Hindawi
Security and Communication Networks
Volume 2022, Article ID 9875199, 15 pages
https://doi.org/10.1155/2022/9875199

WILEY | Hindawi

Research Article

CTI View: APT Threat Intelligence Analysis System

Yinghai Zhou, Yi Tang, Ming Yi, Chuanyu Xi, and Hai Lu

China Academy of Engineer Physics, Institute of Computer Application, Mianyang 621054, China

Correspondence should be addressed to Chuanyu Xi; xicycaep@163.com

Received 23 September 2021; Revised 21 October 2021; Accepted 1 November 2021; Published 3 January 2022

Academic Editor: Gu Zhaoquan

With the development of advanced persistent threat (APT) and the increasingly severe situation of network security, the strategic defense idea with the concept of "active defense, traceability, and countermeasures" arises at the historic moment, thus cyberspace threat intelligence (CTI) has become increasingly valuable in enhancing the ability to resist cyber threats. Based on the actual demand of defending against the APT threat, we apply natural language processing to process the cyberspace threat intelligence (CTI) and design a new automation system CTI View, which is oriented to text extraction and analysis for the massive unstructured cyberspace threat intelligence (CTI) released by various security vendors. The main work of CTI View is as follows: (1) to deal with heterogeneous CTI, a text extraction framework for threat intelligence is designed based on automated test framework, text recognition technology, and text denoising technology. It effectively solves the problem of poor adaptability when crawlers are used to crawl heterogeneous CTI; (2) using regular expressions combined with blacklist and whitelist mechanism to extract the IOC and TTP information described in CTI effectively; (3) according to the actual requirements, a model based on bidirectional encoder representations from transformers (BERT) is designed to complete the entity extraction algorithm for heterogeneous threat intelligence. In this paper, the GRU layer is added to the existing BERT-BiLSTM-CRF model, and we evaluate the proposed model on the marked dataset and get better performance than the current mainstream entity extraction mode.

1. Introduction

High concealed unknown threat was first proposed by the US Department of Defense and the US Air Force. The essence of it is targeted attack, which uses more advanced and stealthiest attack means to carry out long-term and continuous network attack on specific targets [1]. Advanced persistent threat (APT) is a stealthy threat actor, typically a nation state or state-sponsored group, which gains unauthorized access to a computer network and remains undetected for an extended period [2].

Since the exposure of the Operation Aurora attacks against Google in December 2009, high-covert and unknown threats in cyberspace represented by APT are increasingly rampant and show the trend of game and contest among countries. Figure 1 shows the major APT attacks in recent years, including

2009: Google Aurora attacks [3].

2010: Stuxnet attack at Bushehr Nuclear Power Plant in Iran [4].

2011: Duqu, son of Stuxnet [5].

2012: LuckyCat campaign with multiple targets in India and Japan [6].

2013: Dark Seoul Cyberattack [7].

2014: An attack launched by the APT group against an unnamed steel plant in Germany resulted in significant damage [8].

2015: December 2015 Ukraine power grid cyberattack [9].

2016: Bangladesh Bank cyber heist [10].

2017: Russia hacked the US electric grid [11].

2018: Russian interference in the 2018 United States elections [12].

2019: U.S. escalates online attacks on Russia's power grid [13].

2020: Portuguese energy giant hit by ransomware attack [14].

2021: Colonial pipeline cyberattack [15].

2021: Computer giant Acer hit by $50 million ransomware attack [16].

2021: Database leak exposes CPF of almost the entire population of Brazil [17].

2021: DDoS attack took down the websites of more than 200 Belgium organizations [18].

2021: 533 million Facebook users' phone numbers and personal data have been leaked online [19].

Due to the characteristics of small flow and strong concealment of these threats, the misinformation and under-reporting during our daily alarm are very common, which makes the security personnel tired of dealing with it. Traditional security measures, such as firewalls, intrusion prevention systems (IPS), and intrusion detection systems (IDS), are difficult to prevent attacks by high-covert and unknown threats in cyberspace. Moreover, fake data may be injected by malicious workers who camouflage as normal workers to interfere with the accuracy of data analysis [20]. In order to protect systems from such attacks, security experts have proposed cyber threat intelligence (CTI) and indicator of compromise (IOC) to issue early warnings when systems encounter suspicious threats [21].

Cyber threat intelligence (CTI) is essential information recorded by security researchers about a past or present cyber threat or security incident. In 2014, Gartner defined CTI in ≪market guide for security threat intelligence services≫ as follows: threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications, and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard [22]. In short, threat intelligence is the information collection of potential or direct harm to enterprises and institutions. It can help companies assess current developments and trends in cyberspace and reckon the threats they are going to face, which can be used to make decisions in the near future [23]. Indicator of compromise (IOC) is a kind of sample information and evidence with high confidence obtained by security personnel when they face network threats, and they are used to describe the malicious activities of the attacker in the APT field. In general, IOCs encountered during APT threat intelligence text analysis usually include hashes (the hash value of the sample file), IP (an IP address), domain (domain is similar to IP, but it needs to be registered for a fee), Mac (host characteristics indicates the characteristics of an attacker's host in a malicious sample), and e-mail (e-mail address). Using IOCs to analyze APT threat intelligence text allows us to build an overall view to track enemy or attack activity by the relative metrics and gain a deeper understanding of what is happening in the cyber environment. At present, people in the field of cyberspace security have formed a broad consensus that threat intelligence drives network security defense, timely intelligence sharing, and accurate intelligence analysis are the keys to efficiently respond to cyber threats. More intuitive, threat information sharing and utilization are a kind of space-for-time senior security protection strategy. By actively perceiving the existing or potential network threats, the time and scope of attack can be limited, the response time of threat can be greatly shortened, and the asymmetric situation in the process of attack and defense can be changed [23].

However, as more and more enterprises pay attention to security issues, the standard or nonstandard CTI data in the world is growing rapidly. While people using these network CTI data to actively defend against high-covert and unknown threats, three major problems emerge due to the flood of these data in the area of threat intelligence: (1) the sources of threat intelligence data are wide which wears analysts out. (2) Types of CTI are complicated and disorganized, and also, their application scenarios are complex. (3) In the Internet era, information generation is fast and threat intelligence is updated quickly. Although STIX (structured threat information expression) [24] and CAPEC [25] (Common attack pattern enumeration and classification) and other frameworks greatly facilitate the sharing of CTI by security researchers to a large extent and solve the problem of miscellany, however, when analyzing the CTI and extracting the required threat descriptions, a lot of manual checks are still needed; the speed of threat intelligence analysis by analysts is far from the speed of threat intelligence generation. Therefore, in recent years, most security researchers and sectors of society have focused on automating the extraction of IOCs from public sources describing attack events to analyze CTI. Among them, Liao et al. [26] developed a system named IACE, which modeled the task of extracting IOCs as a graphical similarity problem. If the IOCs item has a graphical structure similar to that of the training set, it will be identified as a certain IOC. This enables IACE to automatically extract IOCs from cyber threat intelligence (CTI) and capture their context; Long et al. [27] used deep learning technology to apply end-to-end neural network model to automatic identification of IOCs; they automatically identified IOCS items from network threat intelligence (CTI) and achieved good results; Zhao et al. [28] developed a system, named TIMINER, which combined regular expression, named entity recognition technology and domain-specific syntactic dependency to extract IOCs entries from network security texts. Its extraction method showed better performance in terms of accuracy and coverage compared with IACE.

However, in the scene related to the APT, the defender is always in a relatively passive situation compared with the attacker, and the fundamental reason for this situation is the information asymmetry between the attacker and the defender. The attacker can easily obtain the identity information of the defender, and with the development of communication technology, the information of the defender

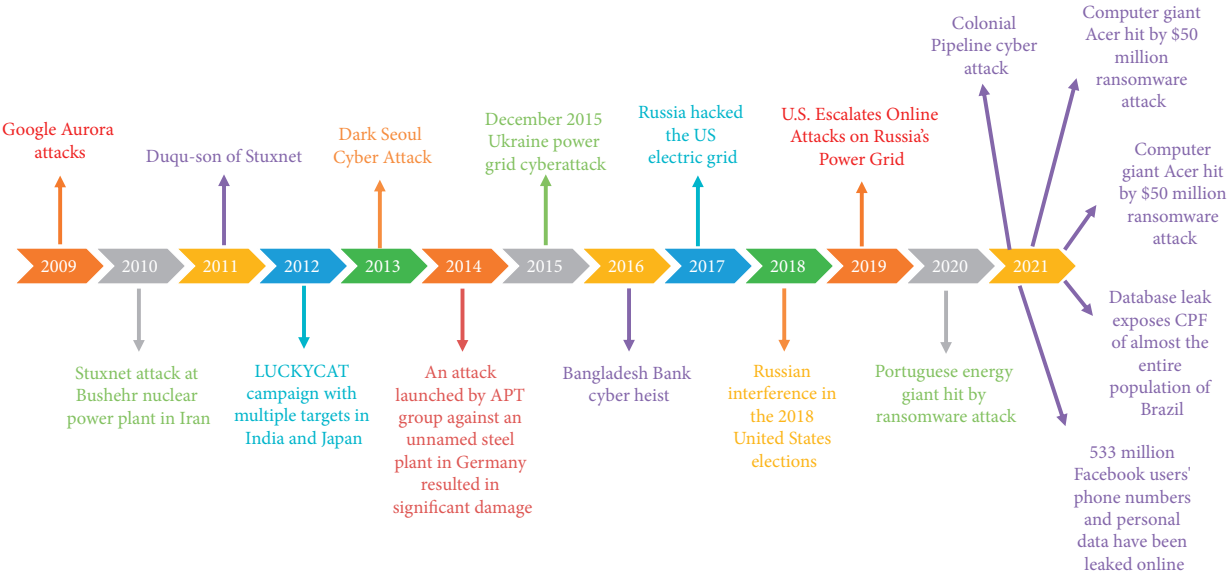

Figure 1: Major APT attacks from 2009 to 2021.

is even more and more easily exposed to the attacker's view. Moreover, in 5G-related communication, the actual identity of the end user is difficult to be anonymous or alias [29]. As long as the attacker changes the IP address or domain name casually, new IOCs can be generated at a low cost, even if the defender owns all the IOCs generated by the previous attack of a hacker or attacking organization, he cannot avoid all risks through these IOCs, let alone making attribution through IOCs. Although IOCs can help us develop better security policies, and the amount of information of IOCs can also affect the final protection effect, IOCs still have the following disadvantages:

(1) IOC cannot represent how the attacker interacts with the victim system.

(2) IOC can only focus on the partial analysis of the attack process and cannot build a complete attack chain, let alone unearth the organizer and executor behind the attack, which leads to inaccurate detection of the attack behavior.

(3) IOC items have low correlation and high independence, which cannot provide strong support for traceability.

In order to solve the problem of multisource heterogeneous CTI and difficulty in analysis, based on the actual requirements, combined with the entity, syntax, and context information related to threat described by CTI, oriented to CTI text, using a variety of natural language processing methods, we designed an analysis system for entity extraction of CTI: CTI View. CTI View's main contributions are summarized as follows:

(1) A CTI text extractor based on web page text recognition technology is designed. The text of shared APT reports from different sources can be easily obtained, which solves the problem of poor adaptability to nonstandard HTML when using crawlers, especially for some websites with anticrawler strategy, CTI View can easily bypass its anticrawler strategy to obtain the web page text.

(2) A threat entity identifier based on bidirectional encoder representations from transformers (BERT) is designed, which can effectively extract the threat entities in CTI text. More specifically, we collect and analyze 120 security texts describing APT threat events, using CTI View to extract the attackers, exploits, regions, industries, and campaigns described in the texts. Experimental results show that the accuracy of this method is more than 72%.

(3) In summary, in order to improve the defense capability of network security, CTI View focuses on the utilization of CTI and comprehensively analyzes the text of CTI by using different methods, which can provide strong data support for security researchers in different stages of threat analysis.

The rest of the paper is arranged as follows: in Section 2, we review the related work; in Section 3, the overall framework of CTI View is summarized, and then each module in the framework is introduced; the focus is on Section 3.4, a BERT-based threat entity identifier is introduced. In Section 4, we carry out experimental verification of our proposed method. Finally, in Section 5, we summarize our work.

2. Related Works

Generally speaking, besides the IOCs, the cyber threat intelligence related to APT involves many other kinds of details, especially semantic information about the attacker, target region, target industry, etc. This information is vital for security researchers to defend against the APT attacks. However, the extraction and analysis of such threat-related information have traditionally relied on a lot of manual

work, which is a tedious task for security analysts. To this end, automated information extraction plays a vital role in the field of CTI analysis.

In the field of statistical machine learning, Mulwad et al. [30] used support vector machine (SVM) classifier to extract concepts related to vulnerabilities, attacks, and threats, but only two kinds of concepts can be identified and extracted by this system, one is the attack means and another is the attack results. Shafiq et al. proposed a method called CorrAUC, which uses the technology of machine learning and effective feature selection to detect malicious network traffic and extract malicious traffic information [31]. Joshi et al. [32] used a conditional random field (CRF)-based method to recognize and classify entities. Jones et al. [33] used semisupervised machine learning methods combined with active learning mechanisms to extract network security entities and relationships.

In the field of deep learning, convolutional neural network (CNN), long short-term memory (LSTM) network, and bidirectional LSTM network are applied to information extraction achieving good results in many fields. In task 8 of the published SemEval-2018, Manikandan et al. [34] used the CNN-CRF model to identify malware-related entities. Dionísio et al. [35] built a named entity recognition model of BilSTM-CRF to identify named entities from tweets related to network security. Luo et al. [36] proposed a method based on deep reinforcement learning to poison the target data using models such as CNN and LSTM, and it can help malicious attackers hide themselves while using TruthFinder to attack. Sun et al. [37] used CNN method for data training and honeypot recognition.

In addition, with the development of natural language processing technology, some scholars use natural language processing technology to analyze network security text from different perspectives. Husari et al. [38] proposed TTPDrill, which uses natural language processing (NLP) and information retrieval (IR) to extract threat actions from unstructured CTI text. Zhao et al. [28] designed an IOC extractor with domain tags, TIMiner, which can automatically extract IOC and the corresponding domain (region information) tags in network security text.

3. Method Description

CTI View consists of five main components, as shown in Figure 2.

The overall architecture of CTI View consists of 4 parts: (1) APT threat intelligence acquisition, (2) text data processing, (3) IOC and TTP extraction, and (4) threat entity extraction. Through the processing and analysis of APT threat intelligence in these four modules, canonical entity data can be extracted to provide strong data support for security research. The function and implementation of each module will be described in detail below.

3.1. APT Threat Intelligence Acquisition. In this part, text recognition technology and natural language technology are used to collect information from the APT threat report, and important data are collected from APT threat reports shared by different security companies, including blogs, hacker forum posts, security news, and security supplier announcements. Firstly, we used a Python automated testing framework "Pyppeteer" [39] with which we converted HTML web content into PDF files. Then, we use PDFMiner [40] (PDFMiner is a Python PDF parser that can extract information from PDF documents) to identify text from PDF files to obtain APT threat report text. The specific process is shown in Figure 3.

Compared with the traditional crawler method, the text information acquisition method proposed in our paper is more universal; the text of APT reports from different sources can be easily obtained, which solves the problem of poor adaptability to nonstandard HTML when using crawlers, especially for some secure websites with anticrawler strategy; by using this method, we can bypass its anticrawler strategy very well, so as to obtain the web page text. However, the disadvantage is that this method will inevitably increase more text noise when obtaining APT shared threat report text, such as advertising, author information, and comments, which makes text data processing more difficult.

3.2. Text Data Processing. In natural language processing (NLP) tasks, most of the data we get are incomplete, inconsistent, missing, or redundant text data. If we directly use the deep learning algorithm to train on such data, the results are often unsatisfactory. Therefore, in order to improve the learning effect and quality, it is necessary to preprocess the data to remove the text noise before using it.

In this part, we denoised and preprocessed the previously extracted text data of APT intelligence, including the removal irrelevant information, for example, the special characters, advertising messages, navigation bars, comments, text clauses, and text participle. We carried out denoising and preprocessing of APT threat intelligence text data in the following order.

3.2.1. Removal of the Special Characters. In this process, we summarized the common special characters in APT threat intelligence and removed them by means of text matching. Detailed information is shown in Table 1.

3.2.2. Removal of Advertising Messages, Navigation Bars, and Comments. After we use the method mentioned in Section 3.1 to extract the APT threat intelligence text data and remove special characters, there are still some irrelevant information that needs to be removed, the advertising messages, navigation bars, and comments. Because each line of text in the navigation bar length is generally less than 30 characters and the comments' advertising information character length is generally greater than 150 characters, we eliminate the whole number less than 30 and more than 150 characters in the text of the line to achieve the purpose of removing advertising information, navigation bar, and comments.

3.2.3. Sentence Split. For the NLP task, the raw data are usually an article or a large section of text. Before other

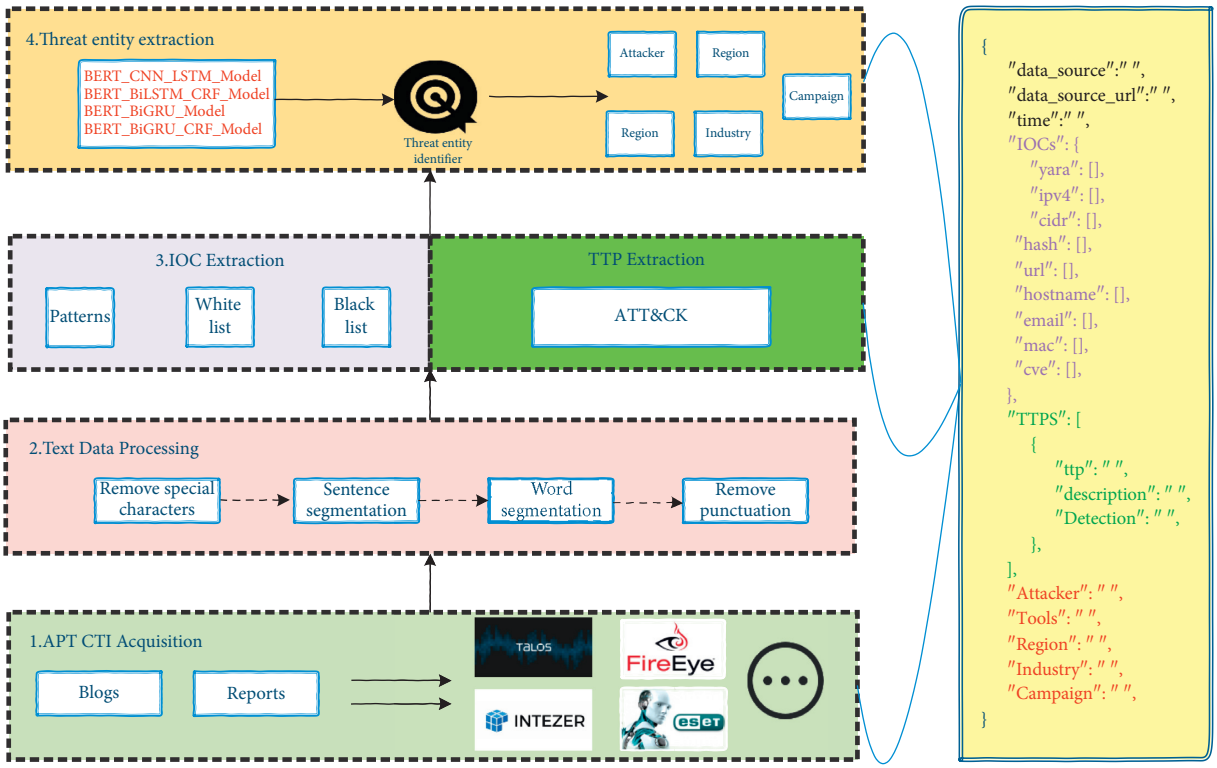

Figure 2: The architecture of our APT threat intelligence analysis system.

Figure 3: Acquisition of APT threat intelligence text.

Table 1: Classification and removal of special character.

Greek letters	Phonetic notations	Symbols for mathematics and unit	Square symbol	Special symbol
αβγδεζ ηθικλμ νξοπρσ τυφχψ ωABΓ ΔEZH ΘIKΛ MNΞO ΠPΣT ΥΦXΨ Ω	ㄅㄉˇㄓ ˊˇㄚㄞㄢ ㄦㄆㄊㄍ ㄐㄔㄕㄧ ㄛㄟㄣㄇ ㄋㄎㄑㄕ ㄤㄈㄌㄏ ㄒㄖㄗㄘ ㄙㄨㄜㄠ ㄤㄈㄌㄏ ㄒㄖㄗㄘ ㄝㄡ āáǎàōóǒò êēéěèīíǐì ūúǔùǖǘǚǜ ńňǹɑg	≈≡≠=≦≧<>≮≯:±+×÷∫∮∝ ∞∧∨∑∏∪∩∈∵∴⊥∥∠⌒⊙ ≌∽√∟∣:≒≦≧⊕◢°′″$ £¥‰%℃¤¢.. .. ≈≌	▗ ▖ ▝ ▂▃▄▅▆ ▇▇▇▇ ▋▋▋▏▒ ▔▕	·ˉˇˋ々 ~‖:'"'| 〃〔〕 〈〉「 」『〖 〗【】 []{}§№ ☆★○● ◎◇◆□■ △▲※ 〓^_¯ ▼▽

processing, the article needs to be cut or processed. Considering our computing resources and the NLP algorithm we adopted, the sentence is a relatively appropriate semantic granularity. Therefore, in this paper, when dealing with APT threat intelligence text, we divide the text into clauses and take sentence level as the smallest unit for subsequent processing (that is, taking sentence as the unit for subsequent training sample).

After the removal of special characters, our APT threat intelligence text is a large section of the irregular text; therefore, we combine the split function of Python and the "SENT_tokenize" library in NLTK [41] to make clause and take the sentence as the following training sample. A concrete example is shown in Figure 4.

3.2.4. Text Participle. Text cannot be sent into the model in segments for analysis, so we usually cut text into words or phrases with individual meanings, this process is called tokenization, and it is a key step in solving natural language processing problems. In this section, we use the "word tokenization" provided in NLTK to participle the text [41].

3.3. TTP and IOC Extraction. In order to build the relationship between CTI information, many enterprises and institutions in the industry have successively launched their own threat intelligence analysis systems based on knowledge graph for different scenarios of cyber environment and have also established a huge IOC database including, but not limited to, IBM's X-Force Exchange [42], 360 alpha threat analysis platform [43], TianJi Partners' Red Queen platform [44], AlienVault [45], and China's first professional threat intelligence company's microstep online platform [46]; with the help of their huge open source IOC database, we can very well screen out false reports and missing IOC in CTI. So, in this section, we introduce an IOC extraction method that combines the black-and-white mechanism with regular expressions. Different from the existing work, we make use of the open-source IOC database of security vendors to make the black-and-white list of IOCs. In the process of extracting IOCs, blacklist and whitelist are used to screen IOCs while the data of blacklist come from the open-source threat intelligence analysis system of many enterprises and institutions in the industry, as for the data of whitelist, it comes from data of threatrack_iocextract [47], data of a security enterprise, and data collected and sorted out by ourselves. After the screening, regular expressions are used to extract IOCs from APT CTI text to get IOCs not in the whitelist (the work here borrows from open-source project threatrack_iocextract [47], which has well summarized some regular expressions for extracting commonly used IOCs), as shown in Table 2.

Then, the blacklist and whitelist are used for IOC screening. As for the data source, the blacklist data come from the open-source threat intelligence analysis system of many enterprises and institutions in the industry. For the whitelist data, part of it comes from threatrack_iocextract [47], internal data of a security enterprise, and another part was collected and sorted out by ourselves.

However, indicator of compromise (IOC) can sometimes be just pieces of data that lack context. When security personnel is trying to protect the corporate environment, what many security personnel really need is to know about the adversary, the tools and techniques and tactics and processes (TTP) used by the adversary. Fortunately, most APT threat intelligence now uses the ATT&CK matrix [48] as the standard to describe the TTP. So, we use ATT&CK Matrix [48] as the standard to extract TTP. Specifically, in APT threat intelligence text, more than 200 unique technologies described in ATT&CK Matrix [48] are mapped according to their description methods and numbers to extract TTPs. Compared with the method proposed in literature [38], this method is more suitable for the APT threat intelligence that describes TTP in a canonical form and can more accurately extract the TTP summarized in CTI reports.

3.4. Threat Entity Extraction. In this paper, in order to analyze CTI texts to provide strong support for traceability, attack chain building, and system defense, we designed a named entity identification model of threat intelligence called BERT-GRU-BiLSTM-CRF for named entity recognition in threat intelligence based on BERT (bidirectional encoder representations from transformers) [49]; denoting BERT-GRU-BiLSTM-CRF, this proposed method is evaluated on dataset, and good results are obtained. The overall architecture of this model is shown in Figure 5.

In the BERT-GRU-BiLSTM-CRF CTI entity extraction framework mentioned above, firstly, the annotated corpus is processed by BERT [49] pretraining language model to obtain the corresponding word vector, and then the word vector is successively input into BiLSTM module, GRU module, and CRF module for processing. In particular, based on the idea of multilayer RNN [50], some scholars proposed a structure of multilayer stacked LSTM [51, 52]. Stacked LSTM makes the model deeper in depth and extracts deeper features and has achieved good results in a wide range of problems related to prediction. While GRU has the advantages of simple model structure, fewer parameters, and effective reduction of overfitting risk, it is suitable to constructing larger networks with GRU. Based on the idea of multilayer stacked LSTM structure and the characteristics of GRU, we add GRU layer between BiLSTM layer and CRF layer on the basis of traditional BERT + BiLSTM + CRF model and achieved good results.

3.4.1. BERT Layer. In this paper, we use "BERT-base, case" [53] as the pretraining model. BERT network architecture essentially uses the multilayer transformer structure proposed in ≪attention is all you need≫ [54]. It converts the distance between two words at any position into 1 through the attention mechanism and effectively solves the thorny long-term dependency problem in NLP. While dealing with the sentences from Section 3.2, we add a special mark (CLS) at the beginning of the sentences and separate sentences with a mark (SEP). At this time, the output embedding of each word in the sequence consists of three parts: token embedding, segment embedding, and position embedding. The sequence vectors are input into bidirectional transformer for feature extraction, and finally, the sequence vectors with rich semantic features are obtained. After that, the vector is entered into the next layer.

The most critical part of BERT is the transformer. Transformer is a deep neural network based on the "self-attention mechanism," which mainly adjusts the weight coefficient matrix by the degree of correlation between

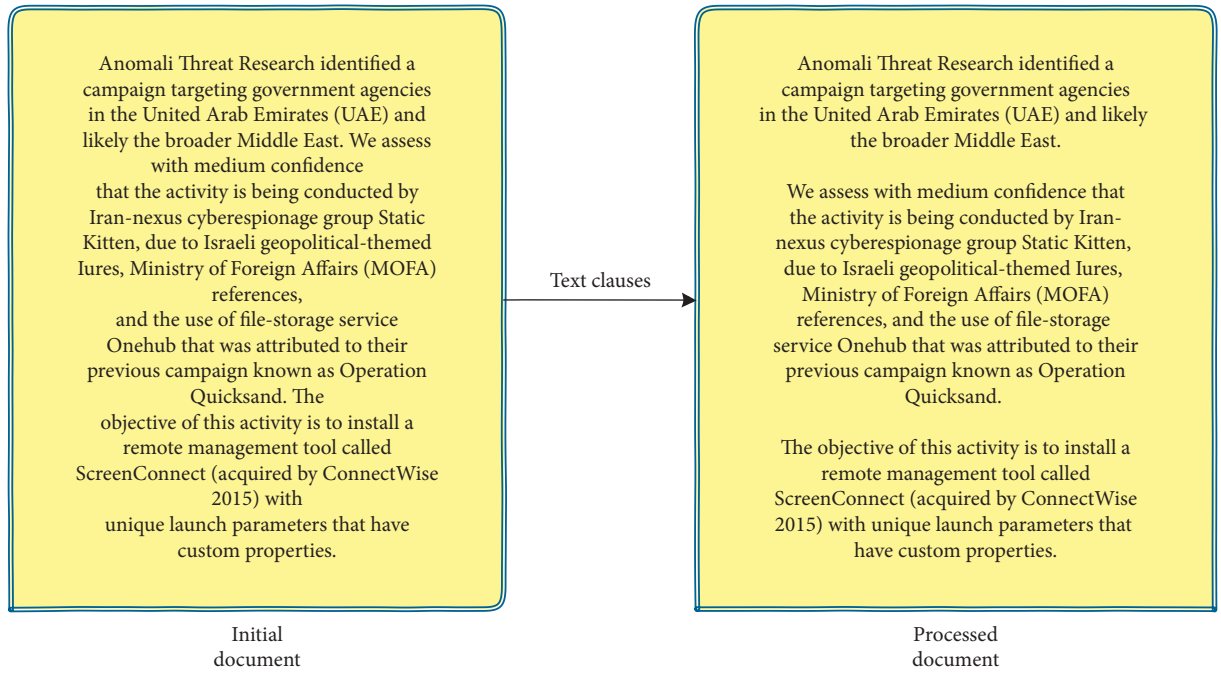

Figure 4: Clause processing.

Table 2: Regular expressions used to extract IOC.

IOC	Pattern
Mac	\b(?:[A-Za-z0-9]{2}:){5}[A-Za-z0-9]{2}\b
E-mail	(?:\b[\@\s=] ∗ [\@\s=\,.]|\“[\\\”]+\”)
IP	(?:(?:25[0-5]|2[0-4][0-9]|1?[0-9]{1,2})\.){3}(?:25[0-5]|2[0-4][0-9]|1?[0-9]{1,2})
cve	CVE-[0-9]{4}-[0-9]{4,6}
Hostname	(?:[A-Za-z0-9\-]{1,64}\.)
Hash	\b[A-Fa-f0-9]{32}(?:[A-Fa-f0-9]{8})?(?:[A-Fa-f0-9]{24})?(?:[A-Fa-f0-9]{64})?\b

words in the same sentence to obtain the representation of words; its calculation process is shown in

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (1)$$

Q, K, and V are word vector matrices and d_k is the embedding dimension. As for details, (1) for each word, 3 vectors Q, K, and V with the same length are generated; (2) the score = $Q * K$ was calculated; (3) the score with Softmax was normalized: Softmaxscore/$\sqrt{d_k}$; (4) the underlying feature information (Q, K, V) was integrated. Through matrix multiplication, the attention that needs to be enhanced can be further increased. We calculated as follows: Softmaxscore/$\sqrt{d_k} * V$; (5) as the depth of calculation increases, there will be gradient problems, so the shortcut structure is added in self-attention to further solve these problems, and formula (1) can be obtained.

3.4.2. BiLSTM Layer. LSTM model is a special recursive neural network, and it has been widely used in NLP tasks because of its excellent performance in solving long-distance dependency problems. For sequence tags in particular, the current output is not only dependent on the current input but is also affected by the previous output, which makes the LSTM model to obtain the context information of words in the sequence marking task. In the LSTM model, the most important concept is the input gate i_t, forgetting gate f_t, output gate o_t, state unit g_t, update state c_t, and output h_t. Input gate i_t: its calculation process is shown in

$$i_t = \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}\right). \quad (2)$$

Forgetting gate f_t: its calculation process is shown in

$$f_t = \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}\right). \quad (3)$$

Output gate o_t: its calculation process is shown in

$$o_t = \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}\right). \quad (4)$$

State unit g_t: record the unit state of current input, which is determined by the last output and the current input; its calculation formula is shown in

$$g_t = \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{(t-1)} + b_{hg}\right). \quad (5)$$

Updated state c_t represents the updated state at time t; its calculation formula is shown in

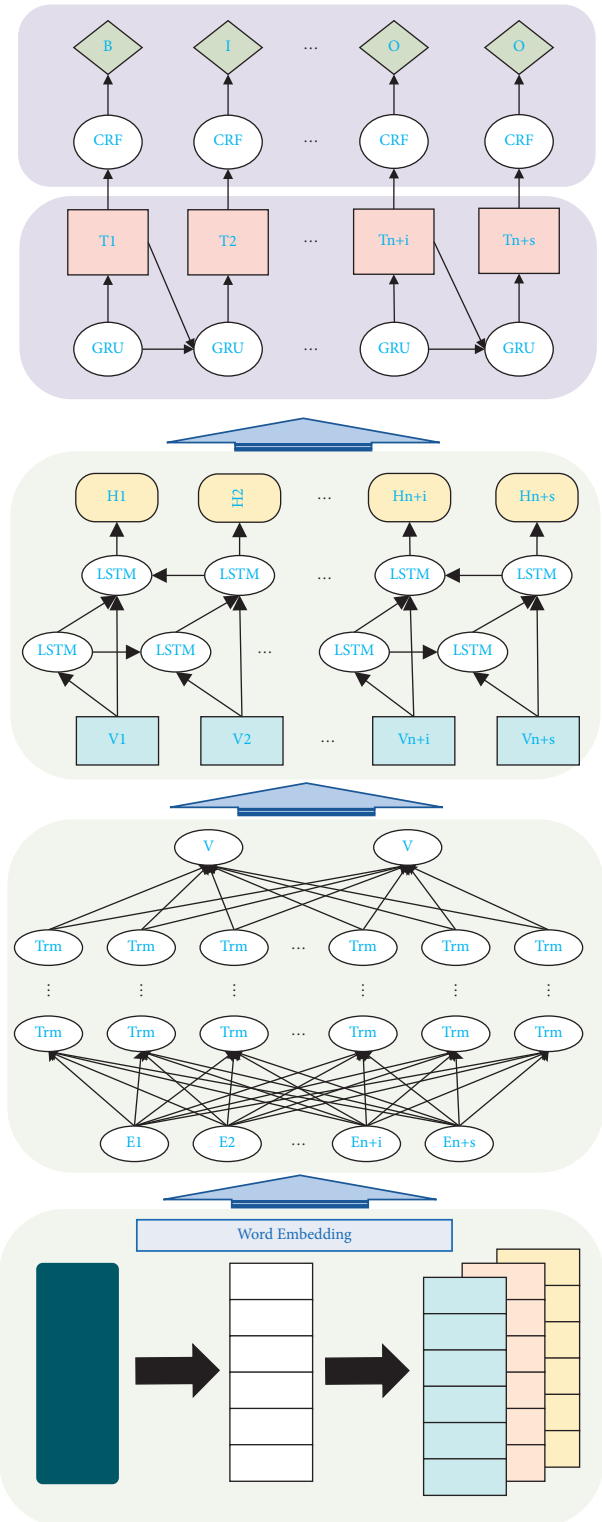

Figure 5: BERT-GRU-BiLSTM-CRF for named entity recognition in APT threat intelligence.

$$c_t = f_t * c_{(t-1)} + i_t * g_t. \tag{6}$$

Output h_t represents the output of the entire LSTM unit at time t; its calculation formula is shown in

$$h_t = o_t * \tanh(c_t). \tag{7}$$

In equations (2) to (7), σ is the activation function, W is the weight matrix, and b is the bias vector.

By analyzing the LSTM model, we can clearly find that the unidirectional LSTM model has an obvious defect that it cannot process the context information at the same time. To solve this problem, Graves and Schmidhuber [55] proposed the BiLSTM (bidirectional long-short term memory) model, namely, the bidirectional long-short term memory network, which is composed of forward LSTM and backward LSTM. BiLSTM can better capture the bidirectional semantic features of sequences than the unidirectional LSTM.

3.4.3. GRU Layer. GRU was proposed by Cho et al. [56] in 2014, which optimized the complex structure of LSTM. Compared with LSTM, GRU can achieve similar results, and it is easier to train and improve training efficiency to a large extent. On the basis of LSTM, GRU mainly makes two changes: (1) GRU combines input gate and forgetting gate in LSTM into one, which is called update gate; (2) the state unit g_t was cancelled and gating was used to perform linear self-update directly. The GRU implementation is as follows:

$$\begin{aligned} z_t &= \sigma(W_z * [h_{t-1}, x_t]), \\ r_t &= \sigma(W_r * [h_{t-1}, x_t]), \end{aligned} \tag{8}$$

$$T_t' = \tanh(W * [r_t * T_{t-1}, x_t]). \tag{9}$$

$$T_t = (1 - z_t) * T_{(t-1)} + z_t * T_t'. \tag{10}$$

Among these formulas, z_t is the update gate, which determines how much information from the past can be transferred to the future; r_t is the reset gate, which determines how much information from the past cannot be passed on to the future. $T_t^{'}$ indicates that the GRU reset gate is used to reset the memorized information (memory information is all the important information recorded by GRU. In the language model, important information such as subject singular or plural, subject gender, and current tense may be preserved); T_t indicates the output of the hidden state at the current time by using the update gate.

3.4.4. CRF Layer. The CRF [57] model plays a key role in the task of sequence labeling. We can get a globally optimal sequence label in the sequence labeling task by CRF model. In this paper, the input sentence sequence is $X = (x_1, x_2, \ldots, x_n)$, the labeled sequence is $Y^* = (y_1^*, y_2^*, \ldots, y_n^*)$, and then the probability of the generation of prediction sequence $Y = (y_1, y_2, \ldots, y_n)$ is

$$p(Y|X) = \frac{e^s(X, Y)}{\sum_{Y^* \in Y_X} s(X, Y^*)}. \tag{11}$$

In formula (8), S is the score function of the prediction sequence $Y = (y_1, y_2, \ldots, y^n)$

$$S(X, Y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=0}^{n} P_{i, y_i}. \tag{12}$$

In formula (13), A represents the transfer fraction matrix, A_{ij} represents the fraction of the label i transferred to the label j, P is the score matrix which is outputted by the GRU layer, and P_{ij} represents the fraction of the No. j label of the No. i word.

In natural language processing (NLP) tasks, CRF is generally used to receive the sequence vector $X = (x_1, x_2, \ldots, x_n)$ and $Y^* = (y_1^*, y_2^*, \ldots, y_n^*)$ passed from the previous layer; then, the prediction sequence $Y = (y_1, y_2, \ldots, y_n)$ can be worked out by formulas (8) and (9).

4. Experimental Results and Analysis

4.1. Dataset of the Experiment

4.1.1. Size of Dataset. In view of the fact that there is no mature and complete APT threat intelligence corpus in the field of named entity recognition, we use the method mentioned in Section 3.1 to collect APT shared threat reports; the data sources include APT threat intelligence regularly released by more than 20 network security vendors such as Recorded Future, Kaspersky, and FireEye. After that, the method mentioned in Section 3.2 is used for data cleaning, and we obtained a total of 120 English APT threat intelligence corpora. Finally, we allocated the training set, verification set, and test set in a ratio of 8:1:1. The specific size of the dataset is shown in Table 3.

4.1.2. Labeling of Dataset. Sequence labeling is an unavoidable problem in natural language processing. The essence of sequence labeling is to label each element of a sequence. Standard practice is to use the method of BIO labeling and label each element as "B-X," "I-X," or "O" where "B-X" means that the element is in a fragment of type X and the element is at the beginning of the fragment, "I-X" means that the element is in a fragment of type X and the element is in the middle of the fragment, and "O" means that the element is not of any type.

As for APT CTI corpus, according to the standard of ATT&CK [48], we use the tool YEDDA [58] to manually annotate according to the format of BIO annotation. In our experiment, based on the actual demand for attribution tracing of APT attacks and establishing knowledge graph of APT threats, five entities are finally selected for annotation, which are attacker, tool, industry, region, and campaign, as shown in Table 4. The distribution of entities on the dataset is shown in Table 5.

4.2. Configuration of Hyperparameters and Index for Evaluation. The parameters of model proposed in this paper are shown as follows:

Table 3: Size of dataset.

Number of samples	Training set	Validation set	Test set
Articles	96	12	12
Sentences	14139	1428	1797

Table 4: Instance of entity labeling.

Entity	BIO
Attacker	B-Attacker/I-Attacker
Tools	B-Tools/I-Tools
Industry	B-Industry/I-Industry
Region	B-Region/I-Region
Campaign	B-Campaign/I-Campaign
Nonentity	O

Table 5: Distribution of entity data.

Entity	Training set	Validation set	Test set
Attacker	1647	92	265
Tools	3310	248	570
Industry	786	129	92
Region	1074	176	104
Campaign	94	12	33

Transformer_layers: It is the number of transformer layers of BERT, and it is used to represent the quantity of learning. The larger the Transformer_layers are, the more the things learned in each iteration of the network, but the training speed will be slower in the meantime.

Embeddings: the dimension of word embeddings of BERT, which indicates the number of dimensional vectors used to represent a word.

attention_heads: the number of attention_head in the encoder layer of BERT. Larger attention_heads means larger subspace, which enables the model to collect semantic knowledge from more angles.

intermediate_size: the number of hidden neurons in the encoder of BERT. Larger the intermediate_size means more parameters of the model, which may result in better fitting effect, but in the meantime, the calculation time of a single round will be longer.

Learning rate: the learning rate of the network.

Batch_size: the number of samples selected for each training.

Epoch: training times. In each epoch, all of the train_data are used for the model training.

Dropout: it is used to prevent network overfitting.

For details, see Table 6.

In terms of evaluation indexes, since the model proposed in this paper extracts multiple entities, in addition to the traditional evaluation indicators, accuracy rate P (formula (14)), recall rate R (formula (15)), and $F1$ value (formula (16)), we also used macroaverage $P_{\text{macro}(P,R,F1)}$ (formulas (17)–(19)) and microaverage $P_{\text{macro}(P,R,F1)}$ (formulas (20)–(21)) to evaluate the overall performance of entity extraction, where the macroaverage $P_{\text{macro}(P,R,F1)}$ is the arithmetic average of the performance indicators of each entity, and the microaverage $P_{\text{macro}(P,R,F1)}$ is the arithmetic average of the performance indicators of instance documents. The calculation method of each evaluation index is as follows:

$$P = \frac{N_c}{N_c + N_d}. \tag{13}$$

$$R = \frac{N_c}{N_a}. \tag{14}$$

$$F1 = \frac{2PR}{P + R}. \tag{15}$$

In formulas (13) to (15), N_c is the number of correctly identified entities, N_d is the number of incorrectly identified entities, and N_a is the number of all entities.

Macroaverage $P_{\text{macro}(P,R,F1)}$:

$$P_{\text{macro}(P)} = \frac{1}{n}\sum_{i=1}^{n} P_i. \tag{16}$$

$$P_{\text{macro}(R)} = \frac{1}{n}\sum_{i=1}^{n} R_i. \tag{17}$$

$$P_{\text{macro}(F1)} = \frac{1}{n}\sum_{i=1}^{n} F1_i. \tag{18}$$

Table 6: Parameters of the experiment.

Parameters	Values
Transformer_layers	12
Embeddings	768
attention_heads	12
intermediate_size	3072
Learning rate	0.01
batch_size	256
dropout_rate	0.4
Epoch	50

Table 7: Comparison of results by macroaverage.

Models	Macro (%)		
	$P_{\text{micro}(p)}$	$P_{\text{micro}(R)}$	$P_{\text{micro}(F1)}$
BERT_LSTM	72.79	59.59	65.53
BERT_BiLSTM	73.56	66.35	69.59
BERT_BiLSTM_CRF	76.36	68.80	72.13
BERT_BiLSTM_GRU_CRF	77.67	69.74	73.20

Microaverage $P_{\text{micro}(P,R,F1)}$:

$$P_{\text{micro}(R)} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}. \tag{19}$$

$$P_{\text{micro}(R)} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i}. \tag{20}$$

$$P_{\text{micro}(R)} = \frac{2 * P_{\text{micro}(R)} * P_{\text{micro}(R)}}{P_{\text{micro}(R)} * P_{\text{micro}(R)}}. \tag{21}$$

In formulas (20) and (21), TP is the number of samples correctly classified as positive examples, FP is the number of samples incorrectly classified as positive examples, and FN is the number of samples incorrectly classified.

4.3. Comparison and Analysis of Correlation Algorithms. We trained and tested our model against the typical NER model based on BERT in our corpus for comparison and evaluated the comprehensive performance of different models by the evaluation indexes: macroaverage and microaverage. The results with macroaverage as the evaluation index are shown in Table 7, and the results with microaverage as the evaluation index are shown in Table 8.

First of all, we can clearly see that the proposed BERT_BiLSTM_GRU_CRF model is superior to other models and achieves the best scores of 73.20% and 73.87% in $P_{\text{macro}(F1)}$ and $P_{\text{micro}(F1)}$.

4.3.1. BERT_LSTM vs BERT_BiLSTM. Based on BERT, BiLSTM can use bidirectional structure to obtain context sequence information, thus the performance of BiLSTM model is significantly improved compared with that of single LSTM, with 4.06% improvement in $P_{\text{macro}(F1)}$ and 5.11% improvement in $P_{\text{micro}(F1)}$.

4.3.2. BERT_BiLSTM vs BERT_BiLSTM_CRF. By comparing the experimental results of BERT_BiLSTM and BERT_BiLSTM_CRF, it can be seen that after the addition of CRF module, $P_{\text{macro}(F1)}$ and $P_{\text{micro}(F1)}$ have been improved by 2.45% and 2.55%, respectively. The main reason is that CRF module can make good use of the relevance of close labels to obtain context information.

4.3.3. BERT_BiLSTM_CRF vs BERT_BiLSTM_GRU_CRF. In the BERT_BiLSTM_CRF model, we add a GRU layer between BiLSTM layer and CRF layer, which makes our BERT_BiLSTM_GRU_CRF model improve $P_{\text{macro}(F1)}$ and $P_{\text{micro}(F1)}$ by 1.07% and 1.25%, respectively, compared with BERT_BiLSTM_CRF model (a model that performs very well in other fields). This is because of the use of multilayer stacked neural network structure, which makes the model deeper in depth and extracts deeper features, thus making the prediction more accurate.

Finally, the performance of different models when extracting different entities is compared, as shown in Figure 6.

4.4. Display of CTI View Results. In order to make better use of our model, we designed a front-end page for CTI View based on VUE's Element UI [59] framework. Here, we select an APT CTI about Lazarus (alias: HIDDEN COBRA) published by ClearSky on August 13, 2020: <Operation 'Dream Job' Widespread North Korean Espionage Campaign>. This intelligence describes ClearSky analysts investigating a suspected Lazarus attack, called Dream Job, and the campaign has been active since the beginning of 2020 and has successfully infected dozens of companies and organizations in Israel and globally; this operation's major targets include defense and government. The targets were infected through complex social work activities, including trojans of PDF files in open-source PDF readers and malicious macros contained in Doc files. By using our CTI View, the article was

Table 8: Comparison of results by microaverage.

Models	Micro (%)		
	$P_{\text{micro}(p)}$	$P_{\text{micro}(R)}$	$P_{\text{micro}(F1)}$
BERT_LSTM	72.13	59.59	64.96
BERT_BiLSTM	74.24	66.35	70.07
BERT_BiLSTM_CRF	76.89	68.80	72.62
BERT_BiLSTM_GRU_CRF	78.52	69.74	73.87

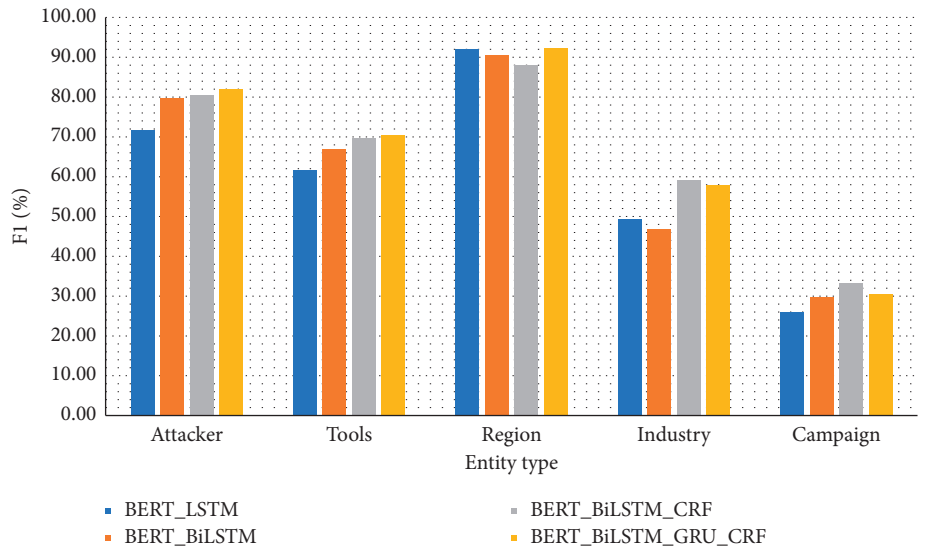

Figure 6: Comparison of different models for multiple entities.

Figure 7: Extraction result of CTI View.

automatically analyzed, and the analysis results are shown in Figure 7.

As shown in Figure 7, CTI View effectively extracted the information of attackers, campaigns, tools used, industries involved, regions involved, and IOC described by the APT CTI we provide.

5. Conclusion and Prospect

5.1. Conclusion. In order to enhance the ability of security researchers to use APT CTI to resist APT attack, in this paper, a new CTI analysis framework, CTI View, is proposed to extract the text information of CTI released by security vendors and analyze this information automatically. To be specific, firstly, this paper proposes an information extraction method using various NLP technologies to extract APT CTI. Then, the IOCs and TTPs information in APT CTI is extracted by regular expression and black-and-white list mechanism. Finally, based on the traditional BERT_BiLSTM_CRF, a threat entity extraction model BERT_BiLSTM_GRU_CRF is designed; based on the actual needs of our threat intelligence corpus, this model achieves better results than other existing models. In short, by using CTI View to analyze CTI in APT field, we can quickly and effectively obtain the key content described by APT CTI, which provides data support for APT threat trend analysis, APT threat attribution, and the strategic thinking of active defense. In short, CTI View can be used to analyze CTI in the APT field to quickly and effectively obtain the key content described in APT threat intelligence, so as to quickly obtain APT knowledge and provide data support for the construction of APT threat knowledge graph. With the help of APT threat knowledge graph, APT threat trend analysis and APT threat attribution tracing are carried out quickly. Meanwhile, active defense strategies can be constructed based on the content extracted from CTI View to provide strategic defense strategy for dealing with highly hidden unknown threats.

5.2. Prospect. This paper focuses on the key technologies involved in APT CTI entity extraction, but the content and technologies involved in this paper are exploratory. There are still the following problems in dealing with APT CTI that need further discussion:

(1) Threat entity and relation extraction based on APT CTI: entity is the most basic element in the knowledge of APT CTI, which describes specific nomenclature reference related to threats. Relationship is used to describe the association between two or more entities at the semantic level, which plays an important role in building a deeper knowledge structure of network APT CTI on the basis of entity recognition. Therefore, the next step is to explore how to combine entity extraction and relation extraction through natural language processing technology and domain knowledge, providing data support for the construction of APT threat knowledge graph. With the help of APT threat knowledge graph, the analysis of APT threat trend is carried out quickly, and there could be more favorable support for APT threat attribution tracing.

(2) Locating the boundary of entities: while CTI View proposed by this paper is extracting entities related to APT threats, due to the strong professionalism in this field, entity boundaries may not be accurately located in the entity identification process if the threat intelligence entity is composed of multiple words, resulting in incomplete entities extracted. Therefore, for the CTI data to be extracted, it is necessary to explore the extraction method fusing multiple features for the accurate localization and extraction of the word feature, character feature, syntactic feature, entity boundary feature, and entity context feature of the CTI entity.

Data Availability

The data used to support the findings of this study are available from the corresponding author (zhouyinghai19@gscaep.ac.cn) upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Tian, "Detection and traceability of high covert unknown threats in cyberspace," *Information and communication technology*, vol. 14, no. 06, pp. 4–7, 2020.

[2] "Advanced persistent threat," 2018, https://en.wikipedia.org/wiki/Advanced_persistent_threat.

[3] "Google Aurora attacks," 2009, https://en.wikipedia.org/wiki/Operation_Aurora.

[4] "Stuxnet attack at Bushehr nuclear power plant in Iran," 2010, https://www.wired.com/images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf.

[5] "Duqu - son of Stuxnet," 2015, https://en.wikipedia.org/wiki/Duqu.

[6] "LUCKYCAT campaign with multiple targets in India and Japan," 2017, https://en.wikipedia.org/wiki/Chinese_intelligence_activity_abroad.

[7] "Dark Seoul cyber attack," 2013, https://en.wikipedia.org/wiki/2013_South_Korea_cyberattack.

[8] "An attack launched by APT group against an unnamed steel plant in Germany resulted in significant damage," 2018, https://www.bbc.com/news/technology-30575104.

[9] "Ukraine power grid cyberattack," 2015, https://en.wikipedia.org/wiki/December_2015_Ukraine_power_grid_cyberattack.

[10] "Bangladesh Bank cyber heist," 2013, https://en.wikipedia.org/wiki/Bangladesh_Bank_robbery.

[11] "Russia hacked the US electric grid," 2018, https://www.vox.com/world/2018/3/28/17170612/russia-hacking-us-power-grid-nuclear-plants.

[12] "Russian interference in the 2018 United States elections," 2018, https://en.wikipedia.org/wiki/Russian_interference_in_the_2018_United_States_elections.

[13] U. S. Escalates Online, "Attacks on Russia's power grid," 2019, https://www.nytimes.com/2019/06/15/us/politics/trump-cyber-russia-grid.html.

[14] "Portuguese energy giant hit by ransomware attack," 2016, https://www.power-eng.com/om/portuguese-energy-giant-hit-by-ransomware-attack/#gref.
[15] "Colonial Pipeline cyber attack," 2021, https://en.wikipedia.org/wiki/Colonial_Pipeline_cyber_attack.
[16] "Computer giant Acer hit by $50 million ransomware attack," 2021, https://www.bleepingcomputer.com/news/security/computer-giant-acer-hit-by-50-million-ransomware-attack/.
[17] "Database leak exposes CPF of almost the entire population of Brazil," 2021, https://olhardigital.com.br/en/2021/01/20/safety/database-leak-exposes-cpf-of-almost-the-entire-population-of-brazil/.
[18] "DDoS attack took down the websites of more than 200 Belgium organisations," 2019, https://www.zdnet.com/article/this-massive-ddos-attack-took-large-sections-of-a-countrys-internet-offline/.
[19] *533 Million Facebook Users' Phone Numbers and Personal Data Have Been Leaked Online*, https://www.businessinsider.com/stolen-data-of-533-million-facebook-users-leaked-online-2021-4, 2021.
[20] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2020.
[21] O. Catakoglu, M. Balduzzi, and D. Balzarotti, "Automatic extraction of indicators of compromise for web applications," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 333–343, Geneva, Switzerland, April 2016.
[22] R. McMillan and K. Pratap, *Market Guide for Security Threat Intelligence Services*, Gartner report (G00259127), 2014.
[23] L. Yue, P. Liu, and H. Wang, "Overview of network security threat intelligence sharing and exchange," *Computer research and development*, vol. 57, no. 10, p. 2052, 2020.
[24] "Stix," 2016, https://oasis-open.github.io/cti-documentation/stix/intro.
[25] "Capec," 2014, http://capec.mitre.org/about/index.html.
[26] X. Liao, K. Yuan, X. F. Wang, Z. Li, and L. Xing, "Acing the IoC game: toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 755–766, New York, NY, USA, October 2016.
[27] 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 *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Budapest, Hungary, July 2019.
[28] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li, "TIMiner: automatically extracting and analyzing categorized cyber threat intelligence from social data," *Computers & Security*, vol. 95, Article ID 101867, 2020.
[29] Y. Wang, Z. Tian, Y. Sun, X. Du, and N. Guizani, "LocJury: an IBN-based location privacy preserving scheme for IoCV," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5028–5037, 2021.
[30] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan, "Extracting information about security vulnerabilities from web text," in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 257–260, Lyon, France, July 2011.
[31] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.
[32] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text," in *Proceedings of the 2013 IEEE Seventh International Conference on Semantic Computing*, pp. 252–259, Irvine, CA, USA, September 2013.
[33] C. L. Jones, R. A. Bridges, K. M. T. Huffer, J. Laval, and A. Bouras, "Towards a relation extraction framework for cyber-security concepts," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, pp. 1–4, New York, NY, USA, April 2015.
[34] R. Manikandan, K. Madgula, and S. Saha, "TeamDL at SemEval-2018 task 8: cybersecurity text analysis using convolutional neural network and conditional random fields," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, pp. 868–873, New York, NY, USA, January 2018.
[35] N. Dionísio, F. Alves, P. M. Ferreira, F. Ferraro, and T. Finin, "Cyberthreat detection from twitter using deep neural networks," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Budapest, Hungary, July 2019.
[36] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for Internet of things via ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810–5818, 2021.
[37] Y. Sun, Z. Tian, M. Li, S. Su, X. Du, and M. Guizani, "Honeypot identification in softwarized industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5542–5551, 2021.
[38] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttpdrill: automatic and accurate extraction of threat actions from unstructured text of cti sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 103–115, New York, NY, USA, December 2017.
[39] "pyppeteer," 2018, https://pyppeteer.github.io/pyppeteer/.
[40] "pdfminer," 2018, https://github.com/euske/pdfminer/.
[41] "nltk," 2021, http://www.nltk.org/.
[42] "X-force exchange," 2021, https://exchange.xforce.ibmcloud.com.
[43] "Alpha," 2020, https://ti.360.net.
[44] "RedQueen," 2018, https://redqueen.tj-un.com/IntelHome.html.
[45] "AlienVault," 2019, https://otx.alienvault.com.
[46] "Threatbook," 2021, https://x.threatbook.cn.
[47] 2019 https://github.com/threatrack_iocextracthttps://github.com/threatrack/threatrack_iocextract.
[48] "ATT&CK," 2015, https://attack.mitre.org/.
[49] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional Transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.
[50] M. Hermans and B. Schrauwen, "Training and analyzing deep recurrent neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, USA, December 2013.
[51] H. Gasmi, J. Laval, and A. Bouras, "Information extraction of cybersecurity concepts: an LSTM approach," *Applied Sciences*, vol. 9, no. 19, p. 3945, 2019.
[52] "Stacked long short-term memory networks," 2021, https://machinelearningmastery.com/stacked-long-short-term-memory-networks/.
[53] "BERT-Base," 2018, https://storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.
[54] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all You need," 2017, https://arxiv.org/abs/1706.03762.

[55] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks : The Official Journal of the International Neural Network Society*, vol. 18, no. 5–6, pp. 602–610, 2005.

[56] K. Cho, B. V. Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Computer Science*, 2014.

[57] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning*, pp. 282–289, San Francisco, CA, USA, April 2001.

[58] J. Yang, Y. Zhang, L. Li, and X. Li, "YEDDA: a lightweight collaborative text span annotation tool," in *Proceedings of the ACL 2018, System Demonstrations*, Melbourne, Australia, July 2018.

[59] "Element UI," 2021, https://madewithvuejs.com/element-ui.