Enhancing Cyber Threat Intelligence with Named Entity Recognition using BERT-CRF

1st Sheng-Shan Chen
Dept. of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, Taiwan
t111599004@ntut.edu.tw

2nd,* Ren-Hung Hwang
College of Artificial Intelligence
National Yang Ming Chiao Tung University
Tainan, Taiwan
rhhwang@nycu.edu.tw

3rd Chin-Yu Sun
Dept. of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, Taiwan
cysun@ntut.edu.tw

4th Ying-Dar Lin
Dept. of Computer Science
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
ydlin@cs.nycu.edu.tw

5th,* Tun-Wen Pai
Dept. of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, Taiwan
twp@ntut.edu.tw

***Abstract*—Cyber Threat Intelligence (CTI) helps organizations understand the tactics, techniques, and procedures used by potential cyber criminals to defend against cyber threats. To protect the core systems and services of organizations, security analysts must analyze information about threats and vulnerabilities. However, analyzing large amounts of data requires significant time and effort. To streamline this process, we propose an enhanced architecture, BERT-CRF, by removing the BiLSTM layer from the conventional BERT-BiLSTM-CRF model. This model leverages the strengths of deep learning-based language models to extract critical threat intelligence and novel information from threats effectively. In our BERT-CRF model, the token embeddings generated by BERT are directly fed into the Conditional Random Field (CRF) layer for efficient Named Entity Recognition (NER), thus preventing the need for an intermediate BiLSTM layer. We train and evaluate the model with three publicly available threat entity databases. We also collect open-source threat intelligence data from recent years for evaluating the applicability of the constructed model in a real-world environment. Furthermore, we compare our model with the most popular GPT-3.5 and the most downloaded open-source BERT question-and-answer models. Through this study, our proposed model demonstrated robust usability and outperformed other models, signifying its potential for application in CTI. In a real-world scenario, our model achieved an accuracy of 82.64%, while with malware-specific threat intelligence data, it achieved an impressive accuracy of 93.95%. The code for this research is publicly available at https://github.com/stwater20/ner_bert_crf_open_version.**

***Index Terms*—cyber threat intelligence, deep learning, cyber security**

I. Introduction

Cyber Threat Intelligence (CTI) is crucial in modern cybersecurity. It bolsters organizational defenses by detecting malicious activities and predicting future attacks. One primary source of CTI is Open-source intelligence (OSINT), which captures security events through indicators of compromise (IoC) from a multitude of sources such as network files, public databases, and social media platforms [6], [19]. However, CTI is expanding at an incredible rate, and extracting critical information from numerous CTI reports may increase the time required, which could cause an organization to miss opportunities to analyze attacker intentions and defense options. Therefore, extracting threat-named entities from CTI reports using automatic methods is valuable to security analysts and an essential step in cyber-security research.

This research problem is Given inputs of various forms of CTI data; the challenge is to decide on the optimal set of techniques to extract threat-named entities with high accuracy and computational efficiency to minimize the discrepancy between the extracted entities and the actual threats. This decision-making process is subject to the constraints of dynamic cyber threat landscapes, the growing size of CTI datasets, and the limitations of existing extraction models. This underscores the need to explore new approaches.

Machine learning and deep learning techniques have markedly improved the extraction of vital entities, with studies confirming their efficacy [15], [20]. In particular, Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM) deep-learning models have achieved significance in NLP in various domains [5], [9]. Although BiLSTM is effective with relatively more minor datasets, recent trends in NLP research highlight Transformers, often exceeding BiLSTM performance [3]. BERT (Bidirectional Encoder

Representations from Transformers) is a large language model that distinguishes itself in this field. This model's strength stems from its extensive pretraining on vast text datasets, allowing it to be subsequently fine-tuned for specialized tasks. Its adaptability ensures consistent top-tier performance across a range of NLP challenges. Building upon the success of models like BERT, the Conditional Random Field (CRF) approach, a form of statistical modeling, has seen extensive adoption in Named Entity Recognition (NER) tasks. CRF's unique capability is its potential to encompass entire word sequences in texts, thus enhancing contextual interpretation and subsequently accuracy. Consequently, integrating BERT, BiLSTM, and CRF resulted in the BERT-BiLSTM-CRF model, which has become a contemporary benchmark in NER efforts [16], [21].

We aim to simplify the model architecture, reduce computational requirements, and potentially improve performance. This led to the formation of a new BERT-CRF model. We tested and evaluated this simplified BERT-CRF model on real-world cyber threat intelligence data and compared its effectiveness in NER with the popular BERT-BiLSTM-CRF model, GPT3.5 [1], and DistilBERT [13].

In the course of our research, we established a website specifically designed for manual annotation of data, which serves as a mechanism to evaluate the precision of each model under study. This critical validation task is entrusted to information security professionals. The decision to employ manual evaluation over automated techniques was driven by the nuanced requirements of our study, with the understanding that this method provides a more accurate and reliable assessment of the model's performance. In the conducted experiment, our model demonstrated an accuracy of 82.64% in real-world environments, which we define as complex, dynamic scenarios that encompass diverse and unpredictable cybersecurity threats. The model also achieved an accuracy of 93.95% when evaluated using limited malicious program threat reports, specifically those related to 'Emotet.'

The main contributions of this paper are summarized as follows:

- Our proposed deep learning model, based on BERT-CRF, achieved an impressive F1 score of 90.02%.
- We demonstrated the robustness of the proposed model by training and evaluating it in three different threat intelligence datasets.
- When evaluated on real-world threat intelligence, our proposed model outperformed even the most popular models currently in use, including chatGPT and DistilBERT.

II. Related Work

Detecting and extracting threat information from Web texts about potential attacks and vulnerabilities is a crucial area of research. The architecture selected for modeling can greatly influence the effectiveness of such efforts. Historical endeavors in this space employed diverse models. Mulwad *et al.* [10] leveraged the Support Vector Machine (SVM) for this purpose, using supervised learning classification and sequence labeling to design rules learned from training samples. In parallel, Chen *et al.* [2] and Lafferty *et al.* [7] respectively probed the maximum entropy model (ME) and CRF. Despite the considerable contributions of these machine learning methods, they frequently required labor-intensive manual annotations.

Recent advances have brought about influential models such as the one from Gasmi *et al.* [4], which employed LSTM-CRF for cybersecurity entity recognition. The addition of a backward LSTM to the original LSTM (known as BiLSTM) was introduced by Schuster *et al.* [14]. This structure captures contextual nuances in sequential data effectively. Kim *et al.* [20] furthered this by incorporating BiLSTM with CRF, enhancing the transition relationships between output sequences and resulting in an impressive F1 score of 75.05

Modern trends in NER are now heavily influenced by transformer methods, with BERT leading the way. Its strength lies in pretraining on large-scale unlabeled datasets, facilitating rich language model training. This pretrained model can then undergo transfer learning, showcasing commendable efficacy even on smaller datasets. For example, Xie *et al.* [18] applied the BERT-BiLSTM-CRF architecture for Chinese entity recognition. Furthermore, Sanh *et al.* [13] introduced DistilBERT, a streamlined variant of BERT, which offers enhanced efficiency with fewer parameters and faster operational speed.

Historically, architectures such as BiLSTM have been revered for recognizing sequential dependencies within texts. However, BERT has reshaped the landscape with its unrivaled ability to discern intricate contextual details from vast data sources, arguably outperforming its predecessors. We combine BERT's contextual prowess with CRF's sequence prediction capabilities. Connecting BERT directly with CRF could bypass intermediary architectures such as BiLSTM. This led to a more efficient model with fewer potential error points. Our BERT-CRF model demonstrated outstanding performance in our experimental evaluations as a testament to its efficacy.

III. Dataset

To validate the performance of BERT-CRF in threat-named entity recognition tasks, we used three publicly available datasets to train and evaluate the proposed model. Data sets were divided into 70% for training, 10% for validation, and 20% for testing. The data sets were annotated using the BIO tagging scheme, where the ”B” prefix indicates the beginning of a label, the ”I” prefix indicates the inside of a label, and the ”O” prefix indicates a token that is outside of the predefined classes. The following subsections describe the details of each data set.

A. DNRTI

DNRTI [17] is a large-scale dataset for threat intelligence named entity recognition, which analyzed more than 300 threat reports with 182,452 words. The DNRTI dataset contains 13 classes: hacker groups, attacks, sample files, security teams, tools, time, purpose, region, industry, organization, method, vulnerability, and feature.

B. CTI-Reports

CTI-Reports [11] are a collection of threat reports released by nlpai-lab on GitHub, comprising 310,406 records. The category is divided into four major items: malware, IP, URL, and hash. The malware category is further divided into ten subcategories: malware.backdoor, malware.infosteal, malware.ransom, malware.unknown, malware.drop, url.normal, url.unknown, url.cncsvr , ip.unknown, and hash.

C. MalwareTextDB

Compared to DNRTI, MalwareTextDB [8] is an older and more established threat intelligence dataset that includes three types of entities: Action, Modifier, and Entity. It comprises a wide range of malicious code files, spam emails, websites, and other malware-related text samples. These text samples can assist researchers and security professionals research and develop solutions for malware detection, analysis, and other related cybersecurity issues.

D. Real-word CTI

We utilized openCTI, an open-source threat intelligence platform that connects various sources of threat intelligence such as AlienVault, VirusTotal, and Treatpost, and collected 9,872 pieces of OSINT from it as a means to verify the performance of the model in real-world scenarios.

IV. Methodology

This study applied a deep learning model based on PyTorch, consisting of BERT and CRF modules. The framework of the proposed model is shown in Figure 1. The threat intelligence was imported into the BERT layer for pretraining, and then the vector sequence of output was entered into the CRF layer to calculate the optimal labeled sequence.

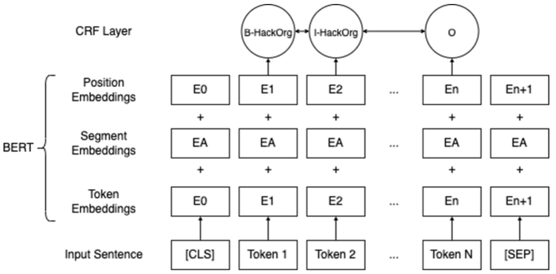


Fig. 1: The proposed BERT-CRF model structure.

A. BERT

BERT is a pretrained model that applies self-supervised learning on a large amount of English data to obtain word embeddings with higher dimensionality to improve word disambiguation. BERT utilized two methods for pretraining: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM randomly masked 15% of the input words in a sentence during training and then predicted the masked words. NSP randomly paired two input sentences and predicted whether they were consecutive. Therefore, BERT's pretrained model can fully utilize information on both sides of a word, resulting in a better representation of word distribution. In this study, BERT-base-uncased was applied, representing the use of the English language, a network structure of 12 transformer blocks, 768 hidden, and 12 attention heads, with a parameter size of 110M. It is worth mentioning that secBERT [12], proposed by Jackaduma on GitHub, is a model specifically trained on threat intelligence datasets. To our knowledge, it is the first preprocessing model that uses cybersecurity threat intelligence as a BERT training set. Therefore, we applied this model as an experimental object.

B. CRF

NER is a sequence labeling task in which our dataset follows the BIO scheme, a classification problem for BERT. BERT cannot handle the task classification directly, so a linear classifier is usually added behind BERT. However, a linear classifier can only consider the information about the features of the current word and cannot capture the global dependency relationship of the sequence. Therefore, we added a CRF layer behind the identification model and replaced the original linear classifier. We defined the target function of the downstream tasks to train the classifier from scratch and fine-tune the parameters of BERT. CRF was able to model the joint probability of the input sequence and its corresponding labeled sequence and maximized the probability of the output sequence by learning probability parameters. Specifically, given a BERT output sequence H, which would be a set of vectors and the corresponding labeled sequence Y. The goal of CRF is to find the best label sequence Y. Therefore, the joint probability of the CRF can be represented as Equation (1).

$$p(Y \mid H)=\frac{1}{Z(H)} \exp \left(\sum_{i=1}^{n} \sum_{j=1}^{k} \lambda_j f_j\left(y_i, y_{i-1}, H, i\right)\right) \tag{1}$$

Regarding the CRF equation, $Z(H)$ is a normalization constant that ensures the sum of probabilities as 1. The function $f_j\left(y_i, y_{i-1}, H, i\right)$ is a feature function that captures different patterns and features in the sequence, while λ_j is the corresponding feature weights that measure the influence of the feature on the labeled sequence. The y_i represents the $i-th$ label in the input sequence, y_{i-1} for the $(i-1)-th$ label sequence, H for the entire vector of features of the input sequence, and i for the position of the current label in the sequence. We obtain the final label sequence by calculating $p(Y|H)$ for each one and selecting the one with the maximum probability.

C. Experimental environment

The hardware environment for this study is 11th Gen Intel(R) Core (TM) i7-11700 @ 2.50GHz, 16GB memory space, Ubuntu 20.04 x64 operating system, and NVIDIA RTX A5000 graphics card. The dependent environment is built on Python3 + PyTorch. To verify the performance of BERT-CRF in the recognition of named entities from threat intelligence, we compared the CRF, secBERT-CRF, BERT-BiLSTM-CRF, secBERT-BiLSTM-CRF, and BERT-CRF models in this study. The effectiveness of our proposed method was evaluated based on the F1 score, the most widely used quantitative evaluation method in NER tasks, and it has become a standard metric to overcome the shortcomings of using only accuracy as a performance metric. The F1 score comprises the harmonic mean of precision and recall.

In our experiment, we used a pretrained BERT model for fine-tuning. The model consisted of 16 layers with 768 parameters in each hidden layer. Two learning rates were established: for fine-tuning parameters, the learning rate was configured to 5e-5, while for the CRF and fully connected layers, it was set at 8e-5. Weight decay was employed as a measure to prevent overfitting. Specifically, a weight decay of 1e-5 was set for the fine-tuning parameters, while a weight decay of 5e-6 was applied to the CRF and fully connected layers. However, it should be noted that no weight decay was applied to the bias terms and the parameters of the layer normalization layers in our model. The Adam optimizer was used to facilitate the model's training process. In each iteration, the batch size, i.e., the number of samples input into the model, was 16. The total number of training epochs was set to 50. Regarding the input text sequences to the BERT model, we limited the maximum length to 256 tokens. This is because the BERT model needs to load the entire input sequence into memory, and too long a sequence may lead to insufficient memory. During the training process, we accumulated the gradients at each iteration. This implies that, in practice, we update the model parameters every 16 samples (batch size). These were the primary parameter settings for training our BERT model. The model was trained on the basis of these configurations, and its performance was subsequently evaluated.

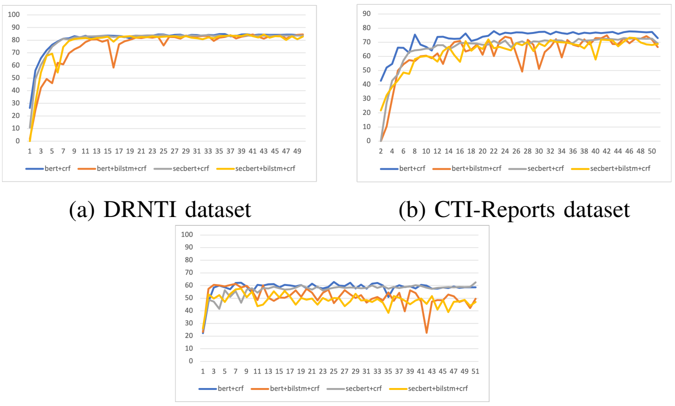
V. Results and Discussion

The results of the tests of the BERT-CRF, secBERT-CRF, BERT-BiLSTM-CRF and secBERT-BiLSTM-CRF models are shown in Table 1.

TABLE I: The comparison of the models

Dataset	Model	Accuracy	Recall	F1-Score
[17]	CRF	84.00%	78.00%	80.00%
[17]	BERT-CRF	96.36%	88.59%	90.02%
[17]	secBERT-CRF	96.00%	88.80%	88.62%
[17]	BERT-BiLSTM-CRF	94.84%	85.03%	84.59%
[17]	secBERT-BiLSTM-CRF	94.52%	84.48%	83.77%
[11]	BERT-CRF	98.37%	74.10%	77.29%
[11]	secBERT-CRF	97.42%	66.69%	72.52%
[11]	BERT-BiLSTM-CRF	97.44%	66.27%	74.39%
[11]	secBERT-BiLSTM-CRF	97.31%	80.31%	68.05%
[8]	BERT-CRF	87.76%	47.39%	58.57%
[8]	secBERT-CRF	87.68%	57.16%	62.53%
[8]	BERT-BiLSTM-CRF	85.59%	38.92%	45.59%
[8]	secBERT-BiLSTM-CRF	85.59%	69.72%	47.07%

Table 1 demonstrates that deep learning-based CRF models achieve higher accuracy compared to traditional machine learning-based CRF models. BERT pretrained models have the ability to capture global information, while BiLSTM selectively integrates valuable information, resulting in insignificant gains. As a result, their performances on all three datasets are inferior to BERT-CRF. In particular, when the input consists of threat intelligence content in the DNRTI dataset, the secBERT-CRF F1 score is approximately 1.4% lower than that of BERT-CRF. The convergence results after training each data set for 50 epochs are illustrated in Figure 2.



(a) DRNTI dataset

(b) CTI-Reports dataset

(c) MalwareTextDB dataset

Fig. 2: Comparison of F1-Score on different datasets

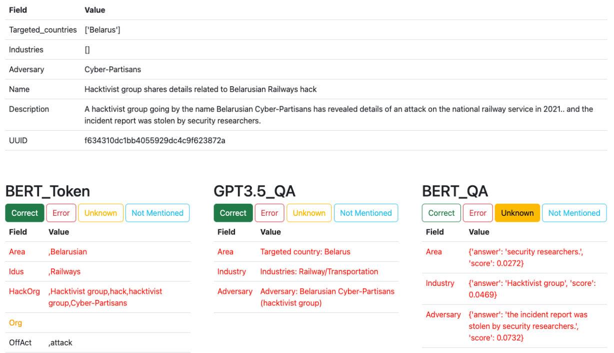
A. Case Study - OSINT

To assess the applicability of the models in real-world scenarios, we conducted a study using OSINT and compared the performance of DistilBERT and GPT3.5 models. To facilitate this study, we developed an annotation website, as depicted in Figure 5, where threat titles and descriptions were provided as input. These inputs were then processed using three different models: BERT-CRF, GPT3.5, and DistilBERT. The evaluation focused on three crucial elements for cybersecurity researchers: Area, Industry and HackOrg (Adversary).

The evaluation methodology involved categorizing items as mentioned or not mentioned in the text. The annotation process followed the guidelines outlined below:

- When the item is mentioned in the text:
  - ”Correct” indicates that the model accurately identified the item.
  - ”Error” indicates that the model incorrectly identified the item, including cases where it identified the item as empty.
- When the item is not mentioned in the text:
  - ”Correct” indicates that the model accurately identified the item, even if the answer was not explicitly mentioned in the text.
  - ”Error” indicates that the model incorrectly identified the item.
  - ”Unknown” indicates that the model's output was empty, labeled ”Not mentioned,” or had a DistilBERT probability below 30

This evaluation methodology allowed us to assess the model performance in correctly identifying relevant elements, even in cases where the information was not explicitly mentioned.



Field	Value
Targeted_countries	['Belarus']
Industries	[]
Adversary	Cyber-Partisans
Name	Hacktivist group shares details related to Belarusian Railways hack
Description	A hacktivist group going by the name Belarusian Cyber-Partisans has revealed details of an attack on the national railway service in 2021.. and the incident report was stolen by security researchers.
UUID	f634310dc1bb4055929dc4c9f623872a

BERT_Token

Correct Error Unknown Not Mentioned

Field	Value
Area	,Belarusian
Idus	,Railways
HackOrg	,Hacktivist group,hack,hacktivist group,Cyber-Partisans
Org	
OffAct	,attack

GPT3.5_QA

Correct Error Unknown Not Mentioned

Field	Value
Area	Targeted country: Belarus
Industry	Industries: Railway/Transportation
Adversary	Adversary: Belarusian Cyber-Partisans (hacktivist group)

BERT_QA

Correct Error Unknown Not Mentioned

Field	Value
Area	{'answer': 'security researchers.', 'score': 0.0272}
Industry	{'answer': 'Hacktivist group', 'score': 0.0469}
Adversary	{'answer': 'the incident report was stolen by security researchers.', 'score': 0.0732}

Fig. 3: Using a self-developed labeling website, cybersecurity professionals were invited to assess the accuracy of the model, and the fields of Area, Industry, and Adversary were compared.

We manually annotated a total of 197 instances of threat intelligence data, and approximately 42.03% of these instances contained corresponding tags mentioned in the text. Table 2 presents the accuracy results for the correctly identified labels, revealing that the average accuracy of BERT-CRF is only 77.71%. Consequently, we decided to further analyze the input data by segmenting it based on its length.

To perform this analysis, we divided the threat intelligence data into different segments, taking into account the length of each instance. By doing so, we aimed to investigate the relationship between input length and model performance. This segmentation approach allowed us to gain deeper insights into the challenges faced by BERT-CRF in accurately labeling instances of varying lengths.

TABLE II: Comparison table of model performance using OSINT dataset

Label	Model	Correct rate
Area	BERT-CRF	63.38%
	GPT3.5	41.43%
	DistilBERT	64.29%
Idus	BERT-CRF	97.02%
	GPT3.5	60.42%
	DistilBERT	35.42%
HackOrg	BERT-CRF	72.73%
	GPT3.5	89.09%
	DistilBERT	64.81%

For comparative analysis, we performed a dataset segmentation based on input length. Specifically, we divided the data set into two groups: inputs with a length greater than 50 characters and inputs with a length less than or equal to 50 characters. The evaluation results for these two groups are presented in Table 3.

The results indicate that selecting threat intelligence with more than 50 characters yields significantly higher accuracy when using the BERT-CRF model. The average precision achieved is 94.93%, which is 35.56% higher than that of threat intelligence with 50 characters or less. Furthermore, the BERT-CRF model outperforms both the GPT3.5 model, which achieves an accuracy of 70.71%, and the DistilBERT model, which achieves an accuracy of 49.60%.

B. Case Study - Emotet

Emotet [12] is one of the most destructive malware affecting governments, businesses, organizations, and individual computer users. It is a highly evolved malware that spreads through spam emails and malicious attachments, enabling remote control of systems and data exfiltration. Our research focused on gathering relevant threat intelligence about Emotet, and we obtained 79 pieces of valid information from the OpenCTI platform. The collected threat intelligence provides valuable insights into Emotet's attack patterns, evolutionary trends, and defense strategies. The statistical results, depicted in Figures 6 and 7, reveal interesting findings. Although most of the threat intelligence collected does not appear explicitly in the text, the BERT-CRF model demonstrates a higher accuracy rate. Additionally, the GPT3.5 model occasionally provides correct answers even when the corresponding information is not explicitly mentioned in the text. These results highlight the potential of these models to understand and process implicit information related to Emotet.

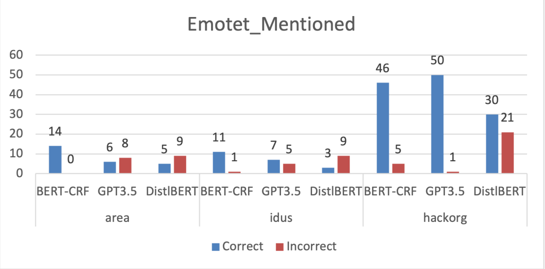


Fig. 4: The text mentions the use of CTI regarding Emotet.

C. Conclusion

In this study, we apply an established methodology to a unique context. More specifically, we utilize the BERT-CRF model in a named entity recognition task tailored to the network security domain. Despite the fact that the usage of BERT-CRF for cybersecurity NER is not novel, our study contributes to the existing body of knowledge by demonstrating how

TABLE III: Accuracy after cutting input by 50 characters

Label	Model	Length >50	Length $\leq$ 50
Area	BERT-CRF	96.55%	40.48%
	GPT3.5	60.71%	28.57%
	DistilBERT	48.28%	75.61%
Idus	BERT-CRF	100.00%	90.00%
	GPT3.5	63.18%	50.00%
	DistilBERT	44.74%	0.00%
HackOrg	BERT-CRF	88.24%	47.62%
	GPT3.5	88.24%	90.48%
	DistilBERT	61.77%	70.00%

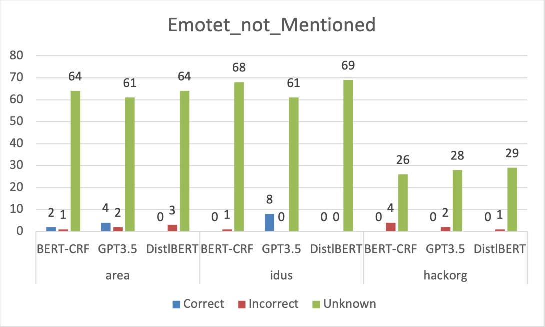


Fig. 5: The text does not mention the use of CTI regarding Emotet.

the model can be effectively implemented in this particular setting. The BERT model adjusted the context of all layers jointly. In contrast, the CRF layer restricted the dependencies of the labels and preserved contextual information to ensure the final output satisfied the BIO label rules. Our experiments demonstrated that BERT-CRF outperformed other models in threat-named entity recognition and effectively solved entity recognition problems in cybersecurity.

D. Future Work

In our future work, we will focus on integrating the NER results to build a comprehensive threat knowledge graph. By leveraging the entities extracted from the NER task, we can establish relationships and connections between various sources of threat intelligence, enabling the integration of cross-threat intelligence. This integration of NER with the construction of threat knowledge graphs aims to capture and represent complex relationships among entities such as malicious software, vulnerabilities, attack vectors, and threat indicators.

The construction of a threat knowledge graph will improve our understanding of cybersecurity risks and facilitate more effective threat analysis and mitigation strategies. The graph will serve as a structured representation of threat-related information, allowing us to uncover hidden patterns and correlations, identify emerging threats, and prioritize response measures based on their potential impact. This integration of NER with knowledge graph construction will provide a unified perspective on the cybersecurity risk landscape, empowering analysts and security professionals to make informed decisions and proactively address evolving security threats.

In summary, our future work aims to bridge NER and threat intelligence integration by effectively leveraging both techniques to construct a comprehensive threat knowledge graph.

Acknowledgment

This work was supported in part by the Cybersecurity Center of Excellence (CCoE) program under National Science and Technology Council (NSTC), Taiwan.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Stanley F Chen and Ronald Rosenfeld. A gaussian prior for smoothing maximum entropy models. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1999.

[3] Aysu Ezen-Can. A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*, 2020.

[4] Houssem Gasmi, Abdelaziz Bouras, and Jannik Laval. Lstm recurrent neural networks for cybersecurity named entity recognition. *ICSEA*, 11:2018, 2018.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] Yong-Woon Hwang, Im-Yeong Lee, Hwankuk Kim, Hyejung Lee, and Donghyun Kim. Current status and security trend of osint. *Wireless Communications and Mobile Computing*, 2022, 2022.

[7] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[8] Swee Kiat Lim, Aldrian Obaja Muis, Wei Lu, and Chen Hui Ong. Malwaretextdb: A database for annotated malware articles. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1567, 2017.

[9] Nut Limsopatham and Nigel Collier. Bidirectional LSTM for named entity recognition in Twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[10] Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin, and Krishnamurthy Viswanathan. Extracting information about security vulnerabilities from web text. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 257–260. IEEE, 2011.

[11] Nlpai-Lab. Nlpai-lab/cti-reports-dataset, 2020.

[12] Constantinos Patsakis and Anargyros Chrysanthou. Analysing the fall 2020 emotet campaign. *arXiv preprint arXiv:2011.06479*, 2020.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[14] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[15] WANG Tong, AI Zhong-Liang, and ZHANG Xian-guo. Knowledge graph construction of threat intelligence based on deep learning. *Computer and Modernization*, (12):21, 2019.

[16] Xuren Wang, Songheng He, Zihan Xiong, Xinxin Wei, Zhengwei Jiang, Sihan Chen, and Jun Jiang. Aptner: A specific dataset for ner missions in cyber threat intelligence field. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1233–1238, 2022.

[17] Xuren Wang, Xinpei Liu, Shengqin Ao, Ning Li, Zhengwei Jiang, Zongyi Xu, Zihan Xiong, Mengbo Xiong, and Xiaoqing Zhang. Dnrti: A large-scale dataset for named entity recognition in threat intelligence. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1842–1848. IEEE, 2020.

[18] Teng Xie, J Yang, and Hui Liu. Chinese entity recognition based on bert-bilstm-crf model. *Computer Systems & Applications*, 29(7):48–55, 2020.

[19] Ashok Yadav, Atul Kumar, and Vrijendra Singh. Open-source intelligence: a comprehensive review of the current state, applications and future perspectives in cyber security. *Artificial Intelligence Review*, pages 1–32, 2023.

[20] Keke Zhang, Xu Chen, Yongjun Jing, Shuyang Wang, and Lijun Tang. Survey of research on named entity recognition in cyber threat intelligence. In *2022 IEEE 7th International Conference on Smart Cloud (SmartCloud)*, pages 68–73. IEEE, 2022.

[21] Shieheng Zhou, Jingju Liu, Xiaofeng Zhong, and Wendian Zhao. Named entity recognition using bert with whole world masking in cybersecurity domain. In *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, pages 316–320, 2021.