Hindawi
Security and Communication Networks
Volume 2021, Article ID 5586335, 12 pages
https://doi.org/10.1155/2021/5586335

Research Article

EX-Action: Automatically Extracting Threat Actions from Cyber Threat Intelligence Report Based on Multimodal Learning

Huixia Zhang,[1,2] **Guowei Shen**,[1,2] **Chun Guo**,[1,2] **Yunhe Cui**,[1,2] **and Chaohui Jiang**[1,2]

[1]*College of Computer Science and Technology, Guizhou University, Guiyang 550025, China*
[2]*Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China*

Correspondence should be addressed to Guowei Shen; gwshen@gzu.edu.cn and Chun Guo; gc_gzedu@163.com

Received 4 February 2021; Accepted 7 May 2021; Published 27 May 2021

Academic Editor: Liguo Zhang

With the increasing complexity of network attacks, an active defense based on intelligence sharing becomes crucial. There is an important issue in intelligence analysis that automatically extracts threat actions from cyber threat intelligence (CTI) reports. To address this problem, we propose EX-Action, a framework for extracting threat actions from CTI reports. EX-Action finds threat actions by employing the natural language processing (NLP) technology and identifies actions by a multimodal learning algorithm. At the same time, a metric is used to evaluate the information completeness of the extracted action obtained by EX-Action. By the experiment on the CTI reports that consisted of sentences with complex structure, the experimental result indicates that EX-Action can achieve better performance than two state-of-the-art action extraction methods in terms of accuracy, recall, precision, and F1-score.

1. Introduction

With the increasing amount of information in modern society, advanced persistent threat (APT) attacks, as a new development of cyber security, have gradually become one of the main attack methods. APT attacks have many characteristics, such as long duration, complex attack methods, and strong concealment. The traditional defense method is a passive defense method, which mainly relies on security equipment and rules matching to generate an alarm for static control. It is not suitable for the protection of APT, 0 day attacks, and other new network security threats [1]. Therefore, many organizations have aimed to develop timely, relevant, and actionable CTI about emerging threats and key threat actors to enable effective cybersecurity decisions [2].

CTI is a kind of information that records current and former security threats [3], which contain information such as the reasoning, context mechanism, observable indicators, mitigation measures, and countermeasures of attacks. It is extremely time-consuming for security practitioners to analyze and utilize multisource and unstructured CTI reports. Therefore, automatic and efficient information extraction from unstructured CTI reports has become one of the main research directions.

Information extraction is the extraction of valuable information from unstructured CTI reports. The extracted information mainly includes cybersecurity entities and its relationship. Cybersecurity entity recognition identifies named entities in the cybersecurity field, which mainly include names of persons, organizations, places, and some security terms. Entity-relationship extraction is to extract the relationships between security entities in unstructured CTI reports. It is mainly for the triple extraction of known entities and predefined relationships. Complex relationships between entities with contextual connections are hard to be identified.

The action consists of the subject, the verb, and the object. Actions not only describe the attack behaviors in the attack process but also include non-predefined entities and their contextual semantic relationships. Therefore, actions are crucial for CTI reports. The subject and the object in actions correspond to a pair of security entities, and the verb describes the semantic relationship between the entity pair.

The entities and the relationship between them do not need to be predefined in the proposed method.

At present, the extraction of actions is mainly based on semantic dependency [4], and the ontology model [5] is used to identify them. Therefore, there are mainly the following challenges in the extraction and identification of threat actions:

(1) Threat actions cannot be accurately extracted in unstructured CTI reports just relying on their semantic dependency.

(2) Relying on ontology methods to identify actions will lose some undefined key threat actions.

(3) Information content of extracted threat actions is incomplete, and it is difficult to measure the information content of the extracted threat actions.

In this study, we propose a multimodal learning approach, named EX-Action, to accurately extract and automatically identify threat actions in unstructured CTI reports. EX-Action is a method based on the combination of mutual information and NLP technology. It can extract more actions based on the syntactic structure. Three main contributions of this study are listed as follows:

(1) We propose an actions extraction framework, named EX-Action. EX-Action extracts threat actions from the unstructured CTI reports that consisted of complex sentence structure by syntactic rule matching. And then, it identifies threat actions by a multimodal learning algorithm.

(2) We use an evaluation indicator named normalized mutual information (NMI) [6] to measure the difference of information content of threat actions, which quantifies the completeness of the information content of threat actions.

(3) We apply EX-Action to extract 18210 actions from 243 unstructured CTI reports, and the experimental result shows that the obtained accuracy, F1-score, and NMI of EX-Action are 79.09%, 85.58%, and 85.26%, respectively.

The rest of this study is organized as follows. We list the related work of extracting information from the CTI report in Section 2. In Section 3, we introduce the EX-Action framework and describe it. Section 4 gives the experimental results. In Section 5, we discuss the proposed method. Finally, Section 6 summarizes this study.

2. Related Work

The fragmentation of information in the era of big data gives unstructured CTI reports the characteristics of diversification, fragmentation, and heterogeneity. For these characteristics of unstructured CTI reports, Liao et al. [7] proposed an approach to automatically recover valuable attack indicators from popular technology blogs and convert them into industry-standard and machine-readable CTI reports. Sara Qamar et al. [8] proposed the construction of the Structured Threat Information eXpression (STIX) analyzer ontology and its ontology model relationship. Their method can determine the threat relevance, possibility, and affect and expose assets by automatically classifying network threats and formulated rules and inferences. Xun et al. [9] proposed an automatic identification model of threat intelligence (TI) based on a convolutional neural network (CNN) for automatically extracting TI from various unstructured TI data sources. These studies reduce the noise data in the CTI report by reorganizing unstructured threat report knowledge to identify cyber threat information in an effective manner.

It is one of the important research contents in CTI analysis that reconstruct CTI knowledge by using the graph mode. Shu et al. [10] used a graph model to organize multisource heterogeneous threat data, which formalize cyber threat intelligence computing into a new security paradigm. Ya et al. [11] proposed an attack entities recognition method to construct a CTI knowledge graph. Jia et al. [12] used existing machine learning technology to organize the knowledge of threat reports and construct a knowledge base of cybersecurity. Du et al. [13] proposed a knowledge graph for human-readable CTI recommendation from the perspective of the attack chain. The threat intelligence knowledge graph helps security practitioners understand cyber threats in a timely and rapid manner.

The current research on CTI reports mainly includes real-time perception, dynamic sharing, and effective application. Regarding the application of CTI, it contains structured and unstructured CTI reports. For structured CTI reports, Kim et al. [14] automatically generate rules without human intervention to mitigate new network security threats that have been discovered in real-time. In response to the lack of domain knowledge analysis under the existing structured CTI reports, Tappeiner et al. [15] proposed a domain recognizer based on a convolutional neural network to identify targeted domain of CTI and automatically generates specific CTI from social media data.

In order to solve the problem of overreliance on the analysis of security practitioners results in the inefficiency of CTI applications, Zhu et al. [16] proposed an end-to-end approach for automatic feature engineering, which identifies abstract behaviors that are associated with malware and map these behaviors to concrete features and generates a characteristic semantic network. Zhu et al. [17] proposed an approach to bridge measurement data with manual analysis and train a multiclass classifier to extract IOCs and further categorize them into different stages. Ayoade et al. [18] have leveraged natural language processing techniques to extract attacker's actions from threat report documents generated by different organizations and then automatically classify them into standardized tactics and techniques.

For threat actions extraction from unstructured CTI reports, Husari et al. [5] proposed a method named TTPDrill to extract actions based on semantic dependence and an ontology database, which is used to map actions to different attack patterns. However, TTPDrill will neglect part of threat actions in clause structure and parallel sentences. And it used ontology structure to identify threat actions, which will lose some undefined threat actions in the ontology structure.

Husari et al. [19] developed an approach named ActionMiner, which used NLP technology and based on information entropy and mutual information, to extract low-level cyber threat actions from publicly available CTI sources. However, ActionMiner has relied on syntactic analysis to extract low-level threat actions. It lacks a behavioral subject, and the information content is difficult to guarantee.

This study proposes a framework called EX-Action. It extracts actions based on the syntactic structure and rules mapping and identifies them by a multimodal learning algorithm. EX-Action identifies actions based on multiple features, which improves the accuracy of action recognition and covers actions in complex sentence structures.

3. Proposed Framework

In this study, we propose a framework called EX-Action. It contains four modules, which are data preprocessing, candidate threat actions extraction, action feature extraction, and action identification. The EX-Action architecture is shown in Figure 1. First, EX-Action preprocesses the obtained CTI report. Second, candidate threat actions are extracted by a rule-based method. And then, candidate action multimodal features are calculated. Finally, EX-Action identifies actions and generates selected actions by a weighted ensemble learning algorithm.

3.1. Data Preprocessing. In this module, EX-Action cleans the data of CTI reports by filtering invalid characters and sentences that do not contain threat actions. There are some cybersecurity terms in CTI reports. However, these cybersecurity terms are not recognized by NLP technology, such as file paths, IP addresses, and so on. EX-Action uses regular expressions to replace and save unrecognizable terms.

3.2. Candidate Threat Actions Extraction. In this module, EX-Action extracts candidate threat actions from preprocessed CTI reports by a rule-based method. A CTI report consists of n sentences, which can be expressed as $T = \{S_1, \ldots, S_i, \ldots, S_n\}$, and each sentence contains several action verbs, $S_n = \{V_1, \ldots, V_i, \ldots, V_N\}$. For each verb, EX-Action extracts many actions based on a rules-matching strategy, denoted as $A_i = \{\text{action}_1, \text{action}_2, \ldots, \text{action}_m\}$. The extracted candidate threat actions are in the format (subject, verb, and object), i.e., an action consists of three elements which are subject, verb, and object. EX-Action matches the parts of speech (POS) for the three elements that consist of the action. POS are used to match the elements in action. The three elements in action rule matching are given in Table 1. The column "POS" represents the POS of each component, and "POS-Symbols" represent the symbol of POS tagged.

In this module, sentences are tagged by the POS tool [20]. Take the verb that is identified in the results of POS tagging as the start of the sliding window. Then, the subject and object are, respectively, searched in the threat description sentence. The window size for searching the subject and object can affect its extraction performance. Some potential objects may have a long distance from the target verb, and therefore, a too small window size cannot get them. However, a too large window size may result in many mismatched VO pairs, which will affect the identification efficiency of EX-Action. EX-Action adopts different strategies to set the sliding window size for searching the subject and the object. For the subject, the sliding window size is set to the number of words before the verb in one sentence, and then, EX-Action matches all nouns and noun combinations in the window with the verb. For the object, a dynamic window mechanism is used to set the sliding window size. This mechanism adopts the number of words after the verb in one sentence as the sliding window size, and the sliding window stops sliding when encountering another verb. Figure 2 shows an example of the action extraction of EX-Action.

In the process of searching the subject and object, there are phenomena that a noun compound structure or pronouns act as the subject or object. To ensure the integrity of the extracted action information, the multinoun compound structure is tokenized as a noun and matched with a verb. The verb and object can retain the basic information content, but when the pronoun is acted as the object, a lot of information might be lost. Therefore, the pronoun as the subject is saved, and the pronoun as the object is discarded in this module.

3.3. Action Feature Extraction. In this module, EX-Action extracts five types of features for each action. The extraction framework of action's features is shown in Figure 3. It contains similarity measurement, probability computation, mutual information value measurement, semantic dependency measurement, and distance computation. The features contains 9 values, $\text{feature}_{\text{action-all}} = \{F_1, F_2, \ldots, F_9\}$. The description of features is given in Table 2. More details of action feature extraction will be described next.

3.3.1. Similarity Measurement. In this subsection, the similarity between candidate actions and a CTI report p is calculated by the TF-IDF and BM25 algorithms. The TF-IDF method is commonly used to calculate the feature item weight in the process of text vectorization [21]. Equation (1) is used to calculate the weight of a feature item of the action.

$$\text{TF} - \text{IDF}(x) = \frac{N}{N(x)} * \left(\log\left(\frac{N+1}{N(x)+1}\right) + 1\right), \quad (1)$$

where N is the total number of words in the CTI report p, and $N(x)$ represents the number of words x in the CTI report p. Since threat actions contain different numbers of words, the average value is used as the similarity measure of candidate actions.

The BM25 [22] is an upgraded algorithm of TF-IDF. It adds a constant to TF-IDF to limit the growth limit of the TF value and uses the document length to evaluate the

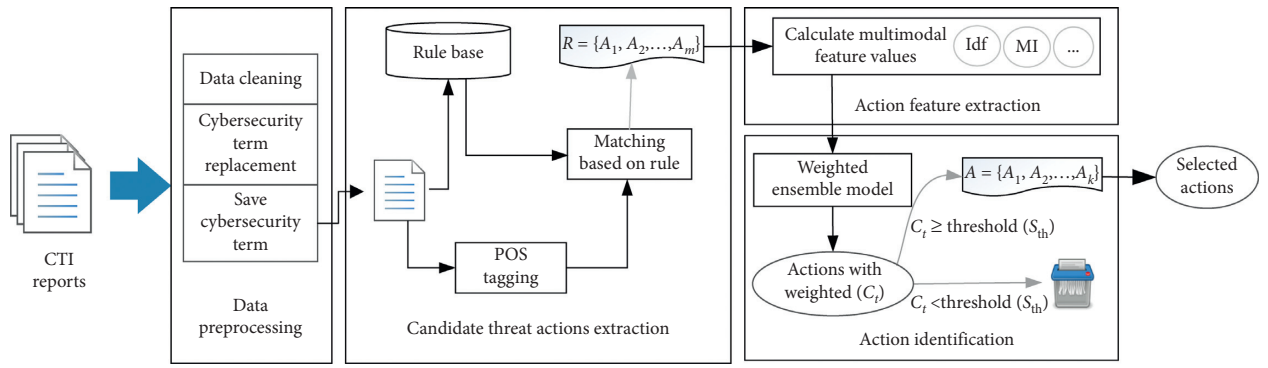

Figure 1: The architecture of EX-Action.

Table 1: The three elements in action rule matching table.

Element in action	POS	POS-symbols
Subject	Noun, noun and noun, pronoun, noun + cardinal number	NN, NNS, NNP, NNPS, PRP, CC, N + CD
Verb	Verb, verb and verb	VB, VBD, VBG, VBP, VBN, VBZ
Object	Noun, noun and noun, noun + cardinal number	NN, NNS, NNP, NNPS, CC, N + CD

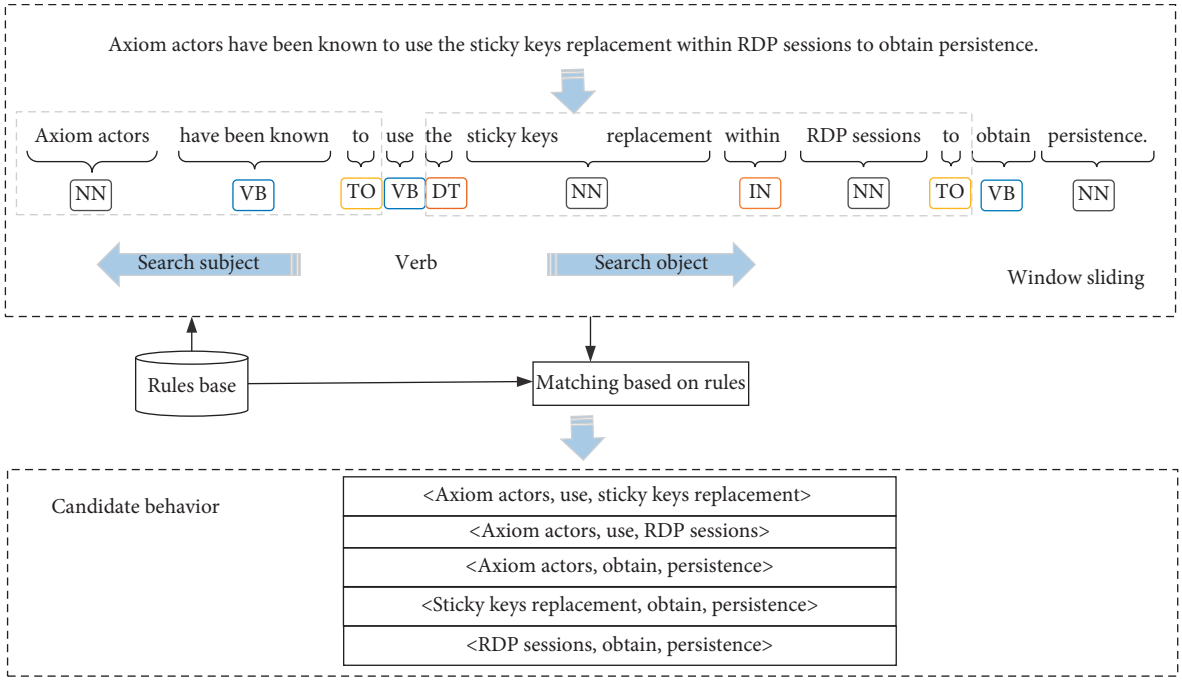

Figure 2: An example of threat action extraction.

importance of candidate actions. It performs a weighted summation on the correlation scores between candidate threat actions and the CTI report p, and equation (2) is used to calculate the BM25 of the action.

$$\text{Simlarity} = \text{idf} * \frac{((k_1 + 1) * \text{tf})}{(k_2 * (1 - b + b * L) + \text{tf})}, \quad (2)$$

where tf represents the frequency of each word, idf represents the inverse word frequency of each word, L is the length of the text, and k_1, k_2, and b are the adjustment factors.

3.3.2. Probability Computation. In this subsection, the co-occurrence frequency of the VO pair is calculated to determine the correlation between the candidate action and the CTI report. In action, the subject usually indicates the attacking subject or organization, the verb represents the attack action, and the object represents the operation target in the CTI report. Since attack organizations are different in the attack process, calculating the co-occurrence frequency of SVO triples will weaken the relevance between the action and the CTI report. Therefore, EX-Action calculates the co-occurrence frequency of the VO pair under a fixed window is taken as a feature. The correlation between the action and the

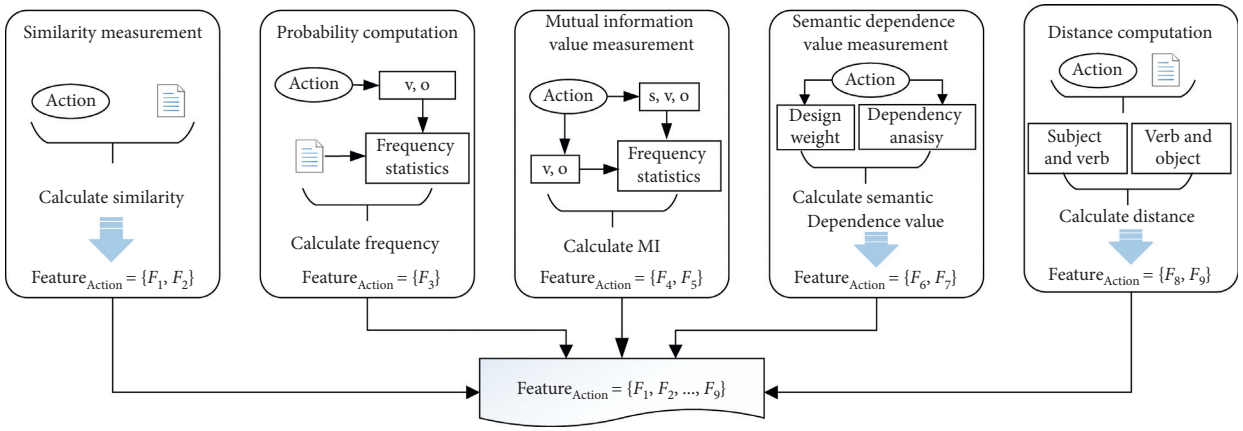

Figure 3: The extraction framework of action feature.

Table 2: The description of five type features.

Type	Feature	Description
Similarity	TF − IDF (F_1), BM25 (F_2)	The similarity between the action and description sentences.
Probability	P_vo (F_3), frequency (F_4)	The probability and frequency of the VO pair.
Mutual information	MI_VO (F_5), MI_SVO (F_6)	The MI contains two values, which, respectively, are the VO pair-MI (MI_VO) and the SVO triples-MI (MI_SVO).
Semantic dependence	Dependence (F_7)	The semantic dependency of each word in action.
Distance	Distance_SV (F_8), distance_VO (F_9)	It contains two values, which, respectively, express the distance between the verb and the subject (distance_SV) and the distance between the verb and the object (distance_VO).

CTI report is proportional to the frequency of the VO pair. Specifically, window size $m = 25$ is used to calculate the co-occurrence frequency of the VO pair in our experiment.

3.3.3. Mutual Information Value Measurement. Mutual information (MI) measures the reduction in uncertainty of information about one random variable, given knowledge of another [23]. The MI of VO pair and SVO triple are calculated to measure the information content of candidate actions. The correlation between candidate actions and CTI report is proportional to the MI value. Equation (3) is used to calculate the MI of SVO triple.

$$\mathrm{MI} = \sum_{s \in S} \sum_{vo \in VO} \log \frac{p(s, \mathrm{vo})}{p(s) * p(\mathrm{vo})}, \tag{3}$$

where $p(s, \mathrm{vo})$ is the frequency of a threat action, $p(s)$ is the number of occurrences of its subject, and $p(\mathrm{vo})$ is the co-occurrence frequency of VO pair.

3.3.4. Semantic Dependence Measurement. There are some candidate actions with high matching degree, but they in fact are inaccurate semantic matching. The feature of semantic dependency is designed to recognizing these actions. The Stanford dependency analyzer [4] is used to analyze the relation of semantic dependency for each sentence. W_1 and W_2 are set as the dependency weight between the subject and the verb and the dependency weight between the verb and the object, respectively. And then, summing the dependency weight (W_1 and W_2) as the feature of semantic dependency, Figure 4 shows an example of Stanford semantic dependency for a sentence.

3.3.5. Distance Computation. In this subsection, two distances are computed. They are, respectively, the distance between subject and verb and the distance between verb and object. The number of word between the verb and the target word is taken as the value of distance. For example, for a word between the subject and the verb, the distance is recorded as 1.

3.4. Action Identification. Ensemble learning promote weak learners to strong learners by constructing and combining multiple base learners to complete the learning task. In this module, the EX-Action automatically identifies candidate actions by a parallel ensemble learning algorithm. The main process is illustrated in Algorithm 1. The time complexity of Algorithm 1 is $O(n^2)$.

In Algorithm 1, the training set D is the input, which contains candidate actions and their features. The ground truth A is the action set of manual extraction. The ground truth A is used to calculate the similarity of candidate actions. Five-base classification learners are used to construct a parallel ensemble classification. They are, respectively, decision tree (tree), random forest (forest), support vector machine (SVM), linear regression (LR), and multilayer perceptron classifier (MLPC). It can be expressed as $T = \{T1, T2, \ldots, T5\}$.

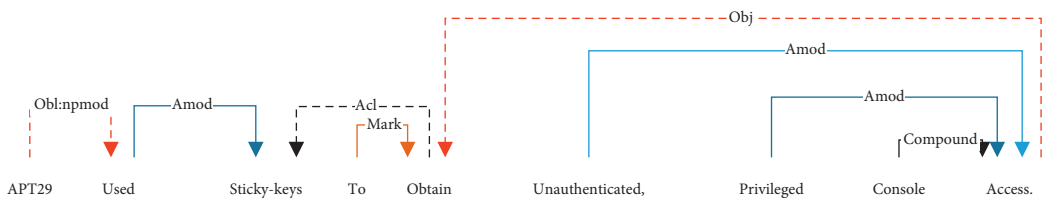

Figure 4: An example of Stanford semantic dependency for a sentence.

Require:
Input:
Training Set: $D = \{(\text{Action}_1^M, \text{Feature}_1), (\text{Action}_2^M, \text{Feature}_2), \ldots, (\text{Action}_m^M, \text{Feature}_m)\}$; //$\text{Action}^M$ is the candidate action of machine identification, Feature_n is the $n-th$ action feature value, which contains 9 values
$A = \{\text{Action}_1^H, \text{Action}_2^H, \ldots, \text{Action}_k^H\}$; //$\text{Action}^H$ is the action of manual extraction
Base classification learners: $T = \{T_1, T_2, \ldots, T_5\}$
Train:
for each $t \in [0, m]$**do**
  for each $i \in [0, 5]$**do**
    $\text{result}_i = T_i(\text{Action}_t^M, \text{Feature}_t)$//$\text{result}_i$ is the predicted value of action in the $i-th$ Base classification Learners
  end for
  $C_t = \sum_{i-1}^{m} \text{result}_i * \omega_i // \omega_i$ is the weighted of action in the $i-th$ Base classification Learners
  if $C_t > S_{th} // C_t$ is the $t-th$ action weight, S_{th} is the optimal voting threshold **then**
    Threat action$\leftarrow \text{Action}_t^M$//Threat action is the selected action set
    $S_t = f(\text{Action}_t^M, \text{Action}_t^H) // S_t$ is the similarity of the $t-th$ action between the ground truth and machine identification
  end if
  if $S_t > \theta // \theta$ is the optimal similarity threshold **then**
    Action_t^M is correct action
  end if
end for
Output: Threat Action $= \{\text{Action}_1^M, \text{Action}_2^M, \ldots, \text{Action}_k^M\}$

Algorithm 1: Threat actions identification.

The result_i is the predicted value for the action generated by the i-$^{\text{th}}$ base classification learner. Then, different weight ω_i is set for the result_i and sum them to gain the weighted C_t of each action. EX-Action identifies selected actions from candidate threat actions set by the weighted voting method and minimizes the loss function through the linear combination of base learners. S_{th} is a predefined voting threshold; if C_t is larger than S_{th}, the candidate action will be regarded as the selected action. Finally, EX-Action calculates the similarity S_t between the selected action and the ground truth. If the similarity S_t is larger than the predefined similarity threshold θ, the action Action_t^M is recognized as the correct action.

In EX-Action, according to the different classification performances of the different models, different weight values are set for each model. It can be seen that the decision tree behaves the best performance in our experiment. Therefore, the decision tree is assigned to the maximum weight, and the weight values of the other four models are all equal.

4. Evaluation

4.1. Experimental Dataset. We obtained 243 security reports from ATT&CK[1]. They contain 5136 sentences. The number of sentences regarding different techniques in CTI reports is given in Table 3. In our experiment, 20% of the CTI reports are randomly chosen as the test data. Figure 5(a) shows the distribution of sentence lengths, and Figure 5(b) shows frequency distribution of the test data.

It can be seen from Figure 5 that the length of the sentence that describing threat action is mainly distributed between 10 and 30. Therefore, the dataset in this study can be regarded as the CTI report with complex sentence and long length.

4.2. Evaluation Metrics. In this study, accuracy, recall, precision, F1-score, normalized mutual information (NMI), and number of extracted actions (number) are used as performance metrics. The accuracy, recall, precision, and F1-score reflect the quantitative difference of threat actions between machine identification and the ground truth. They can be calculated by equations (4)–(7).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{6}$$

Table 3: The number of sentences regarding to different techniques in CTI reports.

Techniques	CTI	Sentences
Initial access	11	166
Execution	35	737
Persistence	38	376
Privilege escalation	17	165
Defense evasion	36	780
Credential access	16	264
Discovery	22	1048
Lateral movement	14	308
Collection	13	386
Command and control	20	654
Exfiltration	6	58
Impact	15	194
Total	243	5136

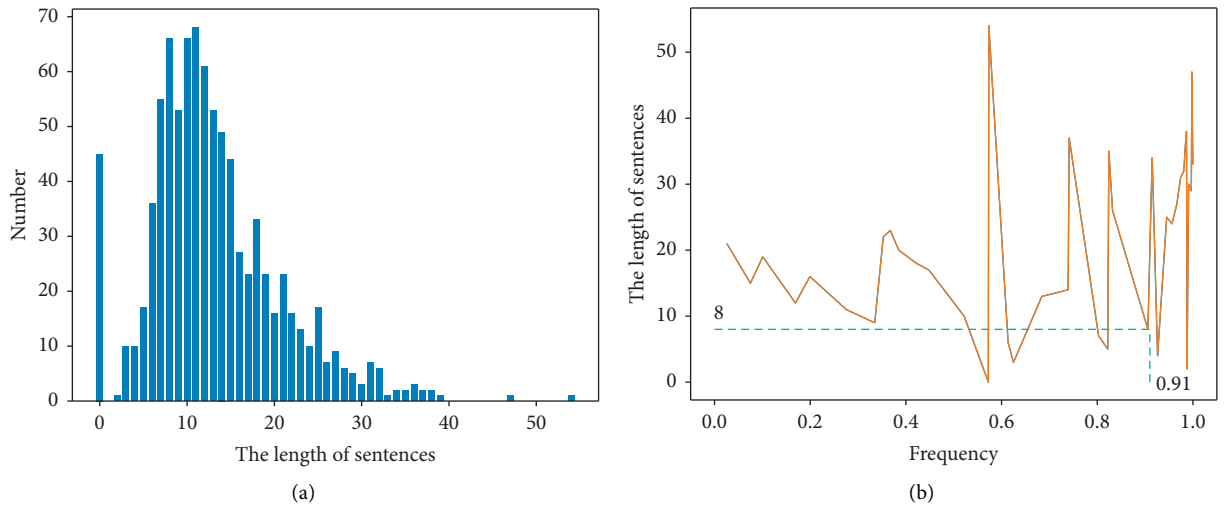

Figure 5: The number of sentence lengths and frequency distribution graphs of the test data. (a) The number of sentence lengths distribution. (b) The frequency distribution graphs of the test data.

$$\text{F1} - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{Precision} + \text{recall}}. \tag{7}$$

The number represents the number of extracted actions, and NMI represents a measure of the difference in threat actions information content between machine recognition and manual extraction. NMI is often used in clustering to measure the similarity of two clustering results. And it is used to measure the difference in the information content in this study. The NMI reflects the similarity of actions between machine recognition and manual extraction. Equation (8) is used to calculate the information content difference of each action.

$$\text{NMI} = \frac{-2\sum_{i=1}^{C_m}\sum_{j=1}^{C_k} C_{ij} * \log C_{ij} * N/C_i * C_j}{\sum_{i=1}^{C_m} C_i * \log C_i/N + \sum_{j=1}^{C_k} C_j * \log C_j/N}, \tag{8}$$

where N represents the number of word nodes of the action, C_m represents the number of word nodes of the machine recognition action, C_h represents the number of word nodes of the manual extraction action, and C_{ij} represents the number of word nodes belonging to both types of actions. The similarity of information between the machine recognition and the ground truth is proportional to the MI value. When the NMI is equal to 1, the information content is equal.

4.3. Results and Analysis. In this section, four subsections are there to show our experimental results and analysis. They are the feature importance ranking of EX-Action, model comparison, threshold determination, and the effect comparison of existing methods. Note that the best values of each metric are bold in each table.

4.3.1. Feature Importance Ranking of EX-Action. This subsection shows the feature distribution of actions, the importance distribution of features, and the performance of different features combination. The features contain 9 values. They are $\text{TF} - \text{IDF}(F_1)$, $\text{BM25}(F_2)$, $P_{\text{vo}}(F_3)$, $\text{frequency}(F_4)$, $\text{MI}_{\text{VO}}(F_5)$, $\text{MI}_{\text{SVO}}(F_6)$, $\text{dependence}(F_7)$, $\text{distance}_{\text{SV}}(F_8)$, and $\text{distance}_{\text{VO}}(F_9)$. The feature distributions of some actions are shown in Figure 6. It can be seen that the feature value distributions of these actions are nonlinear distributions.

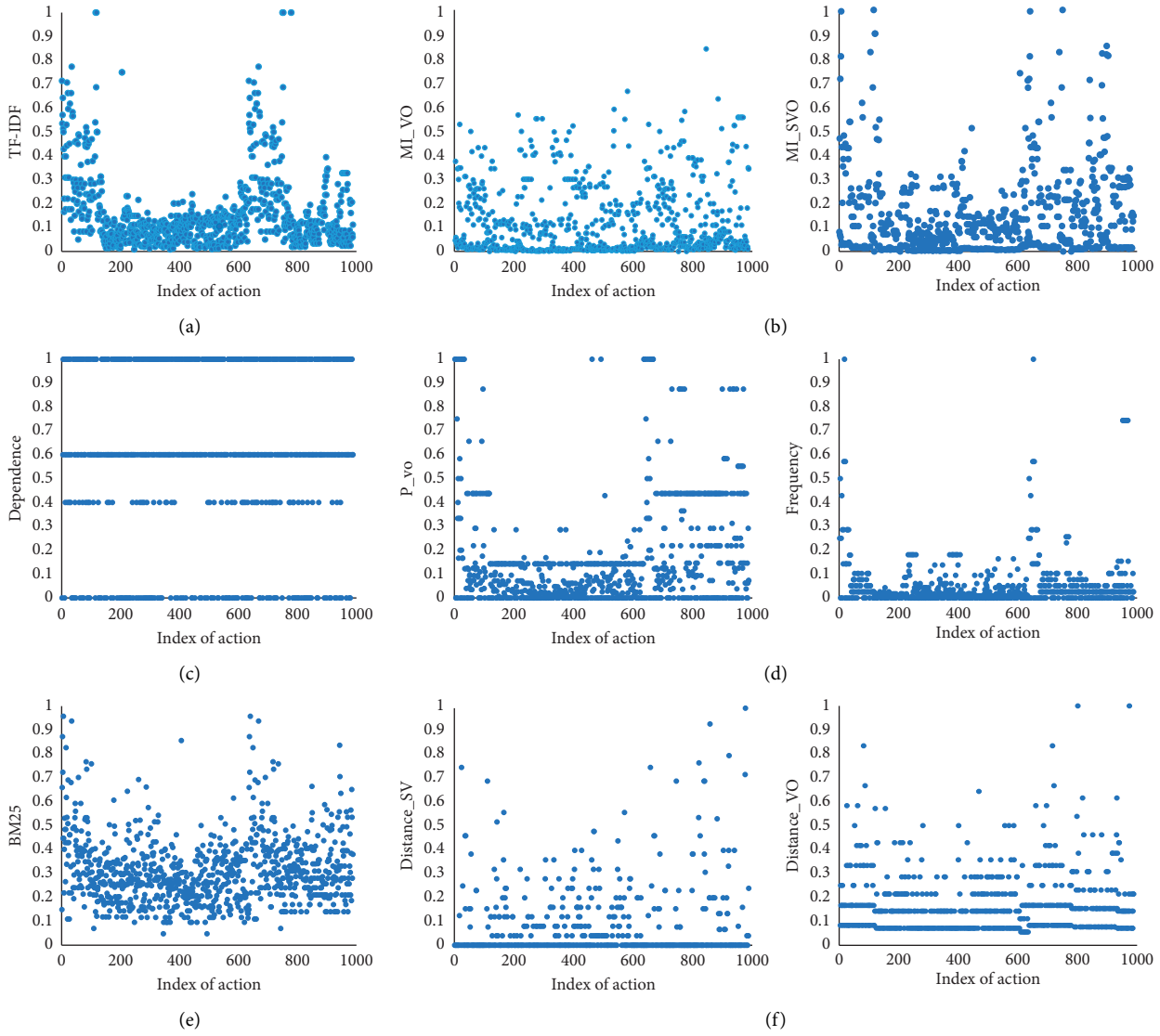

Figure 6: The different feature values distribution of threat actions. (a) Threat actions distribution of feature TF-IDF. (b) Threat actions distribution of feature MI. (c) Threat actions distribution of feature semantic dependence. (d) Threat actions distribution of feature VO pair frequency. (e) Threat actions distribution of feature BM25. (f) Threat actions distribution of feature window length.

Table 4 gives the obtained result of different features combinations. Under the same conditions, the recall of combination$_1$ and combination$_2$ reached the maximum value of 77.82%, and the number of extracted actions reached the maximum value of 1179, but other metrics are lower than combination$_7$. The performance of combination$_7$ is higher than others combinations in terms of accuracy, precision, F1-score, and information completeness. It can be found that combination$_7$ is more appropriate for the feature selection of threat action identification.

The importance distribution of the 9 features is calculated by the Gini index, as given in Figure 7. Figure 7 provides data that the distance of VO pairs has the largest effect on actions recognition. The frequency and the conditional probability of VO pairs are less important than other features.

4.3.2. Model Comparison. This subsection shows the results of the different base learners, unweighted ensemble learning model (unweighted model), and EX-Action (weighted ensemble model). The results obtained by the different base learners, unweighted model, and EX-Action are given in Table 5.

As given in Table 5, the accuracy and F1-score of tree are higher than other base learners, but its accuracy and F1-score are lower than EX-Action. Therefore, in EX-Action, the weight of the tree is the largest, and the weight values weights of other base learners are the same. Comparing the results of the unweighted model and EX-Action, the recall of the unweighted model is 81.06%, which is higher than EX-Action, and the number of extracted actions is also higher than EX-Action. However, the accuracy, precision, F1-score, and NMI values of EX-Action are higher than those of the unweighted model.

Table 4: Comparison of the result of different features.

Combination	Features	Number	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	NMI (%)
Combination_1	$(F_1, F_2, F_5, F_6, F_7, F_8, F_9)$	1179	71.99	**77.82**	82.16	79.93	94.08
Combination_2	$(F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9)$	1179	71.29	**77.82**	80.98	79.36	94.02
Combination_3	$(F_1, F_3, F_4, F_5, F_6, F_7, F_8, F_9)$	1177	72.20	77.68	82.65	80.09	93.81
Combination_4	$(F_1, F_2, F_3, F_4, F_5, F_6, F_8, F_9)$	1169	71.28	77.16	81.63	79.33	93.98
Combination_5	$(F_1, F_2, F_3, F_4, F_5, F_6, F_7)$	1046	64.88	69.04	78.71	73.56	93.09
Combination_6	$(F_1, F_2, F_3, F_4, F_7, F_8, F_9)$	1131	69.82	74.65	81.72	78.03	93.76
Combination_7	$(F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9)$	1167	**72.35**	77.03	**83.60**	**80.18**	**94.26**

The bold values represent the maximum values of each metric.

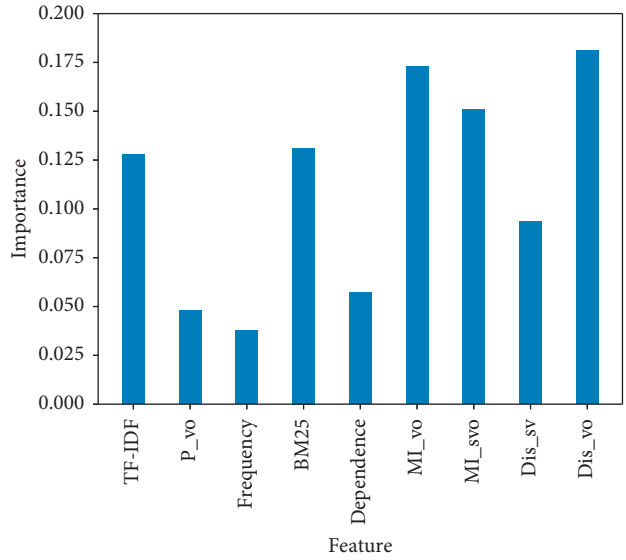

Figure 7: The importance distribution of features.

Table 5: Comparison of the result of different models.

Model	Number	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	NMI (%)
SVM	726	52.25	47.92	79.43	59.78	93.14
Tree	1135	71.64	74.92	84.70	79.50	93.85
LR	736	50.90	48.58	73.75	58.58	93.33
MLPC	716	50.43	47.26	74.74	57.91	93.16
Forest	612	49.38	40.40	**88.95**	55.56	93.64
Unweighted model	1228	67.66	**81.06**	72.58	76.58	93.61
EX-Action	1167	**72.35**	77.03	83.60	**80.18**	**94.26**

The bold values represent the maximum values of each metric.

4.3.3. Threshold Determination. The voting threshold determines the result of the model recognition actions, and the similarity threshold determines the correctness of actions recognization, which will influence the performance of EX-Action. This subsection tests the optimal parameters for EX-Action through the setting of the voting threshold and similarity threshold. The comparison of results under different voting thresholds and different similarity thresholds is shown in Figure 8.

As shown in Figure 8(a), when the similarity threshold and voting threshold are set to 0.2 and 4, respectively, accuracy, F1-score, and NMI are optimal. Besides, as shown in Figure 8(b), the similarity threshold is the degree of difference between machine recognition action and ground truth. It can be seen that the higher the similarity threshold, the higher the information content it contains and the lower the accuracy and F1-score will be.

4.3.4. Performance Comparison with the Existing Approaches. In this subsection, the performance of EX-Action is compared with TTPDrill [5] and ActionMiner [19] in terms of accuracy, recall, precision, F1-score, the number of extracted actions, and NMI. As shown in Table 6, the result of EX-Action is higher than the other two methods in terms of accuracy, recall, precision, F1-score, the number of extracted actions, and NMI.

For extracting actions from the CTI reports with complex structures, TTPDrill mainly relies on semantic

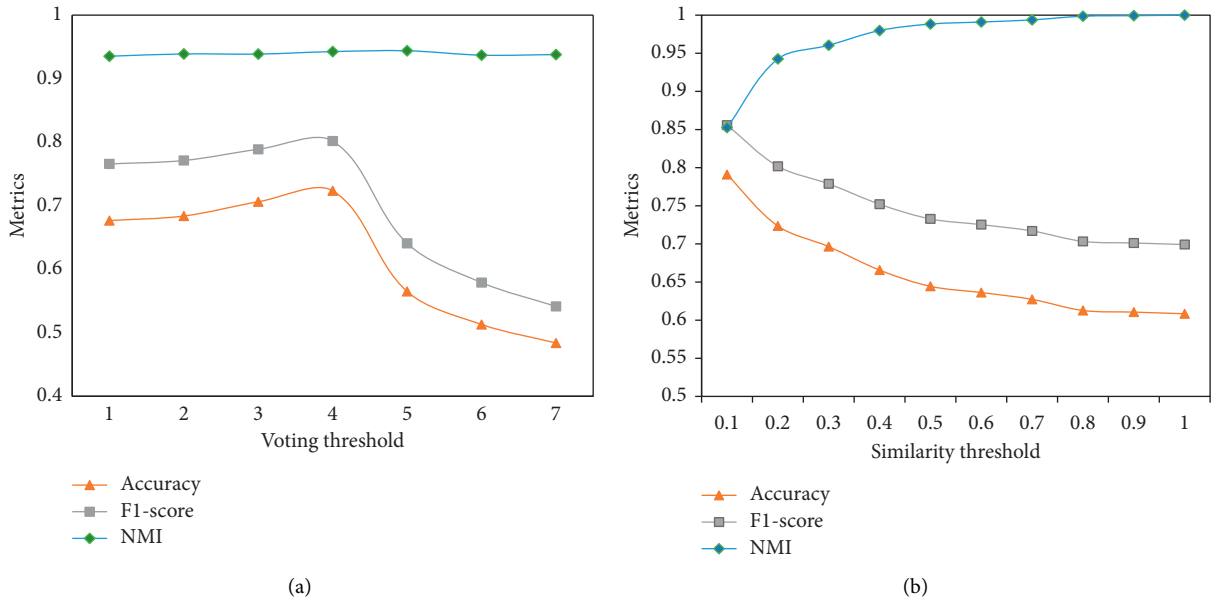

Figure 8: The effect of thresholds. (a) The effect of different voting thresholds. (b) The effect of different similarity thresholds.

Table 6: Performance comparison of the three methods.

Model	Number	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	NMI (%)
TTPDrill	711	54.00	46.93	88.32	61.29	72.86
ActionMiner	1237	75.02	81.65	83.30	82.47	60.90
EX-Action	1246	**79.09**	**82.24**	**89.19**	**85.58**	**85.26**

The bold values represent the maximum values of each metric.

dependence. It will ignore part of threat actions in the complex sentence structure like clauses. Therefore, TTPDrill extracted fewer actions and behaved in poor performance. TTPDrill can retain the main information of the action compared with the ActionMiner, so the NMI is higher than ActionMiner. ActionMiner mainly relies on syntactic structure extraction. It can obtain better accuracy and recalls for low-level actions extraction for complex sentences. However, it does not retain the subject of the action, so its NMI value is low.

Besides, we compare the examples of actions extracted from CTI reports used in our experimental dataset and the literature [19] that proposed ActionMiner, respectively. The examples of the extracted actions obtained by the three methods in the two datasets are given in Table 7.

The actions extracted by the three methods on our experimental dataset are shown on the left of Table 7. It can be seen that TTPDrill has a better effect for extracting sentences with simple structure and obvious dependencies. Therefore, its performance is poor in our experimental dataset. ActionMiner can extract more actions than TTPDrill, but it lacks the subject of the action and behaves in poor performance in retaining the information content of the sentence. EX-Action can achieve better results in the number of extracted actions and the retention of the information of the sentences with complex structures.

For the CTI report mentioned in the literature [19], the actions extracted by the three methods are shown on the right of Table 7. It can be seen that the threat description sentence structure in those CTI reports is relatively simple. Comparing the three methods, it is found that actions extracted by TTPDrill can give a good description, but the composite components of the sentences are still not extracted. ActionMiner can extract more actions, but it lacks the subject. The actions extracted by EX-Action are more complete than that of ActionMiner, and the number of extracted actions extracted by EX-Action is more than TTPDrill.

5. Discussion

The unstructured CTI report records the network attack process, context mechanism, and other information. Accurately extracting and identifying threat actions from unstructured CTI reports will help security practitioners efficiently restore the attack process. In [24], Gao et al. correlated the threat action extracted from the CTI text with the action extracted from the system audit log and constructed a threat action graph to realize an efficient network threat search.

5.1. Contributions. First, EX-Action can be used to extract actions from unstructured CTI reports with complex sentences. It uses syntactic rules to extract threat actions, which can extract more actions in complex sentences. At the same time, machine learning algorithms are used to identify

Table 7: Threat actions extracted from CTI report in different datasets.

	The CTI report in our dataset	The CTI report mentioned in paper named ActionMiner
Threat description	APT29 used sticky keys to obtain unauthenticated, privileged console access. APT3 replaces the sticky keys binary executable file for persistence. Axiom actors have been known to use the sticky keys replacement within RDP sessions to obtain persistence. Deep Panda has used the sticky keys technique to bypass the RDP login screen on remote systems during intrusions. Empire can leverage WMI debugging to remotely replace binaries like executable file, executable file, and executable file with executable file.	It creates the following file: caches_version.db. . .. The Trojan creates the following registry entries. . .. Next, the Trojan steals the following information from the compromised computer: keystrokes, clipboard data, screenshot based on specified keywords in the window title, network adapter information such as MAC address, IP address, adapter name, adapter, and description. The Trojan then saves the stolen information in the following location: caches_version.db.
TTPDrill	APT29 used keys, Deep Panda used technique	Creates file: caches_version.db, Trojan creates registry entries, Trojan steals information from computer, Trojan saves information in location: caches_version.db. . .
ActionMiner	Used sticky keys, obtain console access, use sticky keys, use replacement, obtain persistence, has used technique, bypass RDP login, bypass screen, replace binaries	Creates file, creates registry entries, steals information, steals keystrokes, steals clipboard data, steals screenshot, steals network adapter information, steals MAC address, steals IP address, steals adapter name, steals adapter description, saves information
EX-Action	APT29 used sticky keys, APT29 obtain console access, axiom actors use sticky keys replacement, axiom actors obtain persistence, Deep Panda has used technique, Deep Panda bypass RDP login screen, empire replace binaries	It creates file caches_version, Trojan creates registry entries, Trojan steal information, Trojan steals computer keystrokes, clipboard data, screenshot, Trojan steals MAC address, IP address, adapter name, and adapter description, Trojan saves information, Trojan save location: caches_version

actions based on their own features, which can identify more actions undefined in the ontology model. Second, EX-Action extracted the action that contains the subject, verb, and object. It also provides a method to extract entity relations, which have contextual semantic relations between entities.

5.2. Limitations. There are some shortcomings in this study, such as overreliance on part-of-speech and semantic analysis, which may lose part of threat actions and failed to recognize the pronoun referent.

6. Conclusion

This study proposes a multimodal learning approach to extract and identify threat actions, and it can extract the threat action in complex semantic relationships and recognition of the cybersecurity entity associations with undefined relationships. The experimental result shows that EX-Action can have a certain balance between accuracy and information completeness in the action extraction. In future work, we will research how to avoid overreliance on part-of-speech tagging tools and try to use pronoun resolution to identify the subject and object of the pronoun.

Data Availability

The unstructured CTI reports data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (62062022) and the Science and Technology Foundation of Guizhou Province ([2017]1051).

References

[1] J. H. Li, "Overview of the technologies of threat intelligence sensing, sharing and analysis in cyber space," *Chinese Journal of Netword and Information Security*, vol. 2, no. 2, p. 16, 2016.

[2] S. Samtani, M. Abate, V. Benjamin, and W. Li, "Cybersecurity as an industry: a cyber threat intelligence perspective," *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, vol. 2020, pp. 135–154, 2020.

[3] Y. Lin, P. Liu, H. Wang, W. J. Wang, and Y. Q. Zhang, "Overview of threat intelligence sharing and exchange in cybersecurity," *Journal of Computer Research and Development*, vol. 57, no. 10, p. 2052, 2020.

[4] D. Marneffe, M. Catherine, and C. D. Manning, "The Stanford typed dependencies representation," in *Proceedings of the Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pp. 1–8, Manchester, UK, August 2008.

[5] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, X. Niu, and " 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, Orlando, FL, USA, December 2017.

[6] S. Lee, Y. T. Park, and B. J. d'Auriol, "A novel feature selection method based on normalized mutual information," *Applied Intelligence*, vol. 37, no. 1, pp. 100–120, 2012.

[7] X. J. Liao, K. Yuan, X. F. Wang, Z. Li, L. Y. Xing, and R. Beyah, "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, Vienna, Austria, October 2016.

[8] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, and B.-T. Chu, "Data-driven analytics for cyber-threat intelligence and information sharing," *Computers & Security*, vol. 67, pp. 35–58, 2017.

[9] S. Xun, X. Y. Li, and Y. L. Gao, "AITI: An automatic identification model of threat intelligence based on convolutional neural network," in *Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence*, pp. 20–24, Xiamen China, May 2020.

[10] X. K. Shu, F. Araujo, D. L. Schales et al., "Threat intelligence computing," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1883–1898, Toronto, ON, Canada, October 2018.

[11] Y. Qin, G.-W. Shen, W.-B. Zhao, Y.-P. Chen, M. Yu, and X. Jin, "A network security entity recognition method based on feature template and CNN-BiLSTM-CRF," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 6, pp. 872–884, 2019.

[12] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering*, vol. 4, no. 1, pp. 53–60, 2018.

[13] M. Du, J. Jiang, Z. Jiang, Z. Lu, and X. Du, "PRTIRG: a knowledge graph for people-readable threat intelligence recommendation, Knowledge Science, Engineering and Management," in *Proceedings of the International Conference on Knowledge Science, Engineering and Management*, pp. 47–59, Springer, Athens, Greece, August 2019.

[14] E. Kim, K. Kim, D. Shin, B. Jin, and H. Kim, "CYTIME: Cyber Threat Intelligence ManagEment framework for automatically generating security rules," in *Proceedings of the 13th International Conference on Future Internet Technologies*, pp. 1–5, Seoul, Republic of Korea, June 2018.

[15] E. Tappeiner, F. Finotello, P. Charoentong et al., "TIminer: NGS data mining pipeline for cancer immunology and immunotherapy," *Bioinformatics*, vol. 33, no. 19, pp. 3140-3141, 2017.

[16] Z. Y. Zhu and T. Dumitras, "Featuresmith: automatically engineering features for malware detection by mining the security literature," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 767–778, Vienna, Austria, October 2016.

[17] Z. Zhu and T. Dumitras, "Chainsmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports," in *Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 458–472, IEEE, London, UK, April 2018.

[18] G. Ayoade, S. Chandra, L. Khan, K. Hamlen, and B. Thuraisingham, "Automated threat report classification over multi-source data," in *Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pp. 236–245, IEEE, Philadelphia, PA, USA, October 2018.

[19] G. Husari, X. Niu, B. Chu, and E. Al-Shaer, "Using entropy and mutual information to extract threat actions from cyber threat intelligence," in *Proceedings of the 2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 1–6, IEEE, Vancouver, BC, Canada, May 2018.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, Berlin, Germany, August 2014.

[21] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the First Instructional Conference on Machine Learning*, vol. 242, pp. 29–48, Citeseer, Washington, DC, USA, August 2003.

[22] M. Wijewickrema, V. Petras, and N. Dias, "Selecting a text similarity measure for a content-based recommender system," *The Electronic Library*, vol. 37, 2019.

[23] P. Latham and Y. Roudi, "Mutual information," *Scholarpedia*, vol. 4, no. 1, p. 1658, 2009.

[24] P. Gao, F. Shao, X. Y. Liu et al., "Enabling efficient cyber threat hunting with cyber threat intelligence," 2020, https://arxiv.org/abs/2010.13637.