MODELING AND IDENTIFICATION OF HETEROGENEOUS INFORMATION NETWORK-BASED CYBER THREAT INTELLIGENCE (HINCTI)

[#1]BADDAM SRIKANTH REDDY, Dept of *MCA*,

[#2]P.SATHISH, *Assistant Professor*,

[#3]Dr.V.BAPUJI, *Associate Professor& HOD*,

***Department of Master of Computer Application*,**

VAAGESWARI COLLEGE OF ENGINEERING, KARIMANGAR, TELANAGANA

ABSTRACT: The surge in cyberattacks has exacerbated the cybersecurity problem. Navigating the complex and dynamic cyber environment requires the use of cyber threat intelligence. However, the majority of cyber threat intelligence is unstructured, and security analysts struggle to deal with such a big amount of data. This work proposes a novel threat intelligence information extraction approach that integrates many models, with four fundamental steps: entity extraction, co reference resolution, relation extraction, and knowledge graph creation. In the entity extraction task, a multihead self-attention technique is employed to extract the dependency relationships between words. In the co reference resolution procedure, contextual data and mention embedding are coupled to improve mention representation. In the meantime, a convolution neural network is utilized to extract features from different dimensions. Part of speech, mention breadth, entity type, and distance between entity pairings are all included to the relation extraction task to improve the embedding representation. Finally, a knowledge graph is built to explicitly present entities and their interactions. When compared to the baseline model, our model's F1 score on entity extraction, co reference resolution, and relation extraction is improved by at least 8.87, 9.82, and 10.56 points, respectively. The knowledge graph in Neo4j exemplifies how well our technology works.

1. INTRODUCTION

Cyberattacks on modern IT systems can cause varying degrees of damage. Monitoring devices around the clock, collecting and analyzing data, and producing security reports are all necessary to address this issue. In order to better comprehend the threat environment and coordinate responses to attacks that individuals are unaware of, security professionals recommend implementing Cyber Threat Intelligence (CTI) for cyber security. According to a Gartner analysis from 2013, threat intelligence is data on actual or potential dangers to an asset. Threat information can take several forms, such as hypothetical situations, mechanisms, indicators, revelations, and proposed countermeasures. There are two key distinctions between intrusion detection and monitoring systems and threat intelligence platforms. Cyber threat intelligence, on the other hand, is compiled after an attack and the subsequent response. It could be used to inform risk management decisions by providing access to expert knowledge. Systems designed to detect intrusions will issue a warning whenever one occurs. However, you can get threat intelligence from places like online discussion forums, social media, and security provider alerts. Understanding the mechanics and causes of hacks is helpful. Despite this, threat intelligence has expanded greatly due to the Internet's complexity, the diversity of attack methods, and the abundance of available security measures.

There has been a recent trend in the dissemination of cyber threat intelligence in plain English, with key terms dispersed throughout the text and convoluted relationships. This makes it more difficult to access, analyze, and distribute data. There is a lot of work for security specialists to undertake because of all the alerts. Thus, many warnings are ignored and eventually forgotten. There is an immediate need to address the analysis and management of threat information.

It takes a lot of time, effort, and knowledge of cyber security to manually analyze threat data. This complicates efforts to counteract the rising rate of attacks. Extraction of structured knowledge from unstructured threat intelligence has been the subject of extensive research because of its critical nature. The four main methods that make up this extraction approach are entity extraction, co-reference resolution, relation extraction, and knowledge graph construction.

The following issues must be resolved for automated threat intelligence use: There are specific aspects of the threat intelligence domain that set it apart from other domains. Threat entities including cyber groups, attack tactics, malware, etc. can be difficult to find when employing a broad subject entity extraction strategy.A threat intelligence dossier may contain multiple references to the same company. You need to take a holistic view of the issue and extract semantic information to determine if the references are referring to the same entity.Sentence patterns in reporting about threats may be difficult to follow. Understanding the connections between ideas typically requires more than one line.

Few threat information datasets are available to the general public.In order to better gather threat intelligence, this study proposes a different approach. To construct a knowledge graph, these four phases must be completed: "pulling entities," "resolving co-references," "extracting relationships," and "building." This strategy resolves these issues.

In addition, Zhou et al. developed a technique for extracting APT threat intelligence, however it could only collect related components. Vulcan assumed semantic links between descriptive or static CTI data extracted from unstructured text. However, they don't provide a full picture of the entities and relationships seen in threat intelligence. This paper's key arguments are outlined briefly below. A hybrid approach is utilized to get insight in the field of threat intelligence. The model may structure raw threat intelligence data by converting it into a knowledge graph. The threat intelligence components and their interconnections are displayed in a knowledge graph that is kept current with the Neo4j graph database. Experts in the field of security are better able to comprehend attacks and build up defenses with the aid of this data and the decisions it facilitates.

Entity extraction using multiple attention colors and POS (EEMAP) is a novel entity extraction model. Incorporating "multithread self-attention," the model may generate entity-relevant vector representations. The feature vectors of a recurrent neural network model are then supplemented with vector representations. The text entities are recovered after the combined results are forwarded to a linear layer to create sequence labels.

There is a novel approach to fixing co-reference issues called Co-reference Resolution with CNN and POS (CRCP). The method improves upon mention representation by incorporating environmental data into the embedding process. To compensate for the low recall rate of conventional co-reference resolution approaches, a convolution neural network can be used to extract various aspects of the aforementioned characteristics.

This document compiles and summarizes 227 threat intelligence papers sourced from security vendor newsletters, weblogs, and discussion boards. This is done due to the scarcity of publicly available threat databases. The study's experimental findings corroborate our

hypotheses.

2. RELATED WORK

This section focuses primarily on previous studies conducted on our topic. We take a close look at state-of-the-art entity extraction, co-reference resolution, and relation extraction methods.

Entity Extraction

The method of entity extraction is crucial to NLP. In order to create semi-structured or structured data, it searches through unstructured text for mentions of people, places, businesses, and other entities. This allows for multiple perspectives on the written material. Name sequences play a crucial role in the entity extraction process.

Rules and dictionaries were the mainstays of early entity extraction research. Many rule templates were created by experts, with string matching serving as the primary way of application. Although effective for datasets containing recurrent patterns, this approach is not generalizable and cannot accommodate for novel circumstances. Statistical machine learning can be used to enhance the entity extraction procedure used in cyber security. This approach doesn't call for the creation of rule files manually and may be used anywhere. Models in the field of statistical machine learning include the likes of the Hidden Markov model (HMM), the maximum entropy model (MEM), and the conditional random field (CRF). In order to create a one-of-a-kind cybersecurity ontology, Joshi et al. proposed employing the CRF technique to extract items, concepts, and relationships from cybersecurity blogs and announcements.

Support vector machines (SVMs) were utilized by Mulwad et al. as a predictor to learn about the dynamics of attacks and their outcomes. To prevent the MEM from becoming overly accurate during training, Bridges et al. tested it on a number of security-related corpora. The aforementioned methods are more trustworthy than rule-based ones, but they require extensive work mining text features, making them difficult to use with tiny datasets. The use of deep learning has spread to numerous fields, allowing neural networks to excel at previously difficult tasks such as "extracting entities." Chiu and Nichol's model, for instance, combined BiLSTM and CNN to uncover features at both the character and token levels. They developed a matching algorithm and a novel vocabulary encoding system.

CNN was initially utilized by Dion'sio et al. to determine if the collected tweets contained any data pertaining to IT system security. Then, they employed BiLSTM to identify the well-known businesses and receive warnings about potential threats. Using a technique called dependency analysis, Wu et al. drew methods, tools, and entities from e-commerce threat intelligence to investigate emerging assault trends. Cybersecurity-related concepts were extracted using a BiLSTM-CRF model by Gasmi et al. Three LSTM-based models were then compared to the model. To further depict the inter-IOC dependencies, Zhao and his team constructed a heterogeneous information network using a multigranular attention approach. This allowed for more precise findings.

Co referenceResolution

If the two pairs of references are connected, the coreference resolution will be able to notify you. There may be several occurrences of an object while performing relation extraction at the document level. Two references are said to be "coreferential" if they both refer to the same thing. In one instance, we read, "the malware also silently downloads and installs a known malicious app called ister59.apk (detected as Android. Reputation from the following URL." The identical flaw was discovered by both Ister59.apk andAndroid.Reputation.3. Numerous tasks require the ability to resolve coreferences, including multi-party communication, the construction of abstract meaning, and the identification of causal links between events.

Relation Extraction

Extraction of relationship information from text

is called "relationship extraction." Learning the relationship between two sections of the same sentence was an early focus of research. More and more relationship information must be cobbled together from various lines if relation extraction is to be performed at the document level.

At the document level, the most common relation extraction strategies are either transformation-based or graph-based. Graph-based methods, in example, generate graphs from documents to provide a more intuitive description of entity hierarchies. In order to determine how entities are linked, Zeng et al. constructed document graphs at the mention and entity levels and proposed a new path inference mechanism. To illustrate the intricate relationships between token-level, mention-level, and entity-level components in a twenty-interpretation document, Sun et al. proposed a dual-channel hierarchical graph convolution neural network, or DHGCN. In transformer-based approaches, pre-trained models (such as Bert, Roberta, ERNIE, etc.) are used to assign a representation to each document word. An inter-sentence attention mechanism was employed by Yuan et al. to dynamically bring together essential sentence features when designing a gating function to merge sentence-level information with document-level features. To anticipate entity-level association vectors and collect both local and global data, Zhang et al. developed a U-shaped segmentation module. In order to handle multiunit and multilevel issues in document-level relation extraction, Zhou et al. proposed adopting local context pooling to improve entity embedding and adaptive Thresholding to decrease optimization cost.

3. METHODOLOGY

Model Framework

In this research, we suggest a unified approach to gathering threat intelligence data based on existing models. The proposed system incorporates NLR, CLR, DLR, and KG to accomplish the following: named entity identification, coreference resolution, relation extraction at the document level, and the construction of knowledge graphs. Figure 1 depicts the overall structure of the system. At first, the data is transformed into POS-enhanced embeddings using the BERT and NLTK Python packages. Three distinct models—one for named-object identification, one for coreference resolution, and one for relation extraction at the document level—are given the embedded data. The results are then sorted before being added as triples to the tree of knowledge.

Encoding Layer

In this work, BERT is employed to provide a comprehensive comprehension of word meaning in place of the customary encoding layer, which relies on random word embedding. Mention embedding's ability to characterize text is enhanced by using part-of-speech embedding. The encoder is the pretrained model BERT, and the special tags "[CLS]" and "[SEP]" are appended to the beginning of the text. Every occurrence in the text is preceded and followed by the identifier " ". Tokenizing the document results in a new document, D xt l 1, where xt is the word at position t (see Figure 1). In this section, the document is encoded in order to generate the H representation of context using the BERT-base.

$$H = \mathrm{BERT}([x_1, \ldots, x_l]) = [h_1, \ldots, h_l], \quad (1)$$

The hidden size is represented by d1, therefore H = Rl + d1. The POS order of the text is determined using NLTK. The POS embedding matrix P is constructed as follows:

$$P = \mathrm{Pos}([x_1, \ldots, x_l]) = [p_1, \ldots, p_l], \quad (2)$$

Each has a POS embedding of dimensions P Rld2 and d2. Each token is provided with a more suitable word representation based on the context through contextual and POS embedding.

$$C = [h_1 \circ p_1, \ldots, h_l \circ p_l] = [c_1, \ldots, c_l], \quad (3)$$

where $C \in R^{l \times (d_1 + d_2)}$, and $\circ$ indicates the linking operation.

Entity Extraction

Our entity extraction model employs a multihead self-attention strategy to obtain vector representations of relevant entities. This algorithm can determine the significance of the relationships between each pair of words by assigning different weights to each token representation. Using several attention heads to search for features in various representational subspaces can greatly accelerate model training. In this approach, the attention layer is provided with a set of POS-enhanced token representations that may be used to appropriately embed the current word

.

$$\text{head}_i = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V,$$

$$A = \text{MultiHead}_i Q, K, V \qquad (4)$$

$$= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_{H_1}\right) W_A,$$

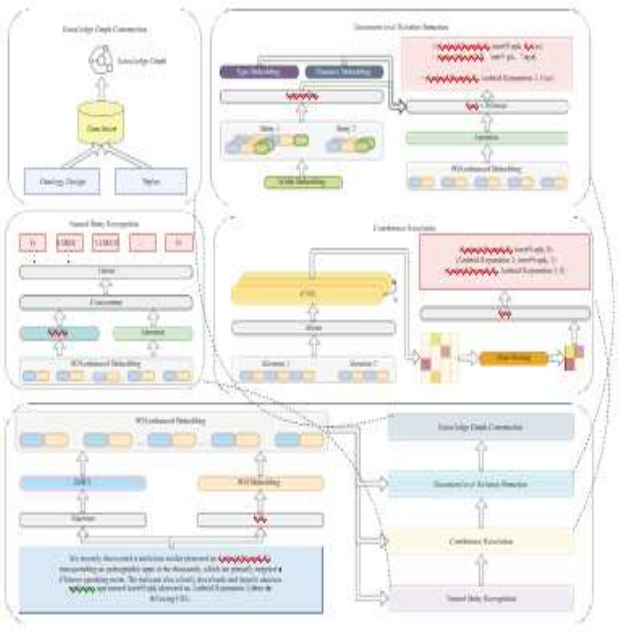


FIGURE1:Theproposedthreatintelligenceinformationextractionsystem.

The question sequence is denoted by Q, the key sequence by K, and the value sequence by V. Key sequence dimension (dk) and value sequence (V) notation.

BiLSTM has been demonstrated to be effective at recording contextual semantic information; this study recommends using it to learn more about the past and future usage of the present term. Layers of connections, backward LSTMs, and forward LSTMs are what make up a BiLSTM network. Each LSTM is constructed from several memory nodes, or subnetworks. At each instant in time, an LSTM storage module is constructed using the current input word embedding operation, the hidden vector from the previous instant, and the storage cell vector from the instant before that.

The BiLSTM is fed a sequence of POS-enhanced token representations, and out comes a feature vector.

$$B = \text{BiLSTM}[c_1, \ldots, c_1] = \left[\vec{b}_1 \circ \overleftarrow{b}_1, \ldots, \vec{b}_1 \circ \overleftarrow{b}_l\right]. \qquad (5)$$

Sequence labels are then generated by feeding the combined feature vectors and relevant contextual embeddings through a linear classifier.

$$y_{ner} = \arg\max\left[W_1 (A \circ B) + b_1'\right]. \qquad (6)$$

Coreference Resolution.

Finding out if two references to the same item actually refer to the same thing is what "resolving a coreference" does. The task of resolving cross-references is viewed as a classification challenge in this study's approach.To create POS-improved versions of mention tokens, the model calculates an average vector for each token.

$$\text{mention}_1 = \text{Mean}\left(c_{1_g}, \ldots, c_{1_l}\right),$$
$$\text{mention}_2 = \text{Mean}\left(c_{2_g}, \ldots, c_{2_l}\right). \qquad (7)$$

CNN employs a moving window to get depth information, which aids in the struggle against reliance on remote sources. A convolution layer is often a single layer that performs convolutional operations on word vectors with the help of a convolution kernel.In the convolution layer, a pooling layer eliminates

redundancy and prevents overrating after the representations have been shrunk and compressed. From the feature values produced by each layer following the convolution layer, the maximum feature value is selected and the remaining features are omitted in this novel approach.

$$\begin{aligned} \text{Mention} - \text{Pair}_i &= \text{Conv}_i(\text{mention}_1 \cdot \text{mention}_2), \\ M &= \text{Concat}(\ldots, \ldots, \text{Mention} - \text{Pair}_N)W_M, \\ \text{MP} &= \text{MaxPooling}(M). \end{aligned} \tag{8}$$

The label probability, or whether two mentions refer to the same item, is calculated using the tanh activation function after the aggregated feature vector of mention pairs has been obtained.

$$y_{\text{CR}} = \tanh(W_2 \cdot \text{MP} + b_2'). \tag{9}$$

Next, sequence labels are generated using the entity extraction model mentioned in Section, and the corresponding occurrences are located using the labels. The coreference resolution mechanism is then used to determine if the mentions pertain to the same entity.

4. RESULT ANALYSIS

Performance on Entity Extraction

Our EEMAP-BERT model is compared to state-of-the-art baselines for the entity extraction task in Table 1. Instead of using the maintained model BERT as the encoder, EEMAP-WE employs a random word embedding. In comparison to the BiLSTM and BiLSTM-CRF baselines (Table 1), our model significantly outperformed them in terms of precision, memory, F1 score, and exact-match accuracy. Extra points of 9.94 and 8.87 were added to the F1 total. The impact that each module had on the overall performance of the model was then determined through ablation experiments.

After POS embedding was removed (labeled NoPOS), the F1 score and exact-match accuracy declined by 0.67 and 0.74 points, respectively. According to threat intelligence, these two forms of embeddings significantly affect the model's performance because entities are primarily words and verbs.

The multihead self-attention (attention) layer was next to be removed. The exact-match precision and F1 score reportedly fell by 0.42 and 0.56 respectively. The experiment results demonstrate that the multihead self-attention mechanism can facilitate the recognition of crucial context and the acquisition of knowledge about distant, interdependent entities. Then, the POS integration and the multihead self-attention layer were disabled (labeled No POS + No Attention). The large decrease in performance is evidenced by the 2.76-point drop in the F1 score and the 3.45-point drop in the exact-match accuracy score. To further investigate how the model encoder impacts performance, we also analyzed the usefulness of random word embedding and BERT. It was discovered that randomly inserting words into models drastically reduces their accuracy. Both the F1 and exact-match scores dropped significantly, by 9.65 and 13.01 points, respectively. The trials validated the hypothesis that a model trained on a large corpus could identify generic representations of language. As a result, it was better able to multitask and complete its goals sooner. As can be seen in Figure 4, both the F1 score and the exact-match accuracy improve when there are five attention heads involved. That's why we've got H2 at five. The fine-grained performance for a single category is broken down in great detail in Figure 5. This opens the door to exploring the performance of the POS embedding and attention mechanism on new data kinds.

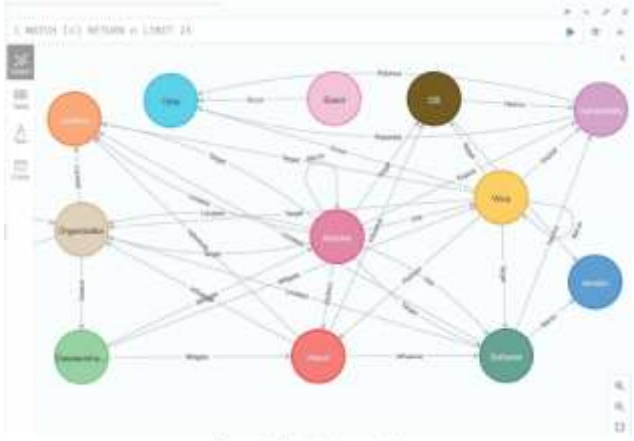
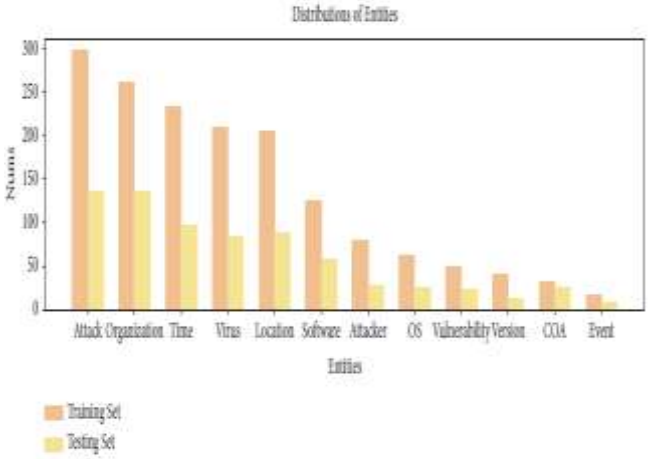


Figure 2: Threat intelligence ontology.



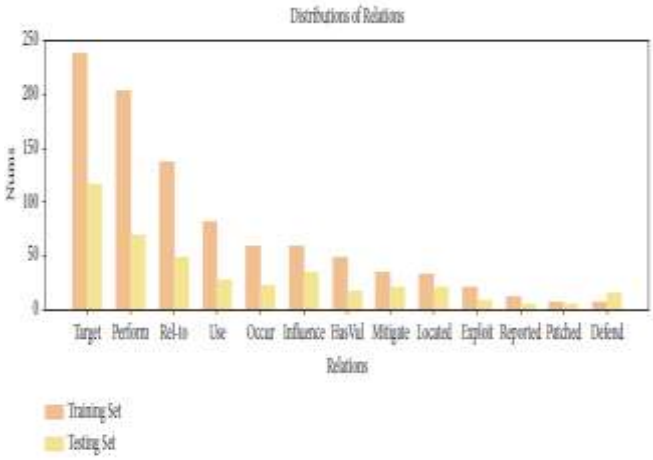


Figure 3: The distribution of entities and relationships.

Table 1: The performance of the entity extraction task.

Model	Precision	Recall	F1	Accuracy
EEMAP-WE	69.23	67.71	68.46	61.12
BiLSTM [11]	69.80	66.62	68.17	62.11
BiLSTM-CRF [13]	70.10	68.40	69.24	62.77
EEMAP-BERT (our model)	79.02	77.22	78.11	74.13
No POS	78.43	76.46	77.44	73.39
No attention	78.56	76.84	77.69	73.57
No POS + no attention	75.97	74.74	75.35	70.68

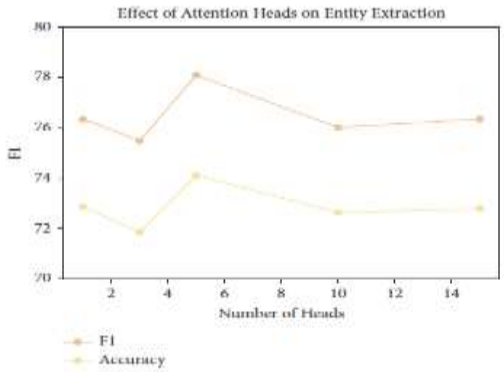
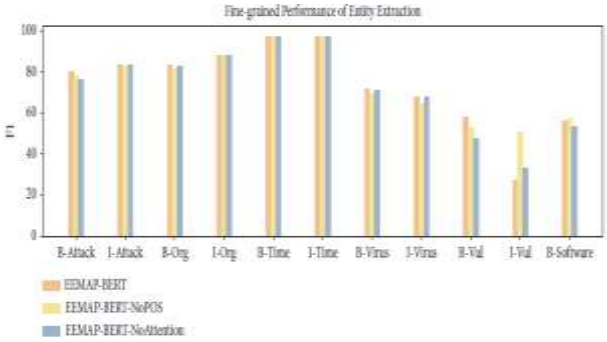


Figure 4: The effect of attention heads on entity extraction.



(a)

Figure 5: Continued.

5. CONCLUSION

This research presents a hybrid strategy for information extraction in threat intelligence, one that takes into account the interplay between entity extraction, coreference resolution, connection extraction, and knowledge graph construction. The entity extraction model incorporates the multihead self-attention technique, which allows for the identification of pertinent contextual vectors. Environment data and encoding of references are employed for coreference resolution. To identify features, a convolutional neural network similarly integrates data from multiple layers. The relation extraction model is then enhanced with additional features, such as the identification of entity pairs and the separation of parts of speech and reference breadth. Experiments show that compared to baselines, our model increases entity extraction by 8.87 points, coreference resolution by 9.82 points, and relation extraction by 10.56 points in terms of the F1 score. The next stage is to construct a knowledge graph for threat intelligence, which will display any potential semantic connections between

components. Finally, our approach can quickly extract data from a wide variety of papers and identify connections between key elements. It provides a solid groundwork for tracking down threats and understanding what's happening. Our entity and relationship classification system will be enhanced, and new instances will be added to our dataset, as a result of the planned work. Knowledge thinking will also make use of a knowledge graph. We can gain more knowledge from this.

REFERENCES

[1] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network," Journal of Parallel and Distributed Computing, vol. 164, pp. 55–68, 2022.

[2] P. Kumar, G. P. Gupta, and R. Tripathi, "Design of anomalybased intrusion detection system using fog computing for IoT network," Automatic Control and Computer Sciences, vol. 55, no. 2, pp. 137–147, 2021.

[3] P. Kumar, R. Tripathi, and P. G. Gupta, "P2IDF: a privacypreserving based intrusion detection framework for software de?ned Internet of ,ings-fog (SDIoT-Fog)," in Proceedingsof the 2021 International Conference on Distributed Computingand Networking, pp. 37–42, Nara Japan, January 2021.

[4] Y. Zhou, Y. Tang, M. Yi, C. Xi, and H. Lu, "CTI view: APT threat intelligence analysis system," Security and CommunicationNetworks, vol. 202215 pages, Article ID 9875199, 2022.

[5] H. Jo, Y. Lee, and S. Shin, "Vulcan: automatic extraction and analysis of cyber threat intelligence from unstructured text," Computers & Security, vol. 120, Article ID 102763, 2022.

[6] X. Liu and J. Li, "Key-based method for extracting entities from XML data," Journal of Computer Research and Development, vol. 51, no. 1, pp. 64–75, 2014.

[7] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text," in Proceedings of the 2013IEEE Seventh International Conference on Semantic Computing, pp. 252–259, IEEE, Irvine, CA, USA, September 2013.

[8] 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 InternationalConferences on Web Intelligence and IntelligentAgent Technology, pp. 257–260, IEEE, Lyon, France, August 2011.

[9] R. A. Bridges, K. M. Hu4er, C. L. Jones, M. D. Iannacone, and J. R. Goodall, "Cybersecurity automated information extraction techniques: drawbacks of current methods, and enhanced extractors," in Proceedings of the 2017 16th IEEEInternational Conference on Machine Learning and Applications
(ICMLA), pp. 437–442, IEEE, Cancun, Mexico, December 2017.

[10] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," Transactions of the associationfor computational linguistics, vol. 4, pp. 357–370, 2016.