Computers, Materials & Continua
DOI:10.32604/cmc.2020.012023
Article

Tech Science Press

Adversarial Active Learning for Named Entity Recognition in Cybersecurity

Tao Li[1], Yongjin Hu[1,*], Ankang Ju[1] and Zhuoran Hu[2]

[1]Zhengzhou Institute of Information Science and Technology, Zhengzhou, 450001, China
[2]College of Letters and Science, University of Wisconsin-Madison, Madison, 53706, USA
[*]Corresponding Author: Yongjin Hu. Email: thechanger@126.com
Received: 10 June 2020; Accepted: 25 July 2020

Abstract: Owing to the continuous barrage of cyber threats, there is a massive amount of cyber threat intelligence. However, a great deal of cyber threat intelligence come from textual sources. For analysis of cyber threat intelligence, many security analysts rely on cumbersome and time-consuming manual efforts. Cybersecurity knowledge graph plays a significant role in automatics analysis of cyber threat intelligence. As the foundation for constructing cybersecurity knowledge graph, named entity recognition (NER) is required for identifying critical threat-related elements from textual cyber threat intelligence. Recently, deep neural network-based models have attained very good results in NER. However, the performance of these models relies heavily on the amount of labeled data. Since labeled data in cybersecurity is scarce, in this paper, we propose an adversarial active learning framework to effectively select the informative samples for further annotation. In addition, leveraging the long short-term memory (LSTM) network and the bidirectional LSTM (BiLSTM) network, we propose a novel NER model by introducing a dynamic attention mechanism into the BiLSTM-LSTM encoder-decoder. With the selected informative samples annotated, the proposed NER model is retrained. As a result, the performance of the NER model is incrementally enhanced with low labeling cost. Experimental results show the effectiveness of the proposed method.

Keywords: Adversarial learning; active learning; named entity recognition; dynamic attention mechanism

1 Introduction

Owing to the growing number of increasingly fierce attack-defense campaigns in cyberspace, cybersecurity situation has become more and more severe, resulting in extensive cybersecurity incidents. Through keep abreast of and then analyzing the past cybersecurity incidents, security analysts can gain a deep understanding of cyber threats and the entire threat invasion process.

As there exists detailed description about cyber threats in cyber threat intelligence, the analysis and sharing of cyber threat intelligence would be helpful for organizations to implement proactive cybersecurity defense. However, such cyber threat intelligence mainly comes from textual sources, like cybersecurity white papers, blogs, vendor bulletins and hacker forums. Textual cyber threat intelligence

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

covers substantial threat-related information and is the primary source for constructing a cybersecurity knowledge base. However, processing such a massive amount of cyber threat intelligence may overwhelm security analysts, especially, because discovering threat-related knowledge from cybersecurity texts manually is cumbersome and time-consuming. To obtain threat-related information automatically, information extraction is required, converting the textual cyber threat intelligence into structured linked data and constructing cybersecurity knowledge graph, which can correlate the numerous seemingly unrelated cyber threat information.

In general, information extraction includes named entity recognition, relation extraction and event extraction. And information extraction in cybersecurity has attracted considerable attention. The Task 8 of the SemEval-2018 workshop on semantic evaluation aims to implement semantic extraction from cybersecurity reports using natural language processing (SecureNLP) [1]. In the SecureNLP task, there are subtasks on the prediction of entity, relation and attribute in malware texts. Named entity recognition (NER) is the major task in information extraction and is the premise for the remaining two tasks. Moreover, NER is also the foundation of constructing knowledge graph and natural language understanding. NER in cybersecurity focuses on specific and important threat-related classes, such as software, malware, vulnerability, tool, attack-pattern [2].

Compared with the earlier NER methods that required handcrafted features, deep learning-based architectures without manual feature engineering have achieved better performance [3]. However, such deep neural network-based models rely heavily on a profusion of labeled training samples [4]. Moreover, in the specific domain, there is limited annotation budget and it is difficult to obtain a large-scale labeled training set. There are two challenges in particular in the annotation of cybersecurity texts: only cybersecurity practitioners or cybersecurity experts can precisely annotate cybersecurity texts; and, compared with the corpora of general domain, cybersecurity texts contain more specific entity types, which requires more manual efforts to complete the annotation.

To address the above issues, active learning is proposed for incrementally selecting more informative samples from the unlabeled data pool. The selected samples are annotated by oracles, which are then added to the training set. As a result, we can incrementally train the model to improve its performance with low labeling cost. However, most traditional active learning algorithms rely on the uncertainty sampling strategy, which is complex when applied to the NER task due to its abundant tag space. In the paper, instead of evaluating the prediction uncertainty of the model, we consider the similarity between labeled samples and unlabeled samples. As a result, we train a discriminator with adversarial learning to sample the informative sentences, and then propose an adversarial active learning scheme for entity extraction in cybersecurity. More specifically, with the extracted features from the input sentence, we train a discriminative model to judge the similarity between the labeled sentence and unlabeled sentence. In addition, leveraging the long short-term memory (LSTM) network and bidirectional LSTM (BiLSTM) network, we propose a novel NER framework with the introduction of a dynamic attention mechanism into the BiLSTM-LSTM encoder-decoder. The two main contributions of this paper are:

1. To solve the lack of labeled *corpus* for training the NER model in cybersecurity, combining adversarial learning with active learning, we propose an adversarial active learning framework to incrementally select informative samples for oracles to be labeled, which reduces the labeling cost for the training set.
2. For entity extraction from cybersecurity texts, we propose a novel dynamic attention-based BiLSTM-LSTM model for the NER task, where a dynamic attention mechanism is presented to adaptively capture dependency between two tokens, and LSTM is used as the tag decoder. Our model outperforms the mainstream NER models used in cybersecurity.

2 Related Work

By constructing the cybersecurity knowledge graph, security analysts can query and manage cybersecurity information intelligently. In addition, with the cybersecurity knowledge graph, security analysts can obtain a complete view of cybersecurity situation and make more accurate predictions on threat development, which would be significant for proactively improving cybersecurity. Since cybersecurity knowledge graph consists of cybersecurity entities and their semantic relations, NER becomes the fundamental and crucial task.

The earlier NER methods were mainly based on statistical machine learning models, including maximum entropy model (MEM), hidden Markov model (HMM), support vector machine (SVM), and conditional random field (CRF). However, such machine learning-based methods rely on considerable feature engineering, which would result in poor robustness and generalization of models. With the success of deep learning in image recognition, speech recognition and natural language processing (NLP), there have been many deep neural networks-based NER methods. Typically, due to capturing the contextual feature information, bidirectional long short-term memory (BiLSTM) neural networks [5] have achieve great success in entity extraction. Gasmi et al. [6] utilized BiLSTM-CRF to extract cybersecurity concepts and entities and achieved promising results. Long et al. [7] applied the BiLSTM-based model to identify indicators of compromise (IOCs). Satyapanich et al. [8] combined BiLSTM with the attention mechanism to classify cybersecurity event nugget types and event argument types. Dionísio et al. [9] introduced a BiLSTM-CRF model to identify named entities from cyber threat-related tweets, which was used to issue cybersecurity alerts and fill the IOCs. Pingle et al. [2] developed a deep neural network-based semantic relationship extraction system for cyber threat intelligence, which was aimed at obtaining semantic triples from open source cyber threat intelligence and then was combined with the security operation center to further enhance cybersecurity defense.

For most deep neural networks-based NER methods, chain CRF [10] acts as the tag decoder. However, as an alternative, recurrent neural networks (RNNs) can be also used for decoding tags of sequences [11–13]. Shen et al. [14] employed LSTM as the tag decoder for NER task. They found it can not only yield performance comparable to CRFs, but also with massive entities, LSTM outperformed CRF for decoding tags and was faster to train.

Adversarial learning derives from the generative adversarial networks (GANs) [15], which were originally designed to synthesize images for the purpose of data augmentation. The GAN architecture consists of a generator and a discriminator, between which there is a two-player minimax game. Through the alternate training, the generator tries to fool the discriminator with the generated data by capturing the data distribution of the real sample, while the discriminator judges the input data as true or fake, distinguishing the generated data from the real data. In early phase, due to generating diverse data in continuous space, GANs were used as generative models. In essence, GAN provides a framework, within which the originally poor data can be enhanced through adversarial training between the generator and discriminator; after the alternate training, a well-trained discriminator can be obtained and further used as a classifier. In recent years, GANs have been extensively applied into fields other than computer vision, including speech recognition and natural language processing. Gui et al. [16] implemented part-of-speech tagging for twitter through an adversarial discriminator to learn common features among the out-of-domain labeled data, unlabeled in-domain data and the labeled in-domain data. Zeng et al. [17] applied adversarial learning for distant supervised relation extraction. They used the deep neural network as the generator to generate the negative class, and trained the piecewise convolutional neural network (PCNN) as the discriminator to efficiently classify the final relation.

Active learning aims to incrementally select samples for labeling, thus achieving better classification performance with lower labeling cost [18]. Current researches on active learning include the query-

synthesizing method and the pool-based method. The query-synthesizing method belongs to the generative model, which generate informative samples to be labeled. Zhu et al. [19] first proposed the GAN-based active learning model to generate the samples to be labeled. However, due to difficulties in training the GAN model, there is pattern collapse of the generated samples, which cannot capture the data distributions of the real samples [15]. In addition, it is also difficult for the oracle to label the meaningless generated samples. Therefore, the performance of query-synthesizing algorithms relies on the quality and diversity of the generated samples.

Pool-based active learning selects the informative samples from the unlabeled sample pool, which is the main research focus of active learning. The use of pool-based active learning algorithms has been explored in many tasks, like image classification, speech recognition, text classification, information retrieval, etc. The representative sampling strategies of pool-based active learning algorithms include uncertainty-based sampling [20], information-based sampling [21], ensemble-based sampling [22], expected model change-based sampling [23] and core set-based sampling [24]. For typical uncertainty-based methods, Houlsby et al. [25] proposed a Bayesian active learning by disagreement (BALD) model, in which the sampling function is estimated by the mutual information of the training samples with respect to the model parameters. Gal et al. [26] measured the uncertainty in the prediction of neural networks by estimating the relationship between uncertainty and dropout, which was then applied to active learning. Sener et al. [24] proposed a core set-based active learning algorithm, which minimized the Euclidean distance between the sampled data points and unsampled data points in the feature space when training the model. Kuo et al. [27] proposed an ensemble-based active learning model to select samples by measuring the uncertainty, but the model tends to cause redundant sampling. Shen et al. [14] investigated the application of active learning for the NER task, and then compared the performance of three typical active learning models: the least confidence (LC)-based model, the BALD model and the maximum normalized log-probability (MNLP)-based model.

3 Proposed Method

In the section, we describe the proposed adversarial active learning framework for NER in cybersecurity. Our proposed model includes two components: the NER module and the adversarial active learning module. The detailed implementation of the proposed model is described below.

3.1 Dynamic Attention-Based BiLSTM-LSTM for NER

Fig. 1 illustrates the architecture of the proposed novel NER model, depicting the embedding layer, BiLSTM feature encoding layer, dynamic attention layer and the LSTM tag decoding layer.

3.1.1 Embedding Representation

The embedding layer converts the cybersecurity sentence into the low-dimensional and dense vector, which is input into the encoder for feature extraction. To obtain high-quality embedding representation, in this paper, we aggregate massive cybersecurity corpora from recent threat intelligence blogs, security news, vulnerability descriptions from common vulnerabilities and exposures (CVE) and advanced persistent threat (APT) reports. Then we use the word2vec algorithm to obtain a 100-dimensional word embedding, which is looked up when converting the input sentence into embedding representation. The obtained word embedding representation serves as the word-level feature of the input sentence.

In addition, to complete the embedding representation, we extract character-level features for each word of the input sentence. Since convolution neural networks (CNNs) can extract local features of the input data and select the most representative local features through pooling strategy, CNN was exploited to obtain character features in [28–29], and achieved promising results in corresponding tasks. In this paper, we use

the same CNN structure as [28] to obtain the character-level feature, which is then concatenated with word embedding. Finally, the concatenated vector is input into the feature encoder.

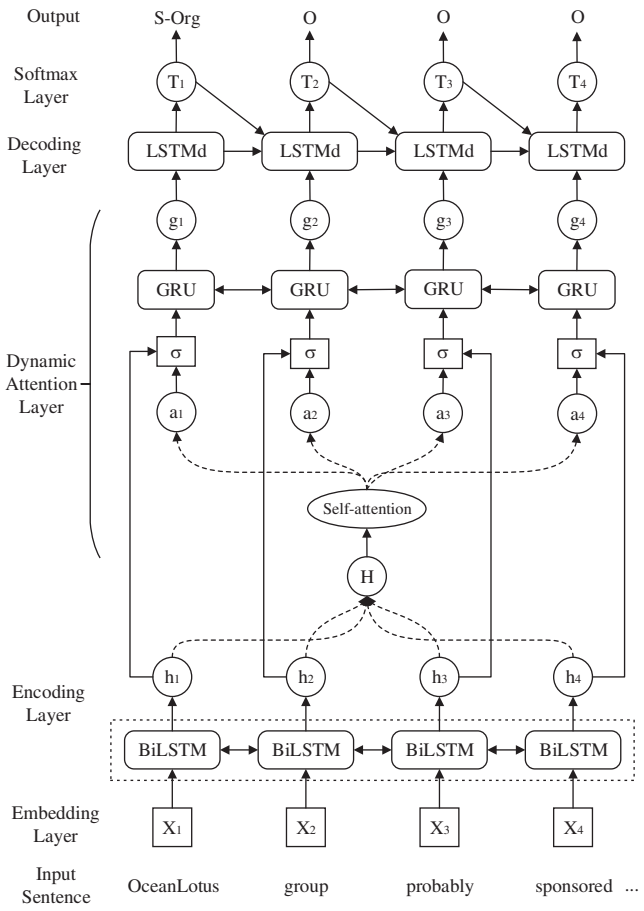

Figure 1: Dynamic attention-based BiLSTM-LSTM model

3.1.2 BiLSTM Layer

LSTM has been widely used in NLP tasks, which has significant advantages in modeling the sequences. LSTM integrates the past time-step information and the current input of the sequence to decide the current output, which solves the problem of long-distance dependency when modeling sequences. LSTM consists of the forget gate, input gate and output gate, which are used to control information flow. The specific implementation of LSTM is

$$\begin{cases} i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ c_t = i_t \cdot \tan h(W_{xc}x_t + W_{hc}h_{t-1} + W_{cc}c_{t-1} + b_c) + f_t \cdot c_{t-1} \\ h_t = o_t \cdot \tan h(c_t) \end{cases} \tag{1}$$

where i_t, f_t, o_t and c_t respectively indicates input gate, forget gate, output gate and cell vector, σ and $\tan h$ are both nonlinear activation function, $W(*)$ are the weight matrices of the corresponding gate, and $b(*)$ are the bias vectors. In addition, h_t indicates the hidden state of LSTM at time step t.

For text mining tasks, it is necessary to make full use of the context information of the sequence. While LSTM solves the problem of long-distance dependency of sequence modeling, it learns features along the forward direction, which can only use the past time-step information. Therefore, we employ the BiLSTM, consisting of the forward LSTM and reverse LSTM, to capture contextual information for the current time step. Specifically, the forward LSTM outputs hidden state $\overrightarrow{h}_t$, and the reverse LSTM outputs hidden state $\overleftarrow{h}_t$. Then we concatenate both $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$, obtaining $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$, which denotes the feature encoding of the current input relying on the context information. $H = \{h_1, h_2, \ldots, h_T\}$ further indicates the final feature of the sequence encoded by BiLSTM.

3.1.3 Dynamic Attention Mechanism

Since the self-attention mechanism directly learns the dependencies between any two tokens within the sequence, it has attained success in machine translation and semantic labeling. Cao et al. [30] applied the self-attention mechanism to Chinese NER task and achieved improved performance. Specifically, with the feature encoding $H = \{h_1, h_2, \ldots, h_T\}$, the calculation of self-attention follows:

$$Q_i\ ,\ K_i\ ,\ V_i\ =\ HW_Q^i\ ,\ HW_K^i\ ,\ HW_V^i \tag{2}$$

$$u_i = \text{Attention}(Q_i,\ K_i,\ V_i) = \text{softmax}\left(\frac{Q_i {K_i}^T}{\sqrt{d}}\right) V_i \tag{3}$$

where with the trainable weight parameters W_Q^i, W_K^i and W_V^i, the feature encoding H is transformed into query Q_i, key K_i and value V_i respectively in h different subspaces. Then, we concatenate all the subspaces' vectors, obtaining the vector $U = \{u_1, u_2, \ldots, u_h\}$. Finally, with the weight parameter W_O the attention vector $A = \{a_1, a_2, \ldots, a_T\}$ is expressed as

$$A = (u_1, u_2, \ldots, u_h) \cdot W_O \tag{4}$$

Although the above self-attention mechanism captures the dependencies between tokens within the sequence, for different words of the sequence, the attention weight remains unchanged, which may lead to inaccuracy. Analogous to human beings' dynamically changed attention, the attention in sequence modeling should also be dynamic, which would take the difference of attention distribution into consideration. Referring to [31] and combining the self-attention mechanism, we propose a dynamic attention mechanism to capture the dependencies. Specifically, in the dynamic attention layer, we first calculate the self-attention vector. Then the hidden state h_t and self-attention vector a_t are concatenated as $[h_t, a_t]$, which is fed into the nonlinear function sigmod for filtering. The result is dot-product with γ_t, whose result is further sent to the gate recurrent unit (GRU). The specific calculation denotes

$$\gamma_t = \text{sigmod}(W_s \cdot [h_t, a_t]) \tag{5}$$

$$\epsilon_t = \gamma_t \cdot [h_t, a_t] \tag{6}$$

$$g_t = GRU(g_{t-1}, \epsilon_t, \theta) \tag{7}$$

where W_S and θ are parameters. In addition, the output of dynamic attention is denoted as $G = \{g_1, g_2, \ldots, g_t\}$, which is further input into the decoding layer.

3.1.4 LSTM Decoding Layer

Compared with chain CRF for decoding tags, the LSTM decoder can significantly speed up the training and can achieve performance comparable to chain CRF decoder [16]. In this paper, we refer to [12] and adopt LSTM for tag decoding. The unit of the LSTM decoder is depicted in Fig. 2.

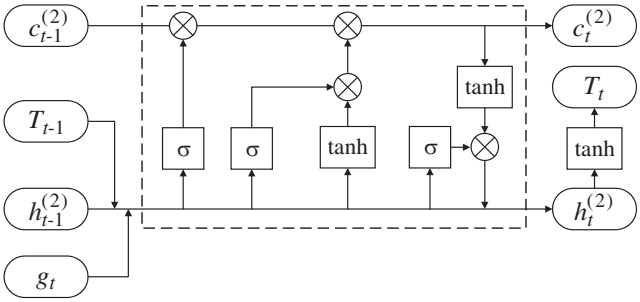

Figure 2: Unit of the LSTM decoder

At time step t, the input to the LSTM decoder consists of g_t obtained from the dynamic attention layer, hidden state h'_{t-1} of tag decoding at the previous time step, predicted tag embedding T_{t-1} at the previous time step, and the cell vector c'_{t-1}. The tag decoding is then calculated by

$$\begin{cases} i'_t = \sigma(W'_{xi} \cdot a_t + W'_{hi} \cdot h'_{t-1} + W_{ti} \cdot T_{t-1} + b'_i) \\ f'_t = \sigma(W'_{xf} \cdot a_t + W'_{hf} \cdot h'_{t-1} + W_{tf} \cdot T_{t-1} + b'_f) \\ c'_t = i'_t \cdot \tan h(W'_{xc} \cdot a_t + W'_{hc} \cdot h'_{t-1} + W'_{tc} \cdot T_{t-1} + b'_c) + f'_t \cdot c'_{t-1} \\ o'_t = \sigma(W'_{xo} \cdot a_t + W'_{ho} \cdot h'_{t-1} + W'_{co} \cdot c'_t + b'_o) \\ h'_t = o'_t \cdot \tan h(c'_t) \\ T_t = W_{ts} \cdot h'_t + b_t \end{cases} \tag{8}$$

where both σ and $\tan h$ are nonlinear activation functions, and i'_t, f'_t, o'_t respectively indicates input gate, forget gate, output gate of LSTM decoder. c'_t is cell vector and h'_t is the hidden state at time step t. T_t indicates the predicted tag embedding of the word x_t.

With the predicted tag embedding, we utilize the softmax classifier to calculate the tag probability, obtaining the probability p_t^i of i-th tag for the current word x_t,

$$p_t^i = p(i|x_t) = \frac{\exp(S_t^i)}{\sum_{j=1}^{N} \exp(S_t^j)} \tag{9}$$

where $S_t = W_s T_t + b_s$ indicates the scores on all the tags of the current word x_t. W_s is the weight parameter, and b_s is the bias parameter of the softmax classifier. Then the predicted tag of the word x_t is obtained by

$$y = \arg\max p(i|x_t) \tag{10}$$

For model training, the loss function of NER can be defined as

$$L_{NER} = \sum_{j=1}^{|D|} \sum_{t=1}^{L_j} \log p_j^t(y^t|s_j^t; \Theta)) \tag{11}$$

where $|D|$ indicates the size of training example, L_j indicates the length of the sentence s_j, and p_j^t is the tag probability of t-th word in sentence s_j.

3.2 Adversarial Active Learning

In the adversarial active learning module, we select informative samples by incrementally evaluating the similarity between the labeled sample and the unlabeled sample. Based on GANs, we combine the adversarial learning with active learning. The architecture of adversarial active learning is depicted in Fig. 3.

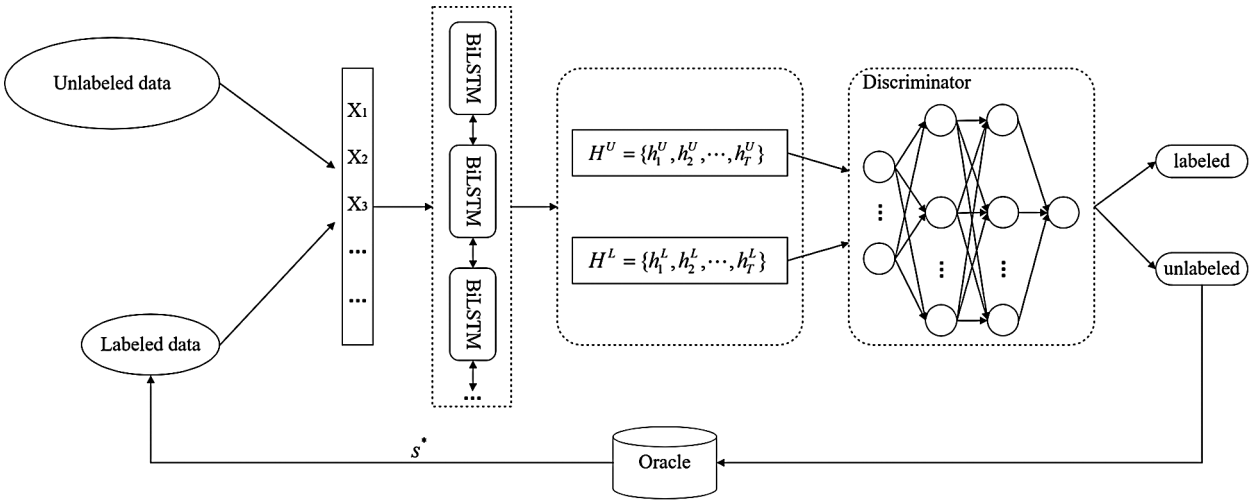

Figure 3: Architecture of adversarial active learning

With the labeled sample set S_L and unlabeled sample set S_U, we select the labeled sequence $s^L \sim S_L$ and the unlabeled sequence $s^U \sim S_U$ as the input into the adversarial active learning module. After the s^L and s^U sequences are first transformed into the embedding representation, the same structure of BiLSTM in the aforementioned NER module is exploited to obtain their feature encoding respectively, obtaining H^U and H^L in the learned latent space. As a result, the BiLSTM encoder and the discriminator are shaped as the adversarial network. Within the adversarial framework, BiLSTM encoder tries to learn feature encoding to deceive discriminator into predicting all the learned features are from the labeled sample set, while the discriminator is trained to distinguish the labeled sample and unlabeled sample. After the adversarial training, the discriminator outputs a similarity score. A high score indicates the unlabeled sample contains the similar informativeness that the labeled sample covers; a low score indicates there is a large difference in informativeness between the labeled sample and the unlabeled sample. Therefore, the unlabeled sample with the low similarity score is annotated by the oracle.

The objective of the BiLSTM encoder in adversarial active learning module is defined as minimizing the following loss function

$$L_G = -E_{s\sim S^L}[\log(D(s))] - E_{s\sim S^U}[\log(D(s))] \tag{12}$$

The objective of the discriminator is defined as minimizing the following loss function

$$L_D = -(E_{s\sim S^L}[\log(D(s))] + E_{s\sim S^U}[\log(1 - D(s))]) \tag{13}$$

We further comprehensively consider both objectives of the proposed NER module and the generator in adversarial active learning, obtaining the overall objective of NER in the proposed model as below

$$L = -\sum_{j=1}^{|D|}\sum_{t=1}^{L_j} \log p_j^t(y^t|s_j^t;\Theta)) \ - \lambda E_{s\sim S^L}[\log(D(s))] - \lambda E_{s\sim S^U}[\log(D(s))] \tag{14}$$

where λ is a hyperparameter used to balance the two parts of the above objective.

After the description of the proposed model, the operation process is depicted in Algorithm 1.

Algorithm 1: Cybersecurity entity recognition with adversarial active learning

Input: The initial labeled training set S_L and unlabeled set S_U. Initialize parameters for the proposed model.
Output: The well-trained dynamic attention-based BiLSTM-LSTM model.
1: **for** epoch = 1~N **do**
2: sample $s^L \sim S_L$:
3: calculate the loss function L_{NER} in Eq. (11)
4: sample $s^U \sim S_U$:
5: calculate the loss function L_G in Eq. (12)
6: min(L) in Eq. (14) to update the parameters θ_{NER} and θ_G
7: min(L_D) in Eq. (13) to update the parameters θ_D
8: calculate sim(s^L,s^U) with the well trained D , and annotate the s^U with low similarity score
9: add the new labeled s to S_L,
10: implement the step 3 to retrain the NER model
11: **end for**
12: **return** well-trained NER model

4 Experiment

4.1 Experiment Setting

The data in the experiment was collected from two sources: With the cybersecurity corpora released in Semeval-2018 Task 8, we selected 500 sentences related to malware, which were annotated and used for initially training the proposed dynamic-attention-based BiLSTM-LSTM model on the NER task before using the adversarial active learning algorithm. Then, we selected and annotated 1464 sentences as the testing set; in addition, we selected 9175 threat intelligence sentences from the AlienVault community, WeLiveSecurity community, Amazon security-related blogs and APT reports of the past two years. We annotated such cybersecurity sentences and added them to the initial training set, which we used for evaluating the performance of the proposed adversarial active learning model.

Note that when annotating the cybersecurity entity in sentences, we refer to the entity types defined in the Unified Cybersecurity Ontology (UCO 2.0) [2] and implemented the annotation referring to the entity types, including organization, location, software, malware, indicator, vulnerability, course-of-action, tool, attack-pattern, and campaign.

In our proposed model, the dimension of word embedding is set to 100, and the dimension of char embedding is set to 25. In addition, to obtain the char embedding, the number of CNN filters is set to 20, and the size of CNN kernels is set to 3. In the feature encoding layer, the dimensions of both the forward LSTM and reverse LSTM are set to 300. And we utilize dropout in the BiLSTM layer to mitigate the overfitting. In tag decoding layer, the dimension of LSTM is set to 600. In model training, the number of epochs is set to 100 and the batch size is set to 128. Furthermore, we train the model by stochastic gradient descent with the initial learning rate of 0.001. The hyperparameters are shown in Tab. 1.

For evaluation, we adopted the conventional criteria for information extraction, including precision (P), recall (R) and F1-score. And F1-score was used to evaluate the comprehensive performance of model.

4.2 NER Performance Comparison

We first evaluated the proposed novel NER module, called Dynamic-att-BiLSTM-LSTM, which was trained on the full training set and then compared with four mainstream NER models: CRF, BiLSTM-

CRF, self-attention based BiLSTM-CRF, and self-attention based BiLSTM-LSTM. The results of the performance comparison are shown in Tab. 2.

Table 1: Hyperparameters of the proposed model

Description of parameters	Value
word_embedding_dim	100
char_embedding_dim	25
cnn_filter_num	20
cnn_kernel_size	3
bilstm_e_dim	300
lstm_d_dim	600
epoch	100
batch_size	128
learning_rate	0.001
dropout_rate	0.5

Table 2: Performance comparison of NER models

Model	Char embedding	P/%	R/%	F1/%
CRF	×	78.69	77.13	77.9
BiLSTM-CRF	×	81.76	79.92	80.83
BiLSTM-CRF	○	84.53	83.35	83.94
Self-att-BiLSTM-CRF	○	86.38	84.24	85.3
Self-att-BiLSTM-LSTM	○	87.07	84.45	85.74
Dynamic-att-BiLSTM-LSTM	○	89.62	87.63	88.61

Note: "○" indicates that the model use char embedding; "×" indicates that the model does not use char embedding.

As can be seen in the results, when BiLSTM is combined with the CRF, in which BiLSTM is used as the feature encoder and CRF is used as the tag decoder, the combined model performs better than the single CRF model, which may be due to its resolution of the long-distance dependency of BiLSTM. Then, with the char-level embedding applied to NER, the performance of the BiLSTM-CRF model is enhanced, which shows the significance of character embedding for identifying the specific token. In addition, when we add the self-attention mechanism to the BiLSTM-CRF model, it turns out that self-attention contributes to the enhancement for entity recognition performance. To demonstrate the effectiveness of LSTM for tag decoding in the self-attention-BiLSTM framework, we compare the LSTM decoder with the conventional chain CRF. It can be seen that LSTM is lightly better for tag decoding. Our proposed Dynamic-att-BiLSTM-LSTM model outperforms other NER models, achieving the best F1-score of 88.61%, which shows that the proposed dynamic attention mechanism can capture more precise dependencies between two tokens.

4.3 Performance of Adversarial Active Learning

We next evaluated the effectiveness of our proposed active learning algorithm. We used the proposed adversarial active learning to select samples which were then annotated by the oracle. Then the NER model was incrementally retrained on the selected set with 100 training epochs. The performance of the proposed NER model trained on the labeled set, which was sampled by the adversarial active learning algorithm, was compared with the performance by using the full labeled training set. The results according to the size of the labeled set are shown in Tab. 3. We can see with the increase in the number of labeled samples, the model's performance improves. When the sampled data size reaches 45% of the full labeled training set, we achieve a good F1-score of 88.27%, which is only 0.34% less than when the full training set is used, again showing the effectiveness of the proposed adversarial active learning.

Table 3: Performance comparison of different size of labeled set

Size of labeled set	P/%	R/%	F1/%
Full labeled set	89.62	87.63	88.61
10% of labeled set	55.27	52.95	54.09
20% of labeled set	75.31	73.03	73.66
30% of labeled set	84.72	82.39	83.54
40% of labeled set	88.75	85.82	87.3
45% of labeled set	89.47	87.1	88.27

We also compared the proposed adversarial active learning model with three conventional uncertainty sampling-based active learning algorithms, which are based on LC, BALD and MNLP. By using these active learning algorithms to select the samples to be annotated, we were able to compare the performance of the proposed NER model based on different sizes of labeled data. We recorded the comparison results till the size of the labeled set reached 45%. The performance comparison is shown in Fig. 4. From the results, we can see that our proposed adversarial active learning algorithm outperforms the other three methods. Such results may be due to the complex computation of the three methods when used in sequence labeling task, which would lead to the inaccurate sampling results. In addition, both MNLP and BALD methods perform better than the LC-based active learning, with MNLP achieving slightly better results than BALD. However, in general, the performance of MNLP and BALD in active learning are quite close to each other. The results show our proposed adversarial active learning method can be used to incrementally improve the performance of the NER task with low labeling cost.

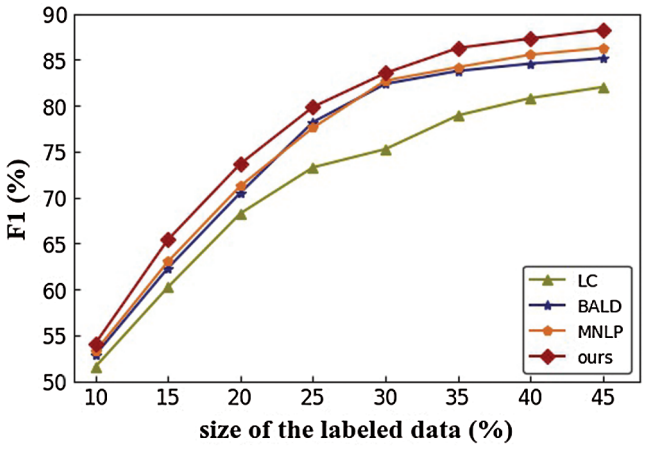

Figure 4: Performance comparison of the active learning algorithms on the NER task

5 Conclusion

Named entity recognition (NER) in cybersecurity is a fundamental task for constructing cybersecurity knowledge graph, which is significant for data-driven proactive cybersecurity. Representative deep learning-based NER models have recently achieved promising performance when there is a profusion of labeled data. However, it is difficult to obtain massive amounts of labeled data for NER in cybersecurity. In addition, the traditional uncertainty-based active learning algorithms are complex when applied to the sequence data. To address the issue, this paper proposes an adversarial active learning framework to incrementally improve the performance of the NER model with low labeling cost. Moreover, a novel dynamic attention-based BiLSTM-LSTM model is presented for the NER task. The model presents a dynamic attention mechanism to adaptively capture the dependency between two tokens, and employs an LSTM decoder for entity tag decoding. Finally, we evaluate our proposed model through a series of comparison experiments. The results show that the proposed NER model attains better performance, and the proposed adversarial active learning scheme is effective in incrementally selecting informative samples. In future work, we would like to introduce the syntactic feature into the model to further enhance the model's performance. In addition, we plan to build a specific domain dictionary with expert knowledge to rectify the extracted cybersecurity entities.

Acknowledgement: The authors thank the editor and the reviewers for their hard work.

Funding Statement: This work was supported by the National Natural Science Foundation of China under grant 61501515.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Phandi, A. Silva and W. Lu, "Semeval-2018 Task 8: Semantic extraction from cybersecurity reports using natural language processing (SecureNLP)," in *Proc. of the 12th Int. Workshop on Semantic Evaluation*, New Orleans, LA, USA, pp. 697–706, 2018.

[2] A. Pingle, A. Piplai, S. Mittal, A. Joshi, J. Holt *et al.,* "RelExt: Relation extraction using deep learning approaches for cybersecurity knowledge graph improvement," in *Proc. of the IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, New York, NY, USA, pp. 879–886, 2019.

[3] Z. Jin, Y. Zhang, H. Kuang, L. Yao, W. Zhang *et al.,* "Named entity recognition in traditional Chinese medicine clinical cases combining BiLSTM-CRF with knowledge graph," in *Int. Conf. on Knowledge Science, Engineering and Management*, Athens, Greece, pp. 537–548, 2019.

[4] S. Sinha, S. Ebrahimi and T. Darrell, "Variational adversarial active learning," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Seoul, Korea, pp. 5972–5981, 2019.

[5] J. Li, A. Sun, J. Han and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[6] H. Gasmi, J. Laval and A. Bouras, "Information extraction of cybersecurity concepts: An LSTM approach," *Applied Science*, vol. 9, no. 9, pp. 3945, 2019.

[7] Z. Long, L. Tan, S. Zhou, C. He and X. Liu, "Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labelling," in *Int. Joint Conf. on Neural Networks*, Budapest, Hungary, pp. 1–8, 2019.

[8] T. Satyapanich, F. Ferraro and T. Finin, "CASIE: Extracting cybersecurity event information from text," in *Proc. 34th Association for the Advancement of Artificial Intelligence*, *New York, USA*, 2020.

[9] N. Dionísio, F. Alves, P. M. Ferreira and A. Bessani, "Cyberthreat detection from twitter using deep neural networks," in *Int. Joint Conf. on Neural Networks*, Budapest, Hungary, pp. 1–8, 2019.

[10] Z. Tang, L. Jiang, L. Yang, K. Li and K. Li, "CRFs based parallel biomedical named entity recognition algorithm employing MapReduce framework," *Cluster Computing*, vol. 18, no. 2, pp. 493–505, 2015.

[11] M. Grégoire, X. He, D. Li and B. Yoshua, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. Conf. of the Int. Speech Communication Association*, Lyon, France, pp. 3771–3775, 2013.

[12] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou *et al.,* "Joint extraction of entities and relations based on a novel tagging scheme," in *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, *Stroudsburg*, pp. 1227–1236, 2017.

[13] F. Zhai, S. Potdar and B. Xiang, "Neural models for sequence chunking," in *Proc. of 31th Association for the Advance of Artificial Intelligence*, San Francisco, CA, USA, pp. 3365–3371, 2017.

[14] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod and A. Anandkumar, "Deep active learning for named entity recognition," in *Int. Conf. on Learning Representations*, Vancouver, Canada, pp. 1–15, 2018.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley *et al.,* "Generative adversarial networks," in *Proc. of the Advances in 27th Neural Information Processing Systems*, Montreal, QC, Canada, pp. 2672–2680, 2014.

[16] T. Gui, Q. Zhang, H. Huang, M. Peng and X. Huang, "Part-of-speech tagging for twitter with adversarial neural networks," in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 2411–2420, 2017.

[17] D. Zeng, Y. Dai, F. Li, R. S. Sherratt and J. Wang, "Adversarial learning for distant supervised relation extraction," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 121, 2018.

[18] L. Yang, Y. Zhang, J. Chen, S. Zhang and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *Medical Image Computing and Computer Assisted Intervention*, Quebec City, QC, Canada, pp. 399–407, 2017.

[19] J. Zhu and J. Bento, "Generative adversarial active learning," arXiv: 1702.07956v5, 2017.

[20] K. Wang, D. Zhang, Y. Li, R. Zhang and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2017.

[21] V. Chandola, A. Banerjee and V. Kumar, "Active learning theory," in *Encyclopedia of Machine Learning and Data Mining*. Berlin, Germany: Springer, pp. 9–19, 2017.

[22] B. Pradhan, M. I. Sameen and B. Kalantar, "Ensemble disagreement active learning for spatial prediction of shallow landslide, " in *Laser Scanning Applications in Landslide Assessment*. Berlin, Germany: Springer, pp. 179–191, 2017.

[23] S. Mizokami, "Deep active learning from the perspective of active learning theory, " in *Deep Active Learning*. Berlin, Germany: Springer, pp. 79–91, 2017.

[24] O. Sener and S. Savarese, "Active learning for convolutional neural networks: a core-set approach," in *Int. Conf. on Learning Representation*, Vancouver, Canada, pp. 1–13, 2018.

[25] N. Houlsby, F. Huszar and Z. Ghahramani, "Bayesian active learning for classification and preference learning," arXiv: 1112.5745, 2011.

[26] Y. Gal, R. Islam and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Int. Conf. on Machine Learning*, Sydney, NSW, Australia, pp. 1183–1192, 2017.

[27] W. Kuo, C. Hane, E. Yuh, P. Mukherjee and J. Malik, "Cost-sensitive active learning for intracranial hemorrhage detection," in *Proc. of Medical Image Computing and Computer Assisted Intervention*, Granada, Spain, pp. 715–723, 2018.

[28] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Transactions of the Association for Computational Linguistics*, vol. 4, no. 1, pp. 357–370, 2016.

[29] C. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Int. Conf. on Machine Learning*, Beijing, China, pp. 1818–1826, 2014.

[30] P. Cao, Y. Chen, K. Liu, J. Zhao and S. Liu, "Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 182–192, 2018.

[31] M. Cheng, M. Hong, J. Tang, J. Zhang, B. Zou *et al.,* "Gated dynamic attention mechanism towards aspect extraction," *Pattern Recognition and Artificial Intelligence*, vol. 32, no. 2, pp. 90–98, 2019.