Array 23 (2024) 100364

Contents lists available at ScienceDirect

Array

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

Threat intelligence named entity recognition techniques based on few-shot learning

Haiyan Wang [a], Weimin Yang [b], Wenying Feng [a], Liyi Zeng [a], Zhaoquan Gu [a,c,*]

[a] Department of New Networks, Peng Cheng Laboratory, Shenzhen, PR China
[b] Huawei Technologies CO. LTD, Shenzhen, PR China
[c] Harbin Institute of Technology, Shenzhen, PR China

ARTICLE INFO

Keywords:
Named entity recognition
Few-shot learning
Threat intelligence
Multi-view learning

ABSTRACT

In today's digital and internet era, threat intelligence analysis is of paramount importance to ensure network and information security. Named Entity Recognition (NER) is a fundamental task in natural language processing, aimed at identifying and extracting specific types of named entities from text, such as person names, locations, organization names, dates, times, currencies, and more. The quality of entities determines the effectiveness of upper-layer applications such as knowledge graphs. Recently, there has been a scarcity of training data in the threat intelligence field, and single models suffer from poor generalization ability. To address this, we propose a multi-view learning model, named the Few-shot Threat Intelligence Named Entity Recognition Model (FTM). We enhance the fusion method based on FTM, and further propose the FTM-GRU (Gate Recurrent Unit) model. The FTM model is based on the Tri-training algorithm to collaboratively train three few-shot NER models, leveraging the complementary nature of different model views to enable them to capture more threat intelligence domain knowledge at the coding level.FTM-GRU improves the fusion of multiple views. FTM-GRU uses the improved GRU model structure to control the memory and forgetting of view information, and introduces a relevance calculation unit to avoid redundancy of view information while highlighting important semantic features. We label and construct a few-shot Threat Intelligence Dataset (TID), and experiments on TID as well as the publicly available National Vulnerability Database (NVD) validate the effectiveness of our model for NER in the threat intelligence domain. Experimental results demonstrate that our proposed model achieves better recognition results in the task.

1. Introduction

With the rapid development of information technologies, the network environment has become increasingly complex. Cybersecurity events, such as eavesdropping and data leakage, occur frequently and seriously endanger the security of society, and cybersecurity has become the focus of current research [1]. Named entity recognition (NER) can be used to process the massive amount of fragmented data generated by these events and collect comprehensive, accurate, and actionable threat intelligence information.

NER is essentially a sequence annotation problem for identifying phrases with special meanings in sentences, which is a fundamental task in Natural Language Processing (NLP) tasks such as Information Retrieval [2], Question and Answer Systems [3], and Machine Translation [4]. It plays a crucial role in the field of NLP as a fundamental work. The NER techniques in the field of threat intelligence can extract the problem of important security entity information from the unstructured text of threat intelligence and help security experts respond to cybersecurity incidents promptly, which is important for the research of cybersecurity.

In recent years, deep learning techniques have been widely used in threat intelligence NER because of their ability to better capture potential semantic information through deep network structures, and some researchers have achieved breakthrough results [5–7].

Although the NER method based on deep learning can achieve more satisfactory results, the performance of such methods relies heavily on a large amount of annotated corpus, and it is difficult to obtain sufficient training data in the threat intelligence domain where annotated data is lacking.

In the research of few-shot NER, although the research of a single model has made great progress, the overall model effect is still far from that of deep learning models trained with large amounts of data. This is mainly because single models do not sufficiently learn features in few-shot scenarios, which can easily cause misclassification in entity classification decisions. Simply fusing multiple views will

* Corresponding author.
E-mail address: guzhq@pcl.ac.cn (Z. Gu).

https://doi.org/10.1016/j.array.2024.100364
Received 25 April 2024; Received in revised form 4 September 2024; Accepted 4 September 2024
Available online 12 September 2024

cause redundancy of sample information, which cannot make full use of view information and affect the calculation of feature weights, resulting in learning bias. There is still much room for improving the effect of few-shot learning models.

In this paper, to address the scarcity of training data in the field of threat intelligence and the poor generalization ability of single models, we have adopted a specific multi-view learning model, named the Few-shot Threat Intelligence Named Entity Recognition Model (FTM). This model leverages the complementary nature of different model views to capture more knowledge in the threat intelligence domain at the encoding level, thereby improving the performance of the Named Entity Recognition (NER) task. We have specifically experimented with the NER task in the few-shot threat intelligence domain and made improvements and refinements in both model architecture and multi-view fusion. Our contributions are as follows:

- We propose a new few-shot threat intelligence named entity recognition model (FTM). The FTM model achieves co-training of the prototype network, pre-trained model, and self-trained model through the Tri-training algorithm, and exploits the complementary nature of the three different model views to capture more threat intelligence domain knowledge at the encoding level.
- We improve the FTM model and propose a few-shot threat intelligence named entity identification model based on an improved GRU fusion approach (FTM-GRU). We simplify the structure of the GRU model while inputting the view features extracted by the three models into the improved GRU gating unit to avoid the redundancy of view information and highlight important semantic features, which further improves the effectiveness of the few-shot threat intelligence NER task.
- We construct a NER dataset for the few-shot threat intelligence domain, which provides an operational procedure and a referenceable idea for constructing a few-shot threat intelligence domain dataset.

This paper is organized as follows: Section 2 introduces the related work, Section 3 describes the principle and calculation method of the model in this paper, Section 4 explains the construction method of the dataset, the experimental procedure and experimental results, and Section 5 summarizes.

2. Related work

Few-shot learning refers to the use of a small number of learning samples to train the classifier, and is used in tasks including image classification, sentiment analysis and NER [8]. During the training phase, the model acquires the capability to recognize named entities through interaction with the query set. The support set further reinforces this learning process, while the validation set assesses the proficiency of the model's named entity recognition skills. The test phase simulates the prediction for other samples with the same label space when the labels of a few-shot of each category are known [9].

Many techniques for few-shot NER have been investigated, including metric learning, self-training, and pre-training methods. Metric learning aims to compute the similarity between elements using distance metric functions or deep learning methods, and the similarity can be applied to NER tasks. Snell et al. [10] applied the prototype network to few-shot learning tasks, where the classification of entities is completed by computing the distance to the prototype representation of each class. They demonstrate in their experiments that simple classification decisions can outperform complex structural models and are beneficial in few-shot scenarios, while further extending the prototype network to zero-shot learning.

Yang et al. [11] constructed the StructShot model based on nearest neighbor and structured inference. They argue that O class entities cannot be simply represented by prototypes because O class entities carry a lot of noise that interferes with the judgment of the model. They abandon the approach of constructing prototypes for each entity type in the prototype network, and each entity is classified based on the principle of nearest neighbors on vector space and predicted using the Viterbi algorithm. This approach avoids the noise interference brought by O class entities and achieves the best results at that time.

Self-training, as a representative of semi-supervised learning, is widely used for few-shot learning. Ye et al. [12] pointed out that when sampling data for training, traditional self-training methods use a fixed exploratory algorithm that performs differently on different datasets while consuming a lot of exploration time. Based on the self-training approach, they applied a reinforcement learning framework in the selection strategy and model training, which enabled the model to automatically select data with higher confidence for training and mitigated error propagation. Zoph et al. [13], in their study of the results of pre-training and self-training, find that self-training is always helpful in improving the model effectiveness, both with high and low resources. Meanwhile, They find that self-training can further improve the model performance when pre-training is useful. Chen et al. [14] used self-training, combined with prompt-based supervised learning and self-supervised learning, to generate two perspectives, weakly and strongly augmented, by different training strategies, which enabled the model to perform well with a small amount of unlabeled data and labeled data scenarios.

In addition, the pre-trained language model can effectively alleviate the problem of insufficient learning due to the lack of data in the target domain by migrating the knowledge from the source domain to the target domain. The more popular ones in the NER domain are BERT [15], RoBERTa [16], and ALBERT [17], where pre-trained models learn information features in a large amount of unlabeled data and then migrate the learned knowledge to the downstream task, requiring only a small amount of data sets from the downstream task to fine-tune the pre-trained model so that the model can be better adapted to the linguistic features of the current task.

To address the problem of poor generalization of a single view, researchers have proposed multi-view learning, which has greatly advanced academic research and industry [18].

Ding et al. [19] propose a joint multi-view character embedding model for Chinese NER (JMCE-CNER) of car reviews, deeper character features are extracted from pronunciation, radical, and glyph views to generate the multi-view character embedding. JMCE-CNER has an enhanced effect on NER tasks. Annamoradnejad et al. [20] propose a multi-view approach, which generates three different feature groups to check the problem from three different perspectives, and the solution can replace manual moderation. Gonçalves et al. [21] use a stacking generalization based on the idea that different learning algorithms provide complementary explanations of the data. Experimental results lead to the conclusion that the application of multi-view techniques to full texts significantly improves the task of text classification. Raeed et al. [22] propose MVGNAS, a multi-view graph neural network automatic modeling framework for biomedical entity and relationship extraction that considers the interactions between nodes and multiple relationships to improve the performance metrics of entity and relationship extraction tasks. Sun et al. [23] propose a novel multi-view CRF model to label sequential data, called MVCRF, which exploits two principles for multi-view learning: consensus and complimentary.

Although there are significant differences in the approaches to integrate multiple views to improve learning performance, they mainly use the consensus principle or the complementarity principle to ensure the success of multi-view learning [24–27]. Therefore, we propose a new multi-view fusion approach, and in this paper, we improve the GRU model structure,[1] use the improved GRU model structure to

[1] The GRU, which has been improved in our structure, needs to be used in conjunction with FTM, because it is meaningless to use this enhanced GRU model alone for Few-Shot NER tasks.

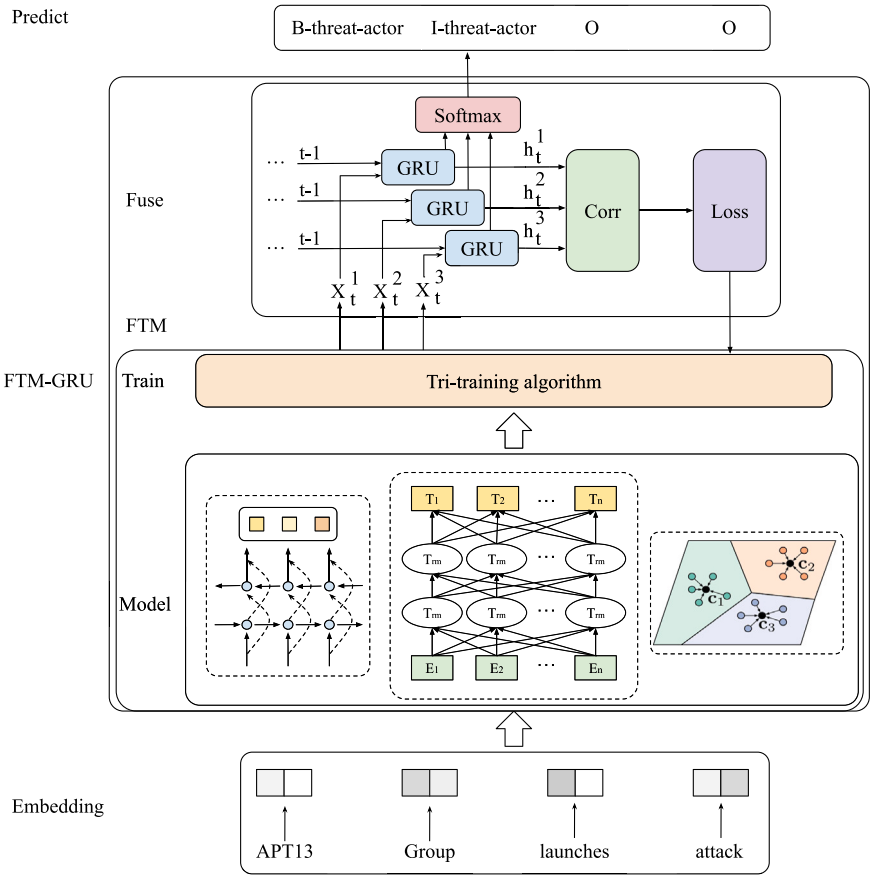

Fig. 1. FTM-GRU architecture.

control the memory and forgetting of view information, and introduce a relevance calculation unit to avoid redundancy of view information while highlighting important semantic features.

3. Main result

3.1. FTM-GRU framework

We propose a new few-shot threat intelligence named entity recognition model (FTM) for the problem of low data volume and poor recognition effect of a single model in the threat intelligence domain. Based on the Tri-training algorithm [28], the model integrates a prototype network model, a pre-trained BERT model, and a self-trained BiLSTM-CRF model [29–31], which effectively learns the semantic information obtained from different views and improves the threat intelligence entity recognition capability through the complementarity between views.

The FTM-GRU model adds a modified GRU module for multi-view fusion, while adding the consistency of the hidden states to the calculation of the loss function. The three view types are input to the modified GRU, and then the gating unit determines the memory and forgetting of the view information. The relevance calculation unit is introduced to avoid redundancy of view information while highlighting important semantic features. The overall model framework of FTM-GRU is shown in Fig. 1.

The FTM-GRU model loss function is set as shown in Eq.

$$L_{FTM\_GRU} = L_{FTM} + \lambda L_{corr}, \tag{1}$$

where L_{FTM} denotes the loss function of the FTM model, L_{corr} is the defined loss term, and λ is the weight parameter that represents the contribution of L_{corr} to the total loss.

3.2. FTM model

The FTM model first samples the data to construct few-shot scenario training data, initializes the weights of the three types of models using the support set data, and then uses the Tri-training algorithm to co-train the three types of models. In the training process, the overall loss function used is:

$$L_{FTM} = \alpha L_1 + \beta L_2 + \gamma L_3, \tag{2}$$

where α, β, γ are learnable parameters that denote the loss weights of the three models, and L_1, L_2, L_3 are the loss functions of the three sub-models, respectively, which are described in detail in the following content.

3.2.1. Prototype network

The prototype network computes the m-dimensional representation $c_k \in R^M$ of each class by the embedding function $f_\theta(x_i)$, calling c_k the prototype of each class. Where θ is the learnable parameter. Each prototype is the mean vector of embedding support points belonging to its class.

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\theta\left(\mathbf{x}_i\right). \tag{3}$$

Given a distance function d, the prototype network calculates the prototype distance of x in the embedding space based on the query sample x, and then derives the probability distribution of entity types based on softmax.

$$p_\theta(y = k \mid \mathbf{x}) = \frac{\exp\left(-d\left(f_\theta(\mathbf{x}), \mathbf{c}_k\right)\right)}{\sum_{k'} \exp\left(-d\left(f_\theta(\mathbf{x}), \mathbf{c}_{k'}\right)\right)}. \tag{4}$$

The prototype network loss function is designed as

$$L_1 = \frac{1}{N_C N_Q}[d\left(f_\theta(\mathbf{x}), \mathbf{c}_k\right) + \log \sum_{k'} \exp\left(-d\left(f_\theta(\mathbf{x}), \mathbf{c}_k\right)\right)]. \tag{5}$$

3.2.2. BiLSTM-CRF self-training

The flow of the BiLSTM-CRF self-training method is as follows. Input a small amount of labeled dataset L, a large amount of unlabeled dataset U, and a BiLSTM-CRF classifier C to be self-trained. Firstly, the classifier C is trained using the dataset L, and determines whether the training reaches the stopping criterion, if not, the trained classifier C classifies part of the unlabeled data U, selects The unlabeled sample S and the classified pseudo-label are jointly added to the labeled dataset L for training. If the stopping criterion is reached, the classifier C is output and the training is terminated.

The BiLSTM+CRF self-training loss function is formulated as follows:

$$L_2 = -\sum_{x,y} \widetilde{P}(x, y) \log P(y \mid x). \tag{6}$$

3.2.3. BERT

To capture the context of a word, BERT abandons the traditional one-way cyclic structure similar to the RNN model and adopts a bidirectional Transformer encoding structure to model the text data based on multi-head attention. Each Transformer encoder contains an attention calculation module, and the attention value calculation formula is shown as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V. \tag{7}$$

The loss function used by BERT in the training process is a negative log-likelihood function, which is calculated as follows:

$$L_3 = -\sum_{x,y} \widetilde{P}(x, y) \log P(y \mid x). \tag{8}$$

3.2.4. Tri-training

The tri-training algorithm is described as follows:

Let L be the labeled dataset and U be the unlabeled dataset, and perform bootstrap sampling with put-back on the dataset L to obtain three different datasets, and train three different models M_1, M_2, M_3 respectively. After training, let two of the models M_2, M_3 re-predict the dataset L and calculate the classification error rate e^i, which is expressed by the following equation:

$$e^i = \frac{\sum P_{error}}{\sum P_{same}}, \tag{9}$$

where P_{same} denotes samples with the same prediction by both classifiers, and P_{error} denotes samples with the same prediction but wrong prediction by both classifiers. Suppose the current classification error rate is less than the last classification error rate $e^i < e^{i-1}$, it means the model is improving, sample part of the unlabeled dataset u, let the model M_2, M_3 label the dataset u^i, and if the labeled result $M_2^y = M_3^y$, then the sample and the pseudo-label are added to the labeled dataset, i.e. $L_1^i \cup u^i \rightarrow L_1^{i+1}$. Marking the update flag as True, the model is rearranged and combined, and the above operation is repeated twice. If the update flag is False all three times, the iteration ends and the three trained models are output.

3.3. GRU fusion module

3.3.1. GRU

The GRU unit is used as the basis of the experiment. To reduce the computational effort, the computational units in the original GRU module are streamlined and the update gate and reset gate are retained. The internal structure of the improved GRU is shown in Fig. 2. The GRU unit is applied to the three-view fusion. Each view information is passed through a separate GRU gating to calculate a multi-view embedding of the time steps. The coded output of the GRU is used to calculate the correlation between the different views, and its output is added to the calculation as a separate term in the total loss.

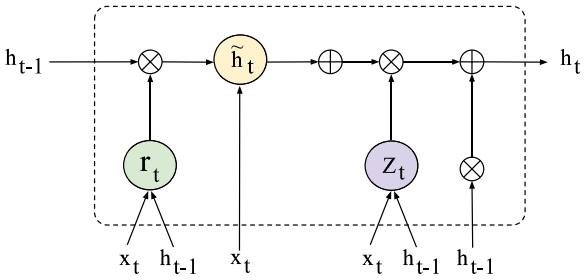

Fig. 2. Improved GRU internal calculation unit.

The GRU module takes as input the encoded state x_t and the label y_t of the FTM at time t, the hidden state h_t^i of model i at time t and the current input x_t concerning the hidden state h_{t-1}^i at the previous moment. x_t is reserved for calculating the correlation between views. The formulas for $r_t^i, z_t^i, \tilde{h}_t^i$ are as follows.

$$r_t^i = \sigma\left(\mathbf{W}_r^i X_t^i + \mathbf{U}_r h_{t-1} + b_r^i\right), \tag{10}$$

$$z_t^i = \sigma\left(\mathbf{W}_z^i X_t^i + \mathbf{U}_z h_{t-1} + b_z^i\right), \tag{11}$$

$$\tilde{h}_t^i = \varphi\left(\mathbf{W}_h^i X_t^i + \mathbf{U}_h\left(r_t^i \odot h_{t-1}\right) + b_h^i\right), \tag{12}$$

where z and r are the inputs to the update and reset gates, $\tilde{h}$ denotes the temporary candidate hidden state during the computation, and h denotes the standard hidden state, σ is the sigmoid function, and φ is the hyperbolic tangent (tanh) function.

3.3.2. Correlation

The purpose of adding correlation to the loss is that correlation can take advantage of the labels available in the dataset while optimizing the correlation between the available views, and the correlation constraint will result in an embedding that contains more information than if only cross-entropy losses were used.

Since the three views are combined in different ways and correlation can only be computed for two variables, the total correlation loss is computed in this paper as the sum of the correlation losses between all view pairs. In each step t, the correlation between views is computed as.

$$c_t = \sum_{\substack{k,\ell \in \mathcal{V} \\ k \neq \ell}} \frac{\sum_{i=1}^{L}\left(h_{it}^{(k)} - \bar{H}_t^{(k)}\right)\left(h_{it}^{(\ell)} - \bar{H}_t^{(\ell)}\right)}{\sqrt{\sum_{i=1}^{L}\left(h_{it}^{(k)} - \bar{H}_t^{(k)}\right)\sum_{i=0}^{L}\left(h_{it}^{(\ell)} - \bar{H}_t^{(\ell)}\right)}}, \tag{13}$$

where there are L samples on each minibatch, i denotes the sample, k is the view, t is the time step, $h_{it}^{(k)}$ is the hidden state t computed specific to the view GRU, and $\bar{H}_t^{(k)}$ denotes the mean value of the hidden state, which is calculated as follows:

$$\bar{H}_t^{(k)} = \frac{1}{L}\sum_{i=1}^{L} h_{it}^{(k)}. \tag{14}$$

The average of the losses computed at each time step t. To maximize the correlation, the negative sign needs to be added. There are three different combinations of the three models in two combinations, and a correction is made by adding a factor 3 to the equation so that the correlation term in the loss is expressed as:

$$L_{corr} = -\frac{1}{3|T|}\sum_{t \in T} c_t^{i,j} \quad (i \neq j \text{ and } i, j \in \{1, 2, 3\}). \tag{15}$$

Table 1
15 entity type definitions.

	Label		Label		Label
1	Malware	6	22.16	11	Vulnerability
2	Family	7	Motivation	12	Tool
3	Threat-actor	8	Campaign	13	IoC
4	Identity	9	Transmission	14	Industry
5	Time	10	Asserts	15	Defense

ABK is a downloader that has been used by BRONZE BUTLER since at least 2019.
malware family threat-actor time

Fig. 3. Example of Doccano annotation result.

During training, the network is trained by jointly minimizing the following objective functions, and the overall loss function of the FTM-GRU model is set as shown in Eq. (16):

$$L_{FTM\_GRU} = L_{FTM} + \lambda L_{corr}, \tag{16}$$

where L_{FTM} denotes the loss function of the FTM model, calculated as (2), L_{corr} is the loss term defined by the formula (15), and λ is the weight parameter indicating the contribution of L_{corr} to the total loss.

4. Experiment

4.1. Few-shot dataset

We refer to threat intelligence specifications such as STIX, MAEC, and CAPEC, and combine them with expert recommendations to define 15 entity types. Compared to other standards, this paper defines many entity types with finer granularity to accurately describe threat intelligence information. The entity definitions are shown in the Table 1.

Advanced Persistent Threat (APT) has the characteristics of being difficult to detect, long latency time and high damage. Therefore, we collect threat intelligence from APT attack reports over the years, crawling a total of 502 APT attack reports from 2006 to 2022. Preprocess the file first and use the PDFMiner tool to convert the report into a plain text file. To condense the amount of information in the sentences and avoid introducing redundant information, we only intercept the text content of the summary and the previous introduction part, and then divide the sentences. Some special symbols and unrecognized characters will still appear in the recognized files, and sentences containing these characters will be screened out in bulk. Due to the excessive amount of data, only a small portion of the data is taken for annotation. To ensure data diversity, the results from the above operation are sampled for annotation. Up to two reports were randomly selected for annotation each year, and a total of 28 APT reports were annotated. The remaining pre-processed partial unannotated reports were used as unlabeled datasets for the BiLSTM+CRF model self-training and Tri-training algorithm.

Doccano is an open source annotation tool that provides annotation functions for a variety of NLP tasks, including text classification, sequence annotation, and sentiment analysis, among other tasks. Using Doccano to annotate the sampled data, the annotation example is shown in Fig. 3.

The annotation results of Doccano are saved in BIO format, and each of the 15 types of entities is annotated with "B-" at the beginning and "I-" at the middle or end. In addition, the other entity types are labeled with "O". The labeled data and the pre-processed unlabeled data together build the Threat Intelligence Dataset (TID) and the statistics of the number of labeled entities in the TID are shown in Fig. 4.

Three few-shot scenarios, 5-way 5-shot, 5-way 10-shot and 10-way 10-shot, were constructed using the N-way K-shot method for sampling. Also to validate the validity of the model, the same operation was performed on the NVD dataset [28].

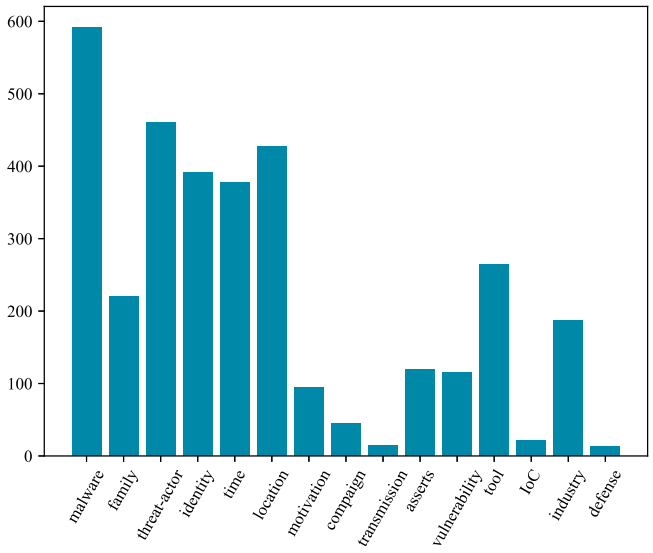

Fig. 4. Distribution of the number of 15 entities.

Table 2
F1 score of different models on TID dataset.

	F1 score on TID datasets		
	5way-5shot	5way-10shot	10way-10shot
BiLSTM-CRF	17.56	24.60	22.16
BERT	25.79	34.61	30.13
Proto	29.43	35.87	32.57
BERT + BiLSTM-CRF	32.67	37.87	34.67
Proto + BiLSTM-CRF	34.27	38.57	36.12
Proto + BERT	35.78	40.51	38.90
Baseline	36.42	42.71	40.38
NNShot	34.72	40.67	39.09
StructShot	37.48	43.25	39.12
FTM(ours)	37.61	44.56	42.79
FTM-GRU(ours)	**40.28**	**49.12**	**43.78**

4.2. Result

To verify the validity of the models, some comparison experiments are done in this paper. The prototype network (Proto), pre-trained model (BERT) and self-trained model (BiLSTM-CRF) are predicted by a single model and simple combined approach, and the model parameters are kept uniform in both approaches. The combined models are each trained separately during the training process, while the prediction is done by weighted voting. The Baseline method was set up to predict the voting after training for each of the three models mentioned above, and additionally using NNShot [14] and StructShot [14] as a comparison. Three sets of training data samples were set up, including three few-shot scenarios of 5-way 5-shot, 5-way 10-shot and 10-way 10-shot. The evaluation metrics of the models used F1 score, and the test results of each model after TID and NVD training are shown in Tables 2 and 3.

The results show that FTM-GRU achieves the best entity identification results on both threat intelligence datasets. Meanwhile, the FTM-GRU model, with the addition of a multi-view fusion approach and consistency calculation, can better perform the few-shot NER task in the threat intelligence domain, and the experiments prove the effectiveness of the model.

The recognition effect of each entity type is further analyzed by taking a 5-way 10-shot as an example. Table 4 records the classification effects of the Baseline method and the proposed FTM-GRU model on each entity in the NVD dataset. The FTM-GRU model outperforms the identification in each entity through the Baseline method. The experimental results show that the FTM-GRU model can effectively

Table 3
F1 score of different models on NVD dataset.

	F1 score on NVD datasets		
	5way-5shot	5way-10shot	10way-10shot
BiLSTM-CRF	6.71	14.38	8.81
BERT	27.63	33.79	32.97
Proto	24.81	38.12	36.02
BERT + BiLSTM-CRF	32.67	39.54	39.78
Proto + BiLSTM-CRF	33.67	39.14	35.12
Proto + BERT	30.12	41.56	32.69
Baseline	42.52	46.48	46.12
NNShot	35.67	42.50	40.18
StructShot	42.40	45.20	44.85
FTM(ours)	44.56	47.13	46.14
FTM-GRU(ours)	**48.13**	**51.42**	**51.08**

Table 4
F1 score of different entities in the NVD dataset.

Entity	F1 score	
Type	Baseline	FTM-GRU
Malware	45.12	**50.98**
Family	45.56	**49.34**
Threat-actor	25.20	**34.50**
Identity	31.22	**37.32**
Time	56.90	**64.02**
Location	51.78	**52.16**
Motivation	22.14	**27.66**
Campaign	36.06	**37.48**
Transmission	26.78	**34.04**
Asserts	43.30	**47.78**
Vulnerability	59.60	**62.12**
Tool	41.10	**44.20**
IoC	24.22	**26.14**
Industry	39.26	**42.20**
Defense	39.40	**42.68**

capture the features at the encoding level of the threat intelligence domain and has an enhanced effect on the recognition of named entities in the threat intelligence domain.

5. Conclusion

In this paper, we proposed a multi-view fusion method based on an enhanced GRU structure, where three distinct views are fed into the GRU. A gating unit is employed to regulate the retention and discard of view information. Additionally, a relevance calculation unit is incorporated to mitigate redundancy while accentuating crucial semantic features. Experiments conducted on the dataset proposed in this paper, as well as publicly available NVD datasets, demonstrate the effectiveness of the FTM-GRU model in better performing few-shot NER tasks in the threat intelligence domain. Nevertheless, the current approach is more modular and entails lengthier computational time. Thus, a potential avenue for future research lies in further optimizing the algorithm's efficiency and reducing the model's complexity.

CRediT authorship contribution statement

Haiyan Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis. **Weimin Yang:** Validation, Software, Resources, Methodology, Data curation, Conceptualization. **Wenying Feng:** Validation, Data curation. **Liyi Zeng:** Validation, Data curation. **Zhaoquan Gu:** Writing – review & editing, Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Major Key Project of PCL (Grant No. PCL2022A03).

References

[1] Zhang K, Chen X, Jing Y, Wang S, Tang L. Survey of research on named entity recognition in cyber threat intelligence. In: 2022 IEEE 7th international conference on smart cloud. smartCloud, 2022, p. 68–73.
[2] Topçu B, El-Kahlout İD. Turkish named entity recognition for search engine queries. In: 2021 29th signal processing and communications applications conference. SIU, 2021, p. 1–4.
[3] Guven ZA, Unalir MO. Improving the BERT model with proposed named entity recognition method for question answering. In: 2021 6th international conference on computer science and engineering. UBMK, 2021, p. 204–8.
[4] Punjabi S, Arsikere H, Garimella S. Language model bootstrapping using neural machine translation for conversational speech recognition. In: 2019 IEEE automatic speech recognition and understanding workshop. ASRU, 2019, p. 487–93.
[5] Gao C, Zhang X, Hang M, Liu H. A review on cyber security named entity recognition. Front Inf Technol Electron Eng 2021;22(9):1153–68.
[6] Georgescu TM, Iancu B, Zurini M. Named-entity-recognition-based automated system for diagnosing cybersecurity situations in IoT networks. Sensors 2019;19(15):3380.
[7] Gasmi H, Bouras A, Laval J. LSTM recurrent neural networks for cybersecurity named entity recognition. ICSEA 2018;11:2018.
[8] Wang Y, Yao Q, Kwok JT, Ni LM. Generalizing from a few examples: A survey on few-shot learning. ACM Comput Surv (CSUR) 2020;53(3):1–34.
[9] Li J, Chiu B, Feng S, Wang H. Few-shot named entity recognition via meta-learning. IEEE Trans Knowl Data Eng 2020;34(9):4245–56.
[10] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. Adv Neural Inf Process Syst 2017;30.
[11] Yang Y, Katiyar A. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In: Proceedings of the 2020 conference on empirical methods in natural language processing. EMNLP, 2020, p. 6365–75.
[12] Ye Z, Geng Y, Chen J, Chen J, Xu X, Zheng S, Wang F, Zhang J, Chen H. Zero-shot text classification via reinforced self-training. In: Proceedings of the 58th annual meeting of the association for computational linguistics.. 2020.
[13] Zoph B, Ghiasi G, Lin T, Cui Y, Liu H, Cubuk ED, Le Q. Rethinking pre-training and self-training. Adv Neural Inf Process Syst 2020;33:3833–45.
[14] Chen Y, Zhang Y, Zhang C, Lee G, Cheng R, Li H. Revisiting self-training for few-shot learning of language model. In: Proceedings of the 2021 conference on empirical methods in natural language processing. 2021, p. 9125–35.
[15] Devlin J, Chang M, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019, p. 4171–86.
[16] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. Roberta: A robustly optimized bert pretraining approach. 2019, arXiv preprint arXiv:1907.11692.
[17] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. In: ICLR. 2020.
[18] Zhao J, Xie X, Xu X, Sun S. Multi-view learning overview: Recent progress and new challenges. Inf Fusion 2017;38:43–54.
[19] Ding J, Xu W, Wang A, Zhao S, Zhang Q. Joint multi-view character embedding model for named entity recognition of Chinese car reviews. Neural Comput Appl 2023;1–16.
[20] Annamoradnejad I, Habibi J, Fazli M. Multi-view approach to suggest moderation actions in community question answering sites. Inform Sci 2022;600:144–54.
[21] Gonçalves C, Vieira A, Gonçalves C, Camacho R, Iglesias E, Diz L. A novel multi-view ensemble learning architecture to improve the structured text classification. Information 2022;13(6):283.
[22] Raeed A, Gao J, J, Chen, Oloulade B, Lyu T. Multi-view graph neural architecture search for biomedical entity and relation extraction. IEEE/ACM Trans Comput Biol Bioinform 2022.
[23] Sun X, Sun S, Yin M, Yang H. Hybrid neural conditional random fields for multi-view sequence labeling. Knowl-Based Syst 2020;189:105151.
[24] Ning X, Wang X, Xu S, Cai W, Zhang L, Yu L, Li W. A review of research on co-training. Concurr Comput: Pract Exp 2021;e6276.
[25] Xu C, et al. Adversarial incomplete multi-view clustering. In: IJCAI. 2019, p. 3933–9.

[26] Xu C, Zhao W, Zhao J, Guan Z, Song X, Li J. Uncertainty-aware multi-view deep learning for internet of things applications. IEEE Trans Ind Inform 2023;19(2):1456–66.
[27] Liu Y, Xu C, Chen L, Yan M, Zhao W, Guan Z. TABLE: Time-aware balanced multi-view learning for stock ranking. Knowl-Based Syst 2024;112424.
[28] Zhou Z, Li M. Tri-training: exploiting unlabeled data using three classifiers. In: IEEE transactions on knowledge and data engineering. 2005, p. 1529–41.
[29] Li W, et al. Drug specification named entity recognition base on BiLSTM-CRF model. In: 2019 IEEE 43rd annual computer software and applications conference. COMPSAC, Milwaukee, WI, USA; 2019, p. 429–33.
[30] Chang C, et al. Multi-information preprocessing event extraction with BiLSTM-CRF attention for academic knowledge graph construction. IEEE Trans Comput Soc Syst 2023;10(5):2713–24.
[31] He B, Chen J. Named entity recognition method in network security domain based on BERT-BiLSTM-CRF. In: 2021 IEEE 21st international conference on communication technology. ICCT, Tianjin, China; 2021, p. 508–12.