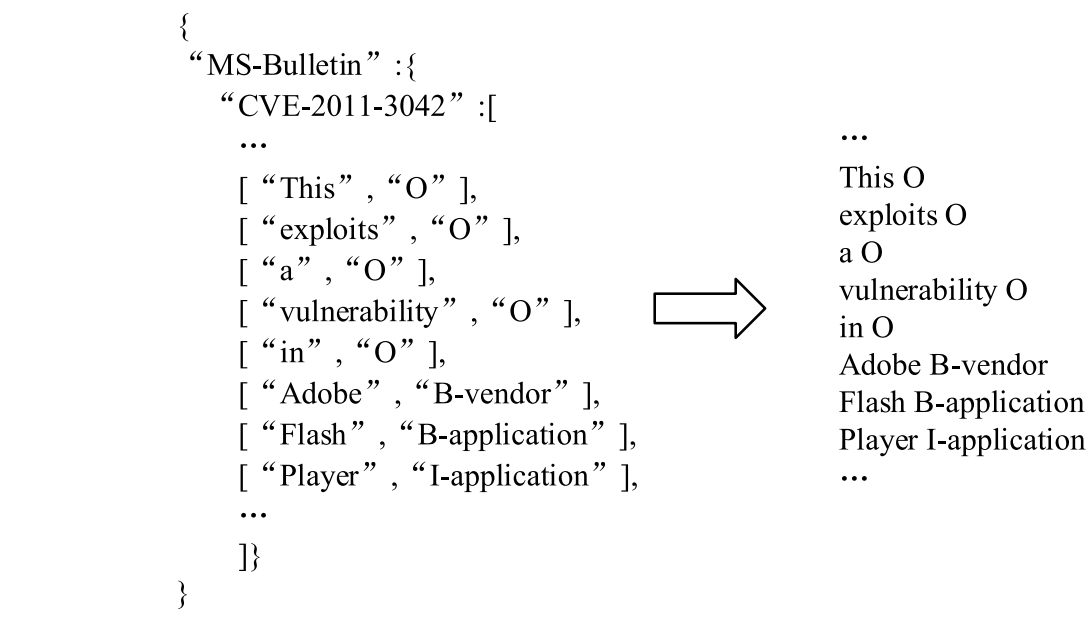

Fig. 6 Data format conversion

to each label. But in the actual tag sequence, the tag itself has certain constraint rules. The word label at the beginning of a sentence can only be "O" or "B-type", but not "I-type". "I-type" must follow "B-type" and cannot appear alone. Therefore, in the CRF layer, we use the conditional random field (CRF) model to decode the information vector generated by the previous layer, improve the accuracy of label prediction by learning the constraints between the labels, and obtain the final predicted label sequence.

Given a set of input sequence $p = (p_1, p_2, \ldots, p_n)$, the score $y = (y_1, y_2, \ldots, y_n)$ of the label is predicted through the CRF layer. At last, the *softmax* function is used to normalize all possible labelled paths to obtain the conditional probability of path y. The calculation formulas are as follows:

$$p_t = \text{Concat}\left(o_t, h_t^g\right) \tag{14}$$

$$s(p, y) = \sum_{i=0}^{n} Z_{i,y_i} + \sum_{i=0}^{n} T_{y_i,y_{i+1}} \tag{15}$$

$$\text{p(y | p)} = \frac{\exp(s(p, y))}{\sum_{\tilde{y} \in Y} \exp(s(p, \tilde{y}))} \tag{16}$$

Where p is the information vector matrix generated by the BiLSTM layer and the Attention layer. Z_{i,y_i} corresponding to the score of the i^{th} word in the sentence may be the label y_i, T represents the transition matrix of the labeling state, $T_{y_i,y_{i+1}}$ represents the transition probability from the labeled state y_i to y_{i+1}, $\tilde{y}$ is the true label value, and Y is the set of all possible label sequences.

Experiment

Figure 5 is an introduction to the specific process of the model, which mainly includes seven steps: data preprocessing, dictionary feature embedding, word embedding, model building, model training and optimization, and test.

Dataset

Since there is currently no unified data corpus in the field of cyber security, we use an open source cyber security

Table 2 Datasets statistics

Datasets	Sentences	Tokens	Entities
Training set	16162	595068	47628
Test set	4635	170736	14002
Validation set	2298	85345	6868
Total	23095	851149	68498

Table 3 Number of entities included in different entity categories

Datasets	Application	Version	Hardware	OS	Edition	File	Vendor
Training set	14053	20243	370	2829	418	1953	7753
Test set	4118	5890	110	918	102	528	2326
Validation set	1998	2951	57	405	55	309	1093

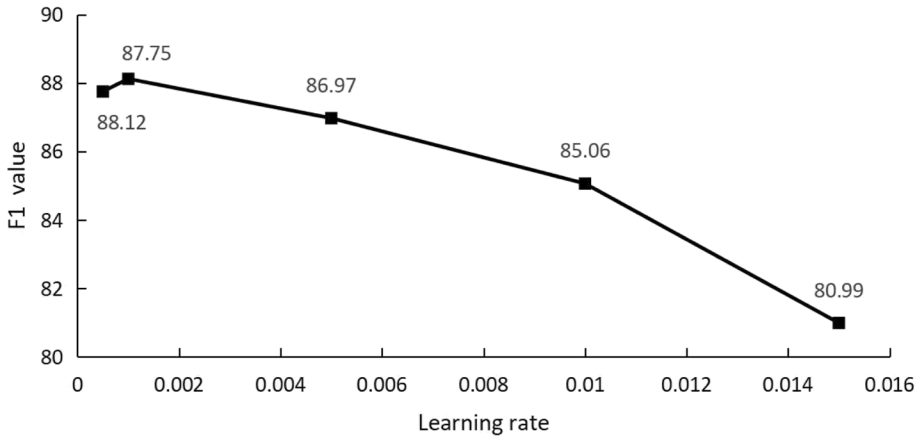

Fig. 7 The effect of learning rate on F1 value

corpus to train and evaluate our model, which was created by Bridges et al. (2013). The dataset is composed of data extracted from the cyber security domain based on machine learning algorithms, including NVD, Metasploit, and Microsoft Security Bulletins. There are 15 entity types defined in the corpus. We selected 7 entity types, which are most relevant to cyber security for model evaluation. They are application, version, hardware, OS, version, file, and vendor. Based on the wikipedia computer security category page, the NVD, and CVE databases, we constructed a cyber security dictionary. According to the original form provided by Bridges et al. (2013) and others, all the data is stored in a JSON file. In order to obtain a file that meets the requirements of the input format, we convert the file into CoNLL 2000 format as the input of the model. Figure 6 shows the data conversion process.

In the new annotation corpus, we removed the separation between each corpus and the annotation in the three corpora, and each word is in a separate line. At the same time, the seven types of entity types we selected are retained, and the tags of other types that are not relevant are defined as "O". Each line contains the words and their entity types mentioned in the text, with spaces as separators. The complete dataset contains 23095 sentences. We select 70% of them as the training set, 10% as the validation set, and the remaining 20% as the test set. Table 2 describes the size of the dataset and Table 3 details the distribution of each entity category in different datasets.

Our cyber security dictionary is mainly extracted from the following three data sources, namely Wikipedia, CVE and NVD, and cyber security blog. Wikipedia contains page classifications of specific categories. According to the method proposed by Zhang et al. (2017), we try to use some candidate keywords to obtain cyberspace security entities in DBpedia. Secondly, we use the seeds provided by Jones et al. (2015) to apply the bootstrap algorithm to the NVD and CVE corpus to obtain the cyberspace security entity sets. The algorithm uses a semi-supervised method to help mark the seeds of a few important relationships by querying users, and iteratively builds on the seeds to generate a larger corpus. Finally, we crawled a large number of related blogs from some cyber security blog sites to extract their tags, such as Cisco Security, Krebson Security and Naked Secuirty.

The dictionary data constructed from the three datasets was manually filtered to ensure its quality. In this manual screening process, the authors of this paper cooperated to filter the constructed dictionary and exclude vocabulary that did not meet the requirements. The three authors screened the initial set of dictionaries respectively, took the intersection of the obtained dictionaries, and discussed the dissent words to form the final domain dictionary.

The final dictionary contains a total of 15,357 words, of which 7,749 words from the NVD and CVE datasets, 3327 words from cyber security blog data, and finally 4281 words from Wikipedia. The specific data description is shown in the following Table 4.

Table 4 Dictionary data statistics

Datasets	Words	Entities	Tags
CVE and NVD	7749	5709	7835
Cyber security blogs	3327	1263	2124
Wikipedia	4281	920	1578

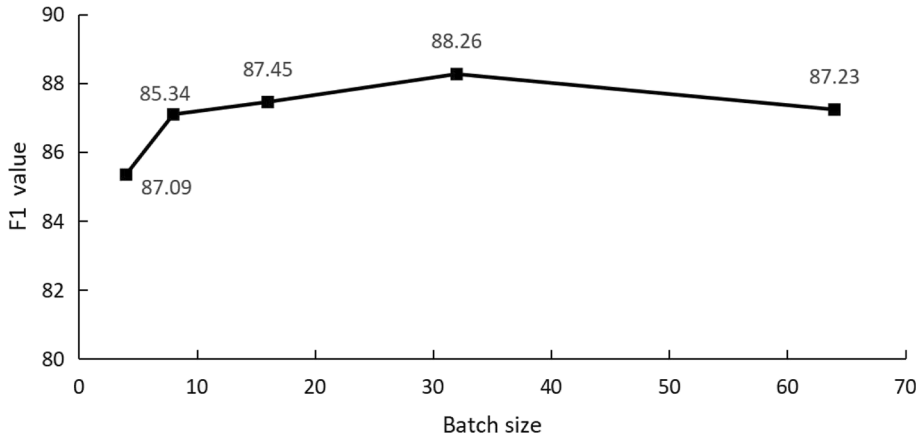

Fig. 8 The effect of Batch size on F1 value

Environment and parameter settings

Our experiment is running on the windows 10 operating system, using the Keras 2.24 framework, and using Python to implement model construction and training. We select the optimal parameters of the model by setting multiple sets of experiments with control variables, among which the experimental results of learning rate and batch size are shown in Figs. 7 and 8.

As can be seen from the Figs. 6 and 7, when the learning rate and batch size are 0.001 and 32 respectively, the F1 value of the model is optimal. The other parameters of the model is determined by the same method. The final settings are shown in Table 5.

Results and analysis

In order to verify the effectiveness of our proposed method, we selected the same dataset and evaluation indicators, and set up multiple sets of controlled experiments to analyse the results.

Usually, the NER system compares the output with human annotations to evaluate system performance. In our experiment, we use the following three indicators to evaluate the cyber security NER model, namely Precision(P), Recall(R) and F1 value (F1). In this paper, we use BiLSTM-CRF as the baseline model. The second model is BiLSTM-Dic-CRF, which adds dictionary features to the baseline model. The third model is BiLSTM-Att-CRF, which employs the attention machine. The fourth model, the BiLSTM-Dic-Att-CRF model, is based on the third model and adds dictionary features.

The experimental results of these models are listed in Table 6. It can be seen from the table that adding dictionary information and attention mechanism to the baseline model can improve the F1 value of NER for cyber security. Therefore, adding appropriate external knowledge or features to the annotation model is beneficial to improve the recognition result. Furthermore, the employment of a multi-head self-attention mechanism captures contextual information in multiple different subspaces, which improves entity recognition performance for irregular text. Our BiLSTM-Dic-Att-CRF model achieves the optimal results, with the precision rate, recall rate and F1 value are 7.00%, 3.98% and 5.45% higher than the baseline model, respectively. It indicates that adding external information on the basis of fully capturing sentence features can better assist the identification of cyber security entities.

To analyse the performance of our model in detail, Table 7 compares the baseline model (BiLSTM-CRF) with the performance indicators for each entity type of our

Table 5 Model parameter settings

Parameters	Setting
Word embedding size	100
Feature embedding size	25
Word BiLSTM hidden node	100
Feature BiLSTM hidden node	25
Batch size	32
Dropout rate	0.5
Optimizier	Adam
Learning rate	0.001
Epochs	15

Table 6 Experimental results of cybersecurity NER

Models	P	R	F1
BiLSTM-CRF	83.19	82.62	82.91
BiLSTM-Dic-CRF	85.37	87.25	86.62
BiLSTM-Att-CRF	85.20	84.86	85.03
BiLSTM-Dic-Att-CRF	90.19	86.60	88.36

Table 7 Baseline and our model experimental comparison

Class (Baseline / Ours)	P	R	F1
Application	72.79 / 84.91	71.86 / 83.20	72.32 / 84.04
OS	86.71 / 92.73	76.03 / 73.64	81.02 / 82.09
Hardware	52.56 / 65.08	37.21 / 37.27	43.62 / 47.40
Version	95.47 / 96.58	93.77 / 94.50	94.61 / 95.53
Vender	80.61 / 87.78	81.17 / 87.45	80.89 / 87.47
File	54.06 / 71.76	79.18 / 63.94	64.25 / 67.58
Edition	79.07 / 78.00	33.33 / 38.24	46.90 / 51.32

proposed model. Overall, the model achieves better performance for most tags, which proves the validity of our model. Moreover, the model performed better on the five entity types with a high number of entities, but worse on the hardware and edition type due to the small size of the training set data. We can also find that deep learning algorithms usually need a large amount of data tags to learn for better prediction. In our training set, only a small number of entities are labelled as these two types. In 23095 sentences, 370 are labelled hardware and 428 are labelled edition. These figures are much lower than other tags such as application (14,053 tags) and vendor (7,703 tags).

In Fig. 9, we compare specific examples of model recognition of rare entities. "app" and "ver" are the abbreviations for application and version, respectively. "hare" refers to the " hardware " label.

For sentence 1, the baseline model failed to identify the rare entity "Mediator Framework" and mistakenly identified its application category as hardware, while our proposed model could identify the correct entity type. For the entity word "Cisco" with a large number of instances, both the baseline model and our model can accurately

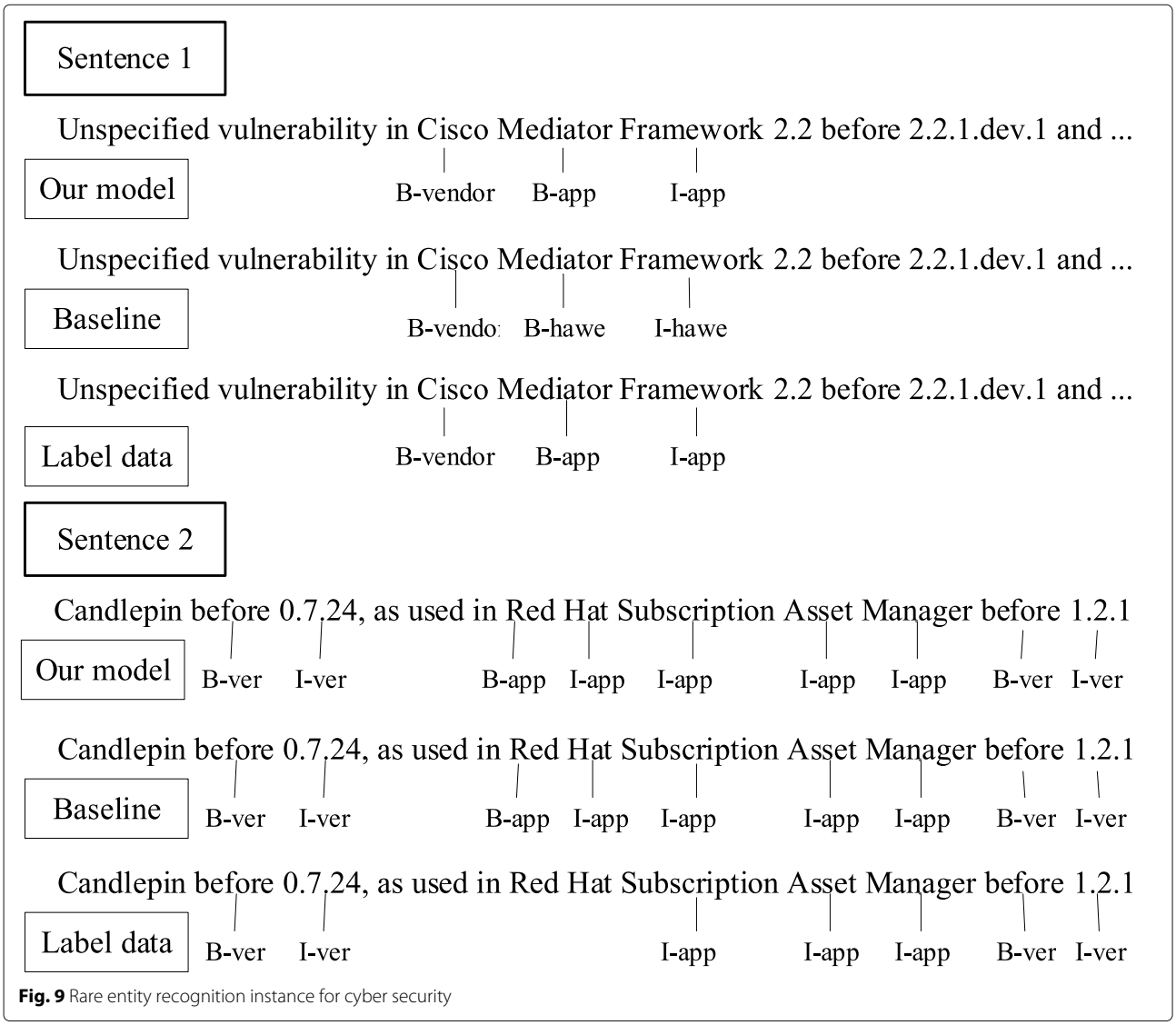

Fig. 9 Rare entity recognition instance for cyber security

Table 8 Model efficiency analysis

Models	Training time(s)	Loading time(s)	Test time(s)
BiLSTM-CRF	545.17	5.88	9.43
BiLSTM-Dic-CRF	576.43	7.30	11.87
BiLSTM-Att-CRF	797.46	7.16	12.20
BiLSTM-Dic-Att-CRF	1015.54	7.64	13.32

identify and classify it. For sentence 2, both models failed to accurately identify the boundary of the entity because the entity contained multiple words and appeared too few times in the training set. It is difficult for both of the models to accurately learn its characteristics. Of course, this is also a direction for future improvement of our work.

The efficiency of the model is also very important in the application of practical scenarios. Therefore, the time efficiency analysis of the model was carried out. By averaging the multiple loading models, the learning time, loading time and testing time of the model were calculated and compared in Table 8.

It can be seen from Table 8 that the loading time of each model is not significantly different. The test time and learning time of our model are longer because the more complexity of our model. In the experiment, the average processing time of the fastest model is the baseline model BiLSTM-CRF, which is 4.11m/s. Correspondingly, our model is 5.37m/s. Generally, the amount of online application data is small, so this difference in speed is not very obvious. However, offline applications have relatively low requirements on time, and F1 value of our model has been improved. In the above cases, the running time is within an acceptable range.

Conclusion and future work

A NER method for cyber security is presented in this paper. The experimental results show that the proposed model obtains better results than the existing common methods by integrating the domain dictionary features and multi-attention mechanism. Especially the recognition effectiveness of rare entities is improved. However, there is room for improvement in the extraction of complex entity. Furthermore, text mining in the field of cyber security is of great significance. Information extraction in the field of cyber security still faces many challenges, such as nested entity identification and overlapping relationship extraction. In future work, we will continue to study these issues of information extraction in the field of cyber security.

Acknowledgements
Not applicable.

Authors' contributions
Chen Gao designed the feature extraction pipeline, performed the experiments and drafted the manuscript. Xuan Zhang and Hui Liu participated in problem discussions and improvements of the manuscript. The author(s) read and approved the final manuscript.

Funding
Project supported by the National Natural Science Foundation of China under Grant No. 61862063, 61502413, 61262025; the National Social Science Foundation of China under Grant No. 18BJL104; the Natural Science Foundation of Key Laboratory of Software Engineering of Yunnan Province under Grant No. 2020SE301; Yunnan Science and Technology Major Project under Grant No. 202002AE090010, 202002AD080002-5; the Data Driven Software Engineering Innovative Research Team Funding of Yunnan Province under Grant No. 2017HC012.

Availability of data and materials
CVE data can be found at: http://cve.mitre.org/ NVD data can be found at: https://nvd.nist.gov/ Cyberspace security experiment data can be found at: https://github.com/stucco/auto-labeled-corpus.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Software, Yunnan University, 650091 Yunnan, China. [2]Key Laboratory of Software Engineering of Yunnan Province, 650091 Yunnan, China. [3]Engineering research center of cyberspace, 650091 Yunnan, China.

Received: 21 October 2020 Accepted: 11 January 2021
Published online: 03 May 2021

References

Bridges R, Jones C, MD. Iannacone KT, Goodall J (2013) Automatic labeling for entity extraction in cyber security. arXiv preprint arXiv:1308.4941
Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
Dionísio N, Alves F, Ferreira P, Bessani A (2019) Cyber threat detection from twitter using deep neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, Budapest. pp 1–8
Gasmi H, Bouras A, Laval J (2018) Lstm recurrent neural networks for cyber security named entity recognition. In: Proceedings of the Thirteenth International Conference on Software Engineering Advances, Nice
Gu X, Liu J, Cheng P, He X (2018) Tweet malware name recognition based on enhanced bilstmcrf model(in chinese). Comput Sci 47:245–250
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780
Huang Z, Xu W, Yu K (2015) Bidirectional lstm-crf models for sequence tagging. International Symposium on Foundations and Practice of Security. arXiv preprint
Jones C, Bridges R, Huffer K, Goodall J (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
Joshi A, Lal R, Finin T, Joshi A (2013) Extracting cybersecurity related linked data from text. In: Proceedings of the 2013 IEEE Seventh International Conference on Semantic Computing. IEEE, Irvine. pp 252–259
Lal R (2013) Information extraction of security related entities and concepts from unstructured text. Dissertation. University of Maryland Baltimore County
LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
Li T, Guo Y, Ju A (2019) A self-attention-based approach for named entity recognition in cybersecurity. In: Proceedings of the 15th International Conference on Computational Intelligence and Security. IEEE, Macao. pp 147–150
Liu W (2020) Network security entity recognition methods based on the deep neural network. In: Data Processing Techniques and Applications for Cyber-Physical Systems. Springer, Singapore. pp 1687–1692
Mazharov I, Dobrov B (2018) Named entity recognition for information security domain. In: Proceedings of the Data Analytics and Management in Data Intensive Domains, Moscow
Mulwad V, Li W, Joshi A, Finin T, Viswanathan K (2011) Extracting information about security vulnerabilities from web text. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. IEEE, Lyon. pp 257–260

Qin Y, Shen G, Zhao W, Chen Y, Yu M, Jin X (2019) A network security entity recognition method based on feature template and cnn-bilstm-crf. Frontiers Inf Technol Electronic Eng 20:872–884
Roy A, Park Y, Pan S (2017) Learning domain-specific word embeddings from sparse cybersecurity texts. arXiv preprint arXiv:1709.07470 47:245–250
Simran K, Sriram S, Vinayakumar R, Soman K (2019) Deep learning approach for intelligent named entity recognition of cyber security. In: International Symposium on Signal Processing and Intelligent Recognition Systems. Springer, Singapore. pp 163–172
Tikhomirov M, Loukachevitch N, Sirotina A, Dobrov B (2020) Using BERT and Augmentation in Named Entity Recognition for Cybersecurity Domain. In: International Conference on Applications of Natural Language to Information Systems. Springer, Cham. pp 16–24
Vaswani A, Shazeer N, Parmar N (2017) Attention is all you need. Attention is all you need. Advances in neural information processing systems. Curran Associates, Inc Vol. 30. pp 5998–6008
Wang X, Xiong Z, Du X, Jiang J, Jiang Z (2020) NER in Threat Intelligence Domain with TSFL. In: CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham. pp 157–169
Weerawardhana S, Mukherjee S, Ray I, Howe A (2014) Automated extraction of vulnerability information for home computer security. Int Symp Found Pract Secur 8930:356–366
Wu H, Li X, Gao Y (2020) An effective approach of named entity recognition for cyber threat intelligence. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, Chongqing. pp 1370–1374
Xiao Z (2017) Towards a two-phase unsupervised system for cybersecurity concepts extraction. In: Proceedings of the 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. IEEE, Guilin. pp 2161–2168
Zhang H, Guo Y, Li T (2019) Multifeature named entity recognition in information security based on adversarial learning. Secur Commun Netw 2:1–9
Zhang X, Liu X, Li X, Pan D (2017) Mmkg: an approach to generate metallic materials knowledge graph based on dbpedia and wikipedia. Comput Phys Commun 211:98–112
Zhou S, Long Z, Tan L, Guo H (2018) Automatic identification of indicators of compromise using neural-based sequence labelling. arXiv preprint arXiv:1810.10156

Publisher's Note