future internet

Article

An Automatic Generation Approach of the Cyber Threat Intelligence Records Based on Multi-Source Information Fusion

Tianfang Sun, Pin Yang *, Mengming Li and Shan Liao

College of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China;
suntianfang@stu.scu.edu.cn (T.S.); jack2017@stu.scu.edu.cn (M.L.); liaoe1110@163.com (S.L.)
* Correspondence: yangpin@scu.edu.cn

Abstract: With the progressive deterioration of cyber threats, collecting cyber threat intelligence (CTI) from open-source threat intelligence publishing platforms (OSTIPs) can help information security personnel grasp public opinions with specific pertinence, handle emergency events, and even confront the advanced persistent threats. However, due to the explosive growth of information shared on multi-type OSTIPs, manually collecting the CTI has had low efficiency. Articles published on the OSTIPs are unstructured, leading to an imperative challenge to automatically gather CTI records only through natural language processing (NLP) methods. To remedy these limitations, this paper proposes an automatic approach to generate the CTI records based on multi-type OSTIPs (GCO), combing the NLP method, machine learning method, and cybersecurity threat intelligence knowledge. The experiment results demonstrate that the proposed GCO outperformed some state-of-the-art approaches on article classification and cybersecurity intelligence details (CSIs) extraction, with accuracy, precision, and recall all over 93%; finally, the generated records in the Neo4j-based CTI database can help reveal malicious threat groups.

Keywords: cyber threat intelligence; open-source threat intelligence platform; nature language processing; machine learning; information extraction; text analytics

check for updates

Citation: Sun, T.; Yang, P.; Li, M.; Liao, S. An Automatic Generation Approach of the Cyber Threat Intelligence Records Based on Multi-Source Information Fusion. *Future Internet* **2021**, *13*, 40. https://doi.org/10.3390/fi13020040

Academic Editor: Georgios Kambourakis

Received: 13 December 2020
Accepted: 29 January 2021
Published: 2 February 2021

1. Introduction

To promote threat intelligence sharing more efficiently, in 2013, Gartner proposed the idea of cyber threat intelligence (CTI) which is defined as "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" [1]. According to its concept, CTI contains a variety of detailed information about current or upcoming cybersecurity threats, which can help organizations to implement a proactive cyber defense against cybersecurity threats. However, the challenge is the low efficiency of gathering CTI records, which restricts the wide application of CTI technology and entails significant burdens for timely analyzing a large amount of data. At present, only a few CTI sources are providing machine-readable CTI records. Most of the CTI sources belong to commercial security companies. Security staff and organizations must pay high prices to access valid CTI information. Although the communities like CleanMX [2] and PhishTanks [3] have provided a large amount of threat information on their platforms, their CTI records only contain little information regarding threats, like malicious URL and IP, which cannot be applied for network security situation awareness system or other defense mechanisms. A direct solution to solve this problem is to generate CTI records with enough information from open-source web articles or reports.

Meanwhile, to adapt to fast-evolving cyber threats, security staff are actively exchanging security event-related details in unstructured contexts through open-source threat intelligence publishing platforms (OSTIPs). These OSTIPs are cybersecurity websites established for publishing newly found security-related research results, and the new findings can be converted into machine-readable CTI records for automatic analysis and quick deployment in various defense mechanisms. However, due to the explosive growth of information shared on the multi-type OSTIPs, manually collecting the CTI has been of low-efficiency, which cannot help security staff deal with the frequently occurring cybersecurity events timely. Thus, automatically generating CTI records from articles published by OSTIPs can meet the demand for machine-readable detailed CTI. To automatically generate CTI records, several challenges remain to be addressed.

With its rapid development, OSTIP has evolved into different types according to their maintainers. Although there are some existing methods in generating CTI records from open-source articles published by OSTIPs, they are only focused on the company type of OSTIPs. Liao et al. [4] proposed iACE to generate the OpenIoC records from articles published by the company type of OSTIPs like FireEyeBlog [5], while Husari et al. [6] proposed TTPDrill to generate the structured threat information expression (STIX) 2.0 records from reports published by cybersecurity companies. None of the existing methods are focused on the other type of OSTIPs—like OSTIPs maintained by individuals, teams, or communities—which also contain large quantities of CTI. Thus, we designed an automatic generation approach of the CTI records based on multi-type OSTIPs (GCO).

To generate CTI records from multi-type OSTIPs, the first challenge is to classify articles into 'threat-article' and 'non-threat-article'. Threat-articles are articles written to share the details found in cybersecurity events, while non-threat-articles are not relevant to the current happening events. Most of the non-threat-articles are advertisements, tips on cyber technology, or news about the cyber industry. There are several feature combinations proposed for training article classification models based on the support vector machine (SVM) algorithm [4,6]. However, these feature combinations are only tested on articles published by company OSTIPs, which are written in a formal style. Thus, we designed an experiment on the often-used article classification features and classification algorithms to build a well-performance article classification model.

The second challenge in generating CTI records from articles published by multi-type OSTIPs is the proper method to extract threat-related details from unstructured contexts. These details, like indicators of compromise (IoCs) and vulnerability records, contain information the authors are willing to share, and we call such details as cybersecurity intelligence details (CSIs). To extract CSIs in context representation, information selection [7] is necessary. According to our observation, the CSIs in threat-articles can be divided into sentence format and list format. Recent studies in the natural language processing (NLP) field propose an effective method to extract sentence format CSIs by using an event template with name entity recognition [4,8]. However, these methods only work on the sentence format CSIs, and the CSIs in list format are ignored for lacking technique on validating the list format CSI-candidates just according to their belonging sentences. Moreover, the existing methods have not taken the article embedded HTML document structure into consideration. Thus, we designed an HTML-based key sentence located (HBKSL) method to cover and extract all the CSIs in articles.

To address the above problems, this paper proposed GCO, an automatic generation approach of the CTI records based on multi-type OSTIPs, by combing the NLP method, machine learning method, and CTI knowledge. Our contributions are as follows:

(1) We train an article classification model with features selected by an experiment, which can classify articles published by multi-type OSTIPs efficiently. The cybersecurity intelligence details (CSIs) candidate number is proposed as a classification feature for the first time.

(2) We present the HBKSL method that can help extract list format CSIs and achieve good coverage of the CSIs. To be noticed, it is the first time that HTML structure is utilized to extract CTI information from open-source intelligence resources.

(3) We construct a Neo4j-based CTI database to store, visualize, and analyze the generated STIX 2.0 records, which can help reveal the interconnection of collected CTI records and the malicious threat groups.

The remainder of this paper is organized as follows: Sections 2 and 3 introduce the background and related works. In Section 4, we propose the method and describe its details. Section 5 presents the experiments and evaluation. Finally, the paper concludes in Section 6.

2. Background

2.1. OSTIP

With the progressive deterioration of cyber threats, more and more researchers pay attention to the solution of network security problems. Zheng et al. [9] proposed an approach to detect malicious TLS network traffic based on communication channel features. An automatic detection method of pornographic and gambling websites based on visual and textual content using a decision mechanism was discussed by Chen et al. [10]. Besides, with the wide application of CTI, security staff is sharing their findings on their blogs or open-source forums. These new findings have important reference significance, which can help security personnel to establish a broader and comprehensive understanding of advanced persistent threat (APT) groups. Discovering and responding in time can resist cyber-attacks better. These blogs or forums are established to publish CTI about the current cybersecurity events, as we call open-source threat intelligence publishing platforms (OSTIPs). However, the shared data are mostly in unstructured contexts and manual efforts are required to convert these contexts into machine-readable CTI records.

The OSTIPs can be divided into four types according to their owner, including company, individual, team, and forum. The company OSTIPs are established by companies, aiming to publish cybersecurity-related events, as well as the news and the product relevant advertisements. Articles published by this kind of OSTIPs are at a high volume and tend to be very detailed and cover every aspect of the cybersecurity events. For example, FireEyeBlog and ThreatVector [11] are famous company OSTIPs for researchers acknowledging recent happening impacted events. Individual OSTIPs are established by security staff who wish to share their new findings. Though the articles published by individual OSTIPs are at a low volume, its CTI information is also important for cybersecurity defense deployment. MalwareBreakdown [12] is a famous individual OSTIP publishing threat-articles weekly. The team OSTIPs are owned by cybersecurity industry teams, who are sharing their findings at a middle volume, like TeamUnit42 [13]. The forum OSTIPs account for a small proportion of all OSTIPs, as the operating cost are much higher than the other three types. Forum participants are sharing and discussing cybersecurity topics and asking for defense measures. Thus, the articles published by forum OSTIPs are written most informally. OTX [14] is a famous cybersecurity forum, where more than 19 million details are shared each day.

2.2. CTI Standard

To share the CTI records among different organizations and individuals, there are several CTI standards populated by security organizations—e.g., OpenIoC [15], CybOX [16], and STIX [17]. After several years' development, the structured threat information expression (STIX) 2.0 standard [18] is published as the next version of OpenIoC and STIX.

To share detailed information about cybersecurity threats, STIX 2.0 defines 12 STIX domain objects (SDOs)—including 'indicator', 'malware', 'report', 'vulnerability', and so on—which categorize each piece of information with specific attributes to be populated. Meanwhile, the STIX 2.0 defines two STIX relationship objects (SROs), which can chain multiple SDOs together through relationships and allow for easy or complex representations of CTI. An indicator SDO is shown in Figure 1. To describe the SDOs in detail, there are several properties defined. Some properties are common properties that can be used in all STIX SDOs and SROs, like type, id, created_by_ref. The other properties are specific properties that represent specific properties of the SDOs. As is shown in Figure 1, the pattern property is a specific property of Indicator SDO.

```
{
    "type": "indicator",
    "id": "indicator--9299f726-ce06-492e-8472-2b52ccb53191",
    "created_by_ref": "identity--39012926-a052-44c4-ae48-
caaf4a10ee6e",
    "created": "2017-02-27T13:57:10.515Z",
    "modified": "2017-02-27T13:57:10.515Z",
    "name": "Malicious URL",
    "description": "This URL is potentially associated with malicious
                    activity
                          and is listed on several blacklist sites.",
    "pattern": "[url:value = 'http://paypa1.banking.com']",
    "valid_from": "2015-06-29T09:10:15.915Z",
    "labels": [
      "malicious-activity"
    ]
```

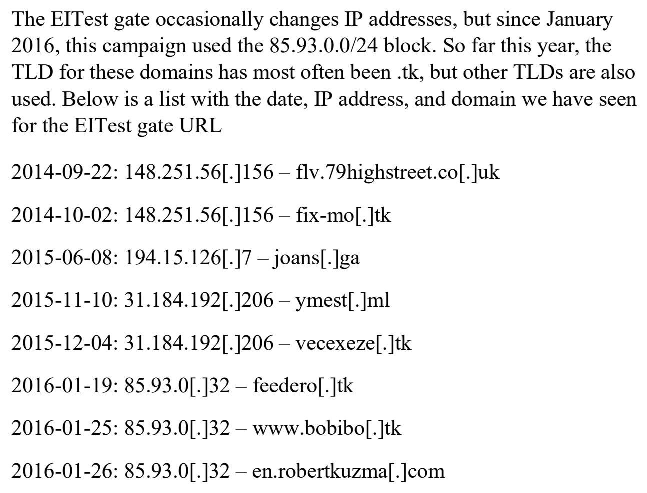
Figure 1. Example of the structured threat information expression (STIX) 2.0 Indicator SDO.

3. Related Work

To generate CTI records from open-source articles, some researches are proposed on combining manual efforts with machine learning techniques. Iqbal et al. [19] proposed STIXGEN to generate STIX 2.0 records from manually selected articles with some manipulation based on expert knowledge. In Porter [20], a method using LDA topic modeling was proposed to analyze the darknet market subreddit for evolutions of tools and trends. However, the CTI they found needed manual efforts to generate machine-readable threat intelligence. There is a method in finding newly developed malware and exploits that have not been deployed from the darknet market, as discussed by Nunes et al. [8], but the method also needed manual effort in converting the CTI into machine-readable CTI records.

To automatically generate CTI records from articles published by OSTIPs, there are mainly two challenges as described before. Several classification models are proposed to classify articles into threat-articles and non-threat-articles. Liao et al. [4] proposed an article classification model by running a support vector machine (SVM) model based on the features of topic word frequencies, article length, and dictionary-word density, and the targeted OSTIPs are selected by human efforts. Li et al. [21] presented an article classification model by running an SVM model with the TF-IDF scores of a list of topic words. In Wang et al. [22], a method was proposed, which utilized TF-IDF scores of article contexts, combined with chi-square statistics, as the feature, and trained the SVM model as the classifier. An automated threat report classification method over company OSTIPs was discussed by Ayoade et al. [23], and the method also utilized the TF-IDF scores as the classification feature. However, the bias-corrected classifier they trained can promote its accuracy and adaptive ability on article classification. Husari et al. [6] proposed running an SVM classifier with features, including article length, SecurityAction-word density, and SecurityTarget-word density, and the articles are only limited within the impacted company OSTIPs, like FireEyeBlog. Yang et al. [24] trained several classifiers by supervised machine learning algorithms, including decision tree (DT), support vector machines (SVM), and multiple layer perception neural networks (MLP-NN) to classify the CTI reports. Perry et al. [25] proposed an article classifier by running an eXtreme Gradient Boosting (XGBoost) model based on many empirical tests. Alves et al. [26] focused on designing a completed online monitoring system for cyber threat tweets on Twitter. They transformed tweets to vector representations and classifies the tweets as cyber threat relevant or irrelevant using binary classification models, particularly multi-layer perceptron (MLP) neural networks. Although there are existing methods with general good performances, most of them are compatible with only company OSTIPs. Specifically, due to the particularity of threat intelligence text structure in individuals, teams, and forum OSTIPs, those methods cannot achieve a satisfying classification performance. None of the existing methods are able to distinguish threat-articles from non-threat-articles published by multi-type OSTIPs. Thus, we summarize the often-used features and classification algorithms and select some of them to build our article classifier.

To extract CTI information and generate CTI records, there are mainly two kinds of methods proposed based on OpenIoC standard or STIX standard. Liao et al. [4] and Li et al. [21] presented different dependency parsers to analyze the CSI data located sentences and extract the CSIs with their relevant details. Dependency parsers mainly utilized NER techniques and machine learning methods. Mavroeidis et al. [27] and Ayoade et al. [23] proposed different CTI information extraction by building cybersecurity threat intelligence ontology. Iqbal and Anwar [28] demonstrated a combined ontological model of CKC [29] and POP [30] which can help extract CTI information from APT event-related articles. The ontology-based methods can fully extract the CTI details among the articles in theory but have not achieved good performances in practice, because the CTI ontology needs to have good coverage for the cybersecurity domain representation. Moreover, the manual ontology building is a heavy burden for cybersecurity researchers. All the existing methods only focus on the CSIs in sentence format and ignore to extract the list format CSIs, which results in low coverage on CSIs. The authors of Threat-articles are willing to share all the CSIs found in cyber events. Some of these CSIs are chosen as examples and described in sentence format, but most of these CSIs are shared in list format as lists of CSIs, and we call these CSIs in list format as CSI group. Figure 2 shows a CSI group, which contains several domains and IPs found in an event. The low coverage of CSIs will mislead the defense mechanisms and weaken the application of the generated CTI records.


The EITest gate occasionally changes IP addresses, but since January 2016, this campaign used the 85.93.0.0/24 block. So far this year, the TLD for these domains has most often been .tk, but other TLDs are also used. Below is a list with the date, IP address, and domain we have seen for the EITest gate URL

2014-09-22: 148.251.56[.]156 – flv.79highstreet.co[.]uk

2014-10-02: 148.251.56[.]156 – fix-mo[.]tk

2015-06-08: 194.15.126[.]7 – joans[.]ga

2015-11-10: 31.184.192[.]206 – ymest[.]ml

2015-12-04: 31.184.192[.]206 – vecexeze[.]tk

2016-01-19: 85.93.0[.]32 – feedero[.]tk

2016-01-25: 85.93.0[.]32 – www.bobibo[.]tk

2016-01-26: 85.93.0[.]32 – en.robertkuzma[.]com

Figure 2. A cybersecurity intelligence detail (CSI) group example.

4. Proposed Method

This section discusses our proposed approach in detail. We first outline the overall framework of the proposed automatic generation approach and then describe more technical details about each module in the framework.

4.1. Overview of the Proposed Approach

The architecture of the automatic approach to generate the CTI records based on multi-type OSTIPs (GCO) is as in Figure 3. The architecture includes article collector, article classifier, CSI extractor, and CTI records generator. Firstly, the article collector crawls articles from multi-type OSTIPs and preprocesses the collected articles. Secondly, for the articles collected by the article collector, the article classifier is trained based on features selected by an experiment to filter out non-threat-articles. Thirdly, the CSI extractor extracts all the cybersecurity intelligence details (CSIs) from threat-articles with the HBKSL method. The HBKSL is designed to locate the key sentence for a CSI-candidate or a CSI-candidate group. A CSI-candidate group means CSI-candidate data in list format, which cannot be extracted by existing methods, and we defined the key sentence as the sentence that can determine whether the CSI-candidates are valid or not. Finally, the CTI records generator converts the extracted CSI details into STIX 2.0 records with a set of mapping rules. To conveniently analyze the generated STIX 2.0 records, a CTI database is constructed by mapping the STIX 2.0 records into a Neo4j database and the interconnections among STIX 2.0 are revealed by a set of correlation analyses. The details about the proposed approach are described in the rest of this section.

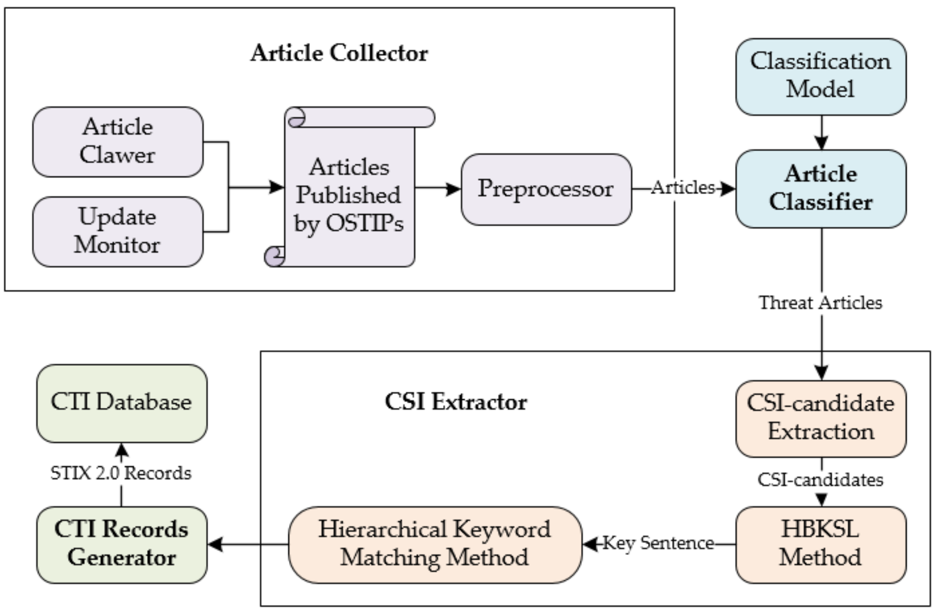


Figure 3. Architecture of the automatic approach to generate the cyber threat intelligence records based on multi-type OSTIPs (GCO).

4.2. *Article Collector*

To collect OSTIPs, firstly, we crawled the articles from the OTX forum, where members share their newly reading threat-articles. Every article indicates an OSTIP. After investing in these articles, we collect their reference URLs, which can link to other OSTIPs. Totally, there are 103 OSTIPs collected, containing 54% of company type of OSTIPs, 22% of individual OSTIPs, 18% of team OSTIPs, and 6% of community OSTIPs.

The article collector consists of an article clawer, an update monitor, and a preprocessor.

4.2.1. Article Clawer and Update Monitor

The article clawer collects all the articles published by an OSTIP for the first time and keeps on collecting articles under the monitoring of update monitor. We designed spider scripts for each OSTIP and extract relevant details from the webpage, including title, article content, publish time, and so on.

4.2.2. Preprocessor

The articles published by OSTIPs are mainly written in two kinds of documents. One type is HTML, which is commonly used as web pages. The other is PDF, which can be a formal report about a cybersecurity event or a threat group. HTML is the standard markup language for creating web pages, and the article is the main part of a web page along with other contexts, such as advertisement. HTML elements are the building blocks of HTML pages, which are delineated by tags, e.g., <title> and <p>. The tags can be ordered by category according to their function [31]. To improve the extraction efficiency, we manipulate the tags according to their category, as is listed in Table 1. To simplify the method, we used pdf2htmlEX [32] to convert the PDF document into HTML structure and parsed it like the normal HTML document. The online publishing tool renders PDF files in HTML, utilizing modern web technologies. It is also flexible for many different use cases.

Table 1. Different manipulation on tags according to the category.

Categories	HTML Tags	Manipulation
Basic HTML	body, br, head, p, title	None
Formatting	mark, small, strong	Remove the tag but keep the context
Forms and Input	button, input, output, select	Remove the tag and the context
Audio/Video	audio, source, track, video	Remove the tag and the context
Link	a, link, nav	Transform into basic tag and keep the context
Lists	dd, dir, dl, dt, li, ol, ul	None
Tables	caption, table, td, th, tr	Transform lines into Lists tags

Besides, to prevent the readers from inadvertently clicking on risky CSIs, the authors change the “http” to “hxxp” or add “[]” among dot in a URL, and so on. The misspelled CSIs are transformed into their original context by NLP techniques.

4.3. Article Classifier

Article classifier is designed to filter out Non-threat-articles, which includes a feature extraction module and a machine learning module. The framework of the article classifier is shown in Figure 4. We extracted features from the text classification data set described in Section 5.1 and used them to train the text classification model. We performed feature extraction and construct feature vectors for the new input text, and finally used the trained text classification model to determine whether the text is a threat intelligence text.

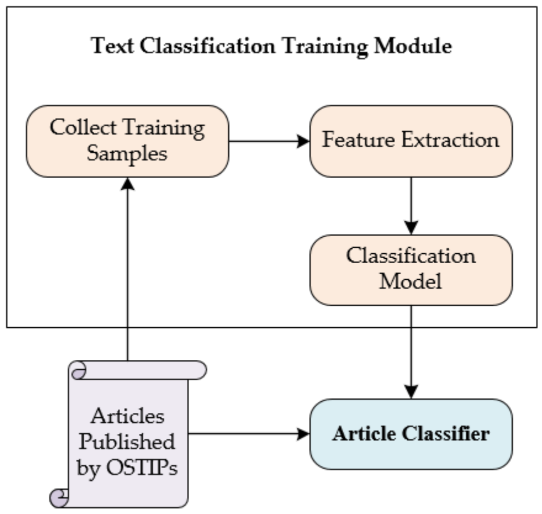


Figure 4. Framework of the article classifier.

4.3.1. Feature Extraction

To classify articles into threat-articles and non-threat-articles published by multi-type OSTIPs, several features can be used to train article classification models as described before. iACE [4] uses the features including topic word, dictionary-word ratio, and article length. TTPDrill [6] uses the combination which contains SecurityAction-word density, SecurityTarget-word density, and Article length. To be noticed, the CSI-candidate number is proposed as a classification model feature for the first time.

CSI-candidate number: We defined the domain, URL, hash, CVE records, and IP in the article as CSI-candidate, as these data have a possibility to be CSI. In Non-threat-articles, most of the CSI-candidates are referring to article URLs.

Topic word: Threat-articles focus on security risks and vulnerabilities, whose theme is reflected by its topic terms. These terms are less likely to appear in a non-IoC article. We firstly construct a set of topic words by combining the top 10 terms of all articles and then compute the TF-IDF score of topic words respectively. The vector of TF-IDF scores is used as the feature for classification.

Article length: The non-threat-articles published by the OSTIPs mostly are product advertisements, company events, or some conference notification, and tend to be short. The threat-articles contain many details about the security events, which results in its longer length.

Dictionary-words density: Compared with Non-threat-articles, Threat-articles tend to have a lower dictionary-word density, because they contain more none-dictionary nouns (e.g., registry, file path). The dictionary words can be found by the pyspellchecker library. Then, the density is calculated as the ratio of the dictionary words with regards to all the words the article contains.

SecurityAction-word density: We extracted all verbs from various highly-reputable publicly available information security standards, namely, ATT&CK, CWE, and STIX 2.0 standard using the part-of-speech (POS) NLP technique. We calculate the SecurityAction-word density by computing the percentage of times these verbs appear in an article compared to the total number of words. Threat-articles tend to have a higher SecurityAction-word density than non-threat-articles.

SecurityTarget-word density: Threat-articles describe threats. Therefore, they contain more security nouns (e.g., registry, weapon). Like SecurityAction-word density, we extract all noun phrases from the same information security standards and use them to calculate SecurityTarget-word density, by computing the percentage of times these noun-phrase appear in an article compared to the total number of words in it.

According to feature engineering technology, we used feature combinations instead of a traditional single feature. For example, the combination contains the topic word, CSI-candidate number, and article length. The feature extraction module extracts three types of features, namely the extracted topic word represented by GetTopicWord, the number of extracted suspected CSI data represented by GetCSINumber, and the extracted article length represented by GetContextLength. The system controls the extraction of three types of features through ArticleFeatureExtractor. The specific feature extraction sequence diagram is shown in Figure 5.

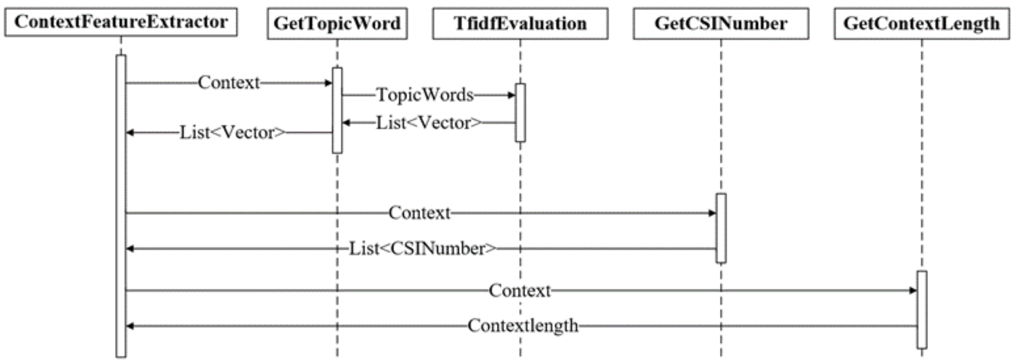


Figure 5. Specific feature extraction sequence diagram.

We experimented all the combination of these features with classification algorithms and showed the top five results. The experiment is described in Section 5.2.

4.3.2. Classification Based on the Multi-Layer Perceptron

We trained the multi-layer perceptron (MLP) model by extracting features from articles. Firstly, the topic words extracted from the article are evaluated by the TF-IDF algorithm, and the topic words extracted from the training set are mapped to the word frequency vector T.

Then, we combined the features into an input vector $Z = [T, N_{IP}, N_{domain}, N_{URL}, N_{hash}, L]$ to construct an MLP model with a hidden layer, as shown in Equation (1)

$$F_{MLP} = W^T S\left(V^T Z\right) \tag{1}$$

N_{IP} represents the number of suspected IP data in the article. N_{domain} represents the number of suspected domain name data in the article. N_{URL} represents the number of suspected URL data in the article. N_{hash} represents the number of suspected hash value data in the article. L represents the length of the article, V represents the interconnection weight matrix between the input layer and the first hidden layer, and W represents the interconnection weight matrix from the first hidden layer to the output layer. S is the activation function of the MLP model. The L-BFGS algorithm is selected to solve the MLP model, and after model training, a threat intelligence text classification model is obtained.

The MLP algorithm is based on the MLPCLassifier of the Scikit-Learn library. The core code of MLP model construction is as follows:

```
from sklearn.neural_network import MLPClassifier
# Define the MLP model
clf = MLPClassifier(hidden_layer_sizes = (30,20),
activation = 'relu', solve r= 'lbfgs', alpha = 0.0001,
learning_rate_init = 0.01, power_t = 0.5,
max_iter = 180, nesterovs_momentum = True,
early_stopping = True, validation_fraction = 0.1
```

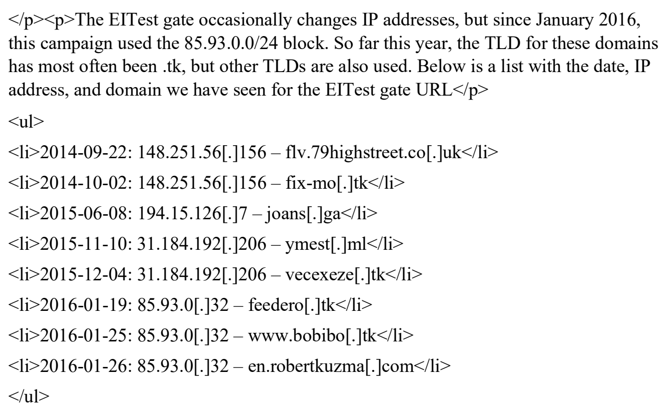
We concluded the article classification model by running a multi-layer perceptron (MLP) model based on the feature combination of CSI-candidate number, topic word, and dictionary-words density, which reached an accuracy of 94.29%.

4.4. The CSI Extractor

The CSI extractor extracts all the cybersecurity intelligence details (CSIs) from threat-articles with the HBKSL method. The HBKSL is designed to locate the key sentence for a CSI-candidate or a CSI-candidate group. We define the domain, URL, hash, CVE records, and IP in the article as CSI-candidate, as these data have a possibility to be CSI. A CSI-candidate group means CSI-candidate data in list format, which cannot be extracted by existing methods, and we defined the key sentence as the sentence that can determine whether the CSI-candidates are valid or not.

As Figure 2 shows, a CSI group cannot be extracted according to their embedded sentences. However, the sentence "Below is a list with the date, IP address, and domain … " indicates the following CSI-candidates are valid. By utilizing the HTML tag distribution characteristics of the CSIs and its relevant key sentence, GCO has achieved good coverage of the CSIs.

As Figure 6 shows, "fix-mo.tk" and "flv.79highstreet.co.uk" can be validated by their relevant key sentence, and the two CSIs have the same HTML properties. We define a CSI-candidate with its embedded HTML tag, HTML attribute, the parent tag of the HTML tag, and the attribute of the parent tag as a CSI-dataset. The CSI-datasets with the same HTML properties are clustered into a CSI-candidate group. Then we apply the HBKSL method and a hierarchical keyword matching (HKM) method on the CSI-candidate groups to extract the CSIs with its relevant details, e.g., type. The working flow of the CSI Extractor is shown in Figure 7.



```
</p><p>The EITest gate occasionally changes IP addresses, but since January 2016,
this campaign used the 85.93.0.0/24 block. So far this year, the TLD for these domains
has most often been .tk, but other TLDs are also used. Below is a list with the date, IP
address, and domain we have seen for the EITest gate URL</p>
<ul>
<li>2014-09-22: 148.251.56[.]156 – flv.79highstreet.co[.]uk</li>
<li>2014-10-02: 148.251.56[.]156 – fix-mo[.]tk</li>
<li>2015-06-08: 194.15.126[.]7 – joans[.]ga</li>
<li>2015-11-10: 31.184.192[.]206 – ymest[.]ml</li>
<li>2015-12-04: 31.184.192[.]206 – vecexeze[.]tk</li>
<li>2016-01-19: 85.93.0[.]32 – feedero[.]tk</li>
<li>2016-01-25: 85.93.0[.]32 – www.bobibo[.]tk</li>
<li>2016-01-26: 85.93.0[.]32 – en.robertkuzma[.]com</li>
</ul>
```

Figure 6. CSIs in its Hyper Text Markup Language (HTML) structure.

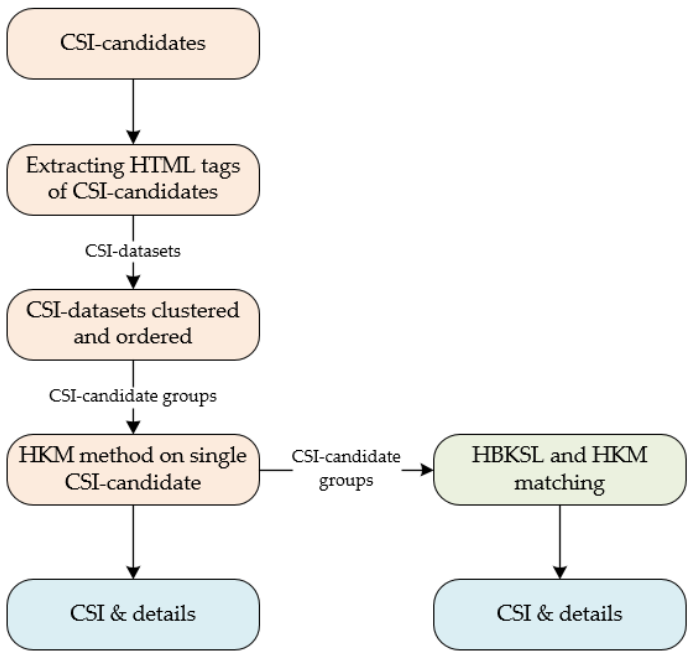


Figure 7. Clustering and extraction working flow.

The CSI-candidate embedded sentence is firstly checked by the HKM method, aiming to get the sentence format CSIs validated and out of the CSI-candidate group. The HBKSL method and HKM method are described in the rest of this section.

4.4.1. HTML-Based Key Sentence Location Method

The algorithm of the HBKSL method is shown as in Algorithm 1, and its main idea is to find the sentence ahead of the present CSI-candidate. A present CSI-candidate is defined as the first CSI-candidate sorted in the CSI-candidate group. We used the CSI-candidate group and its present CSI-candidate as input. According to its HTML structure, firstly, we found out whether there are sibling nodes that are not empty. Then, parent nodes that are not empty should be noticed, and finally, we judged whether there are non-empty uncle nodes. Taking the CSI-candidates shown in Figure 6 as an example, the present CSI-candidate is "148.251.56.156" and the relevant key sentence is embedded in the HTML <p> ahead of the present CSI-candidate.

Algorithm 1. HTML-based key sentence location.

```
Input: CSI-candidate group, present_CSI_candidate;
Input: preprocessed HTML document;
Output: key_sentence;
1.  function find_text_based_html(present_CSI, html)
2.    pre_items ← present_CSI_candidate.tag.find_previous_tag()
3.    if pre_items.text not NULL then
4.        key_sentence ← pre_items.text
5.        return key_sentence
6.    parent_items ← present_CSI_candidate.tag.find_parent_tag()
7.    if parent_items.text not NULL then
8.        key_sentence ← parent_items.text
9.        return key_sentence
10.   uncle_items ← present_CSI_candidate.tag.find_nearest_uncle_tag()
11.   if uncle_items.text not NULL then
12.       key_sentence ← uncle_items.text
13.       return key_sentence
14.   return NULL
```

According to the HTML structure, the key sentence embedded tag can be the previous tag of the CSI-candidate contained tag, as well as the parent tag and the nearest uncle tag. The nearest uncle tag means the tag is in the same hierarchy with the parent tag and ahead of the parent tag. Figure 8 shows the relationship between nodes detailly in the example. The <p> tag is the nearest uncle tag of the <li> tag in Figure 6.

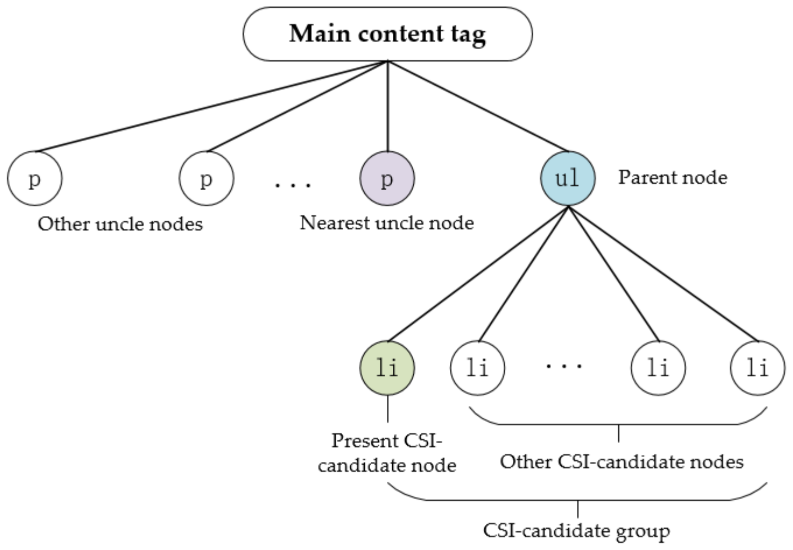


Figure 8. Relationship between nodes in HTML structure.

4.4.2. Hierarchical Keyword Matching Method

To determine whether the CSI-candidate embedded sentence or the key sentence is valid or not, we use a hierarchical keyword matching (HKM) method. HKM method can help determine the CSI type, e.g., 'phish'. We gathered first-class keywords and second-class keywords from key sentences by running the HBKSL method over classified articles. The first-class keywords contain words like 'zombie' and 'worm' that can indicate the type of CSI-candidate. The second-class keywords contain words like 'sample' and 'indicator'. The second-class keyword can indicate the related CSI-candidate group are the list of CSIs but cannot determine the type of CSIs, e.g., 'sample'. For the CSI distinguished by the second-class keywords, the HKM method will match the first-class keywords from the topic words of the article. Algorithm 2 shows the hierarchical keyword matching method.

Algorithm 2. Hierarchical keyword matching.

```
Input: decision_text; (key_sentence)
Input: topicword_list;
Input: keyword_list;
Output: CSI_type;
1.  function match_keyword(decision_text, topicword_list)
2.      sentences ← sentence_token(decision_text)
3.      last_sentence ← find_last_sentence(sentences)
4.      words ← PorterStemmer(last_sentence.words)
5.      for word in words do
6.          if word in keyword_list.first_class then
7.              CSI_type ← word
8.              return CSI_type
9.          if word in keyword_list.second_class then
10.             for topicword in topicword_list do
11.                 if topicword in keyword_list.first_class then
12.                     CSI_type ← topicword
13.                     return CSI_type
14.                 else
15.                     CSI_type ← topicword
16.                     return CSI_type
17. return NULL
```

4.5. CTI Records Generator

As described before, the CTI can help cybersecurity researchers understand the network situation and take proper measures against potential threats. Thus, the automatically generated CTI records should be concise and readable [33]. The threat intelligence generator is designed to generate STIX 2.0 records over the extracted CSIs and their related details, including the type and other properties of threat-article.

Specifically, our method is designed to generate indicator, vulnerability, and report STIX domain objects (SDOs) along with the SROs among them. Taking Indicator SDO as an example, the CSI value will fill the pattern property, while the modified property and the created property will be filled with the system's current time. The description property will be filled by the article's summarizations, which are generated from the article's content with the TextRank algorithm. Summarizations can reveal the development of cyber events. Here we established mapping rules to convert the matched CSI type value into formal STIX vocabulary. Part of the mapping method is shown in Table 2. With the mapping method, one type of value can be converted into multi labels.

Table 2. Mapping method of labels and CSI types.

SDO	Labels	CSI Type Value
Indicator	compromised	backdoor, injected, payload, indicator
	anomalous-activity	spammer, advertisement, abus, ad
	malicious-activity	phish, brute, suspicious, scammer
	anonymization	tor, vpn, dns, botnet, proxy
Report	vulnerability	exploitd, vulner, kit, cve
	indicator	ioc, sample, refer, domain, indicator
	campaign	attack, campaign

4.6. CTI Database

To use the generated STIX 2.0 records efficiently, a CTI database is constructed based on Neo4j. As Neo4j contains four basic data structures: node, label, relationship, and property [34], a set of mapping rules are designed as below:

Nodes—Nodes are used to represent entities of the graph. Indicator, vulnerability, and report SDOs are regarded as nodes and the name of each node is assigned as their type. The indicator SDO is mapped as the Indicator node. The vulnerability SDO is mapped as the vulnerability node. The report SDO is mapped as the report node.

Label—Labels are used to shape the domain by grouping nodes into sets where all nodes that have a certain label belongs to the same set. Each node can have one or more labels, in case when a report contains more than one type of property. We use labels property of the STIX 2.0 record as the node label.

Property—Property is the information of a node that does not interact with other same type nodes. We regard all the metadata of the STIX 2.0 record as the node's property. Property is represented as multi key-value pairs.

Relationship—The relationship is a connection between different nodes. Since the generated STIX 2.0 records only contain three types of SDOs, the only relationship that needs to be established is 'REFERS_TO' while the report nodes as the start node.

5. Experiments and Evaluations

In this section, we described the design and process of the experiments to verify the effectiveness of the proposed method. First, we described the dataset collected for our experiment. Then the experimental design and result analysis are described in two parts to verify the effectiveness of our proposed method. Finally, we constructed a Neo4j-based CTI database to store, visualize, and analyze the generated STIX 2.0 records.

5.1. System Implementation and Dataset

All experiments in this work are performed on Windows PC, Table 3 summarizes the major hardware and software platforms environment for GCO.

Table 3. Platforms environment.

Platforms	Content
Hardware dependencies	Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz 16GB 1600MHz DDR3
Software dependencies	Microsoft Windows 10 Python v3.8.1 MongoDB Community Server 4.2.5 Neo4j 4.0 Java(TM) SE Runtime Environment (build 1.8.0_241-b07)

Due to the sensitivity of the confidential project, the datasets used in most related work are not open to the public. To verify the effectiveness, we run GCO over the collected OSTIPs and gathered 24,835 articles published from 2010 to 2019, the contents of articles are shown in Table 4. These OSTIPs contain many famous publishing platforms, like FireEyeBlog and ThreatVector. Some sources of threat intelligence are shown in Table 5. GCO generated 92,012 CTI in STIX 2.0 format—including 7179 report SDOs; 84,309 IoC SDOs; 524 vulnerability SDOs.

Table 4. Article samples dataset.

Number	Articles	Sample Size
1	Threat-articles	7179
2	Non-threat-articles	17,656

Table 5. Some sources of threat intelligence dataset.

Number	The Type of OSTIPs	Threat Intelligence Source
1	Team	Checkpoint, f5lab, f-secure, badcyber
2	Individual	Dynamo, bartblaze, phishtotal, eternaltodo
3	Company	Fireeye, bleepingcomputer, trendmicro
4	Forum	Conta, technetMicrosoft, juniper

We also construct a labeled dataset to select the feature combination and classification algorithm for the article classification model. The articles are collected from multi-type OSTIPs and parsed by the article collector. The dataset is manually checked and contains 500 threat-articles and 500 non-threat-articles. The collected threat-articles included impacted events like 'Torii Botnet' and 'MuddyWater APT attack', as well as some little influenced accidents reported by teams and individuals. The detailed information is shown in Table 6.

Table 6. Labeled datasets for training article classification model.

Dataset	Team OSTIPs	Individual OSTIPs	Company OSTIPs	Forum OSTIPs
Threat-articles	90	130	250	30
Non-threat-article	90	130	250	30

Below, we evaluated the article classification method and the CSI extraction method, which restrict the performance of our proposed approach. We also designed a set of correlation analysis on the generated STIX 2.0 records based on the constructed CTI database to mine the relationships among threat-articles and STIX 2.0 records.

5.2. Article Classification Model

To select a high-performance feature combination, we trained the classification model on the traditional machine learning algorithms with combinations of features and obtain the classification performance by average the results from running 10-fold cross-validation 10 times. In each fold, a random sample of approximately 10% was hidden from the training algorithm and used as test data. Even though the SVM algorithm is the most common traditional classifier in threat-article classification, other classifiers were also trained as comparisons, such as multi-layer perceptron (MLP), decision trees, and random forest. Moreover, the XGBoost algorithm [35] is also tested on building a threat-article classification model, which is also a popular classification algorithm and can solve many data science problems in a fast and accurate way. The five classifiers are built based on the Python scikit-learn library [36], and we utilized the default parameters specified in the Python scikit-learn library. The top five results are shown in Table 7. We use some feature combinations representing the combination of multi-features.

Table 7. Classification accuracy of five classifiers with different feature combinations.

Feature Combination	Feature Name	XGBoost	MLP	SVM	Decision Trees	Random Forest
A	CSI-candidate number Topic word Dictionary-word ratio	94.85%	94.29%	82.52%	92.19%	60.13%
B	CSI-candidate number SecurityTarget-word density Topic word Dictionary-word ratio	94.81%	94.25%	82.79%	92.09%	60.48%
C	CSI-candidate number Article length	94.75%	81.79%	72.24%	91.36%	91.77%

Table 7. *Cont.*

Feature Combination	Feature Name	XGBoost	MLP	SVM	Decision Trees	Random Forest
D	CSI-candidate number Topic word	94.68%	94.29%	82.05%	92.05%	60.13%
E	CSI-candidate number SecurityAction-word density Topic word Dictionary-word ratio CSI-candidate number	94.65%	94.27%	82.53%	92.23%	60.53%

The results clearly indicate that the XGBoost outperforms traditional algorithms. The differences between accuracy values of XGBoost and SVM are significant, as well as decision trees and random forest. While the XGBoost and MLP results are on par with each other, the XGBoost has a slight advantage. Furthermore, according to the result, the CSI-candidate number is a distinguishing feature in building the threat-article classification model.

Then we turn our focus to the training time and testing time of the top five classification accuracy feature combination. The training time and testing time are shown in Table 8. As we can see, the MLP classifiers need more training time than the XGBoost classifier but need less testing time in most cases. However, for combination A, the MLP classifiers show better performance including higher accuracy, less training time, and less testing time. Considering the application of the article classifier, we trained threat-article classifier by running the MLP model over the feature combination of CSI-candidate number, topic word, and dictionary-words density.

Table 8. Training time and testing time of MLP and XGBoost on the top five classification accuracy feature combination.

Feature Combination	Training Time (sec)		Testing Time (sec)	
	MLP	XGBoost	MLP	XGBoost
A	37.7004	41.3502	0.0005	0.0046
B	37.4184	16.5013	0.0003	0.0052
C	0.2479	1.5594	0.0004	0.0055
D	38.3013	17.4294	0.0009	0.0041
E	37.6032	15.8128	0.0008	0.0037

To compare our approach with state-of-the-art classification methods, we select iACE [4] and TTPDrill [6] as comparisons. iACE uses the combination F and trained SVM model as a threat-article classifier, and the TTPDrill uses the combination H and trained SVM model as a threat-article classifier. The results are shown in Table 9. Although the article length has been used in iACE and TTPDrill, the results indicate the article length does not have a good performance as the threat-article feature, comparing with the CSI-candidate number. Considering the multi-types of OSTIPs, the threat-article length has a poor distinguishability on building threat-article classification models. Although the SVM algorithm has been considered as a common solution in article classification, the experiment shows that the XGBoost and MLP can classify the threat-articles from different types of OSTIPs better.

Table 9. Classification accuracy of classifiers with article length as feature.

Feature Combination	Feature Name	XGBoost	MLP	SVM	Decision Trees	Random Forest
F	Topic word Dictionary-word ratio Article length	80.13%	65.71%	51.15%	74.73%	60.00%
H	SecurityAction-word density SecurityTarget-word density Article length	63.16%	54.44%	49.99%	60.31%	64.13%

5.3. CSI Extraction Accuracy and Coverage

To validate the performance of CSI extraction, we employ the measurement of precision and recall. As described before, the written styles are different among the four types of OSTIPs, and the difference results in different CSI extraction performances. Thus, we randomly sampled 10 articles from each type OSTIPs and statistic the CSIs in these articles manually. Then we validated the extracted CSIs with the verified CSIs. The results are shown in Table 10. As we can see from the result, GCO has a good performance in extracting CSIs from company OSTIPs and team OSTIPs and does not achieve high performance on forum OSTIPs. That is because the articles on forums are sometimes written in informal ways. The description of the shared CSIs may not be put in an easy understanding way, and the ambiguous written style lowers the extraction of GCO. The missing extraction CSIs mainly resulted from the key sentences like 'References', which can indicate a set of CSIs discussed in the article, as well as a list of reference article links.

Table 10. Accuracy of CSI extractor.

OSTIP Type	CSIs in Articles	Pre.	Rec.
Team	121	95.04%	93.50%
Individual	183	94.35%	86.08%
Company	116	98.28%	99.13%
Forum	94	85.88%	91.25%

Compared with the other OSTIPs, the GCO achieved high performance on company OSTIPs with a precision of 98.28% and a recall of 99.13%. That is because the articles on company OSTIPs are carefully written, and the CSIs are described in formal ways. Moreover, the GCO's extraction ability is better than iACE, which is designed for company OSTIPs and has a precision of 98% and a recall of 93% on CSI extraction.

5.4. Analysis the STIX 2.0 Records

By the analysis, we got the Threat-articles clustered into their related events and got the events clustered into potential threat groups that may be responsible for the cybersecurity events.

Firstly, we did a correlation analysis on the report nodes to generate "Event" nodes. Specifically, for two report nodes, an Event node will be established if their referred IoC nodes and vulnerability nodes have more than two pattern properties with the same value. The relationship between the newly built Event node and the two nodes is defined as "RELATED_TO", and a report node must be related to an Event node. Secondly, we generated the "Potential-threat-group" nodes from the established Events nodes. Specifically, two Event nodes will be clustered into the same Potential-threat-group node if their correlated IoC nodes or vulnerability nodes have more than one pattern properties with the same value. Note that a Potential-threat-group node must contain more than three articles. We defined the relationship between the Potential-threat-group node and the Event node as "RESPONSIBLE_FOR".

Figure 9 is an established Potential-threat-group node and its related nodes. By checking in manually, we found the Potential-threat-group node indicated the APT28 group, and the related report nodes indicated the articles published by cybersecurity teams aiming to discover this Russian government conducting threat group.

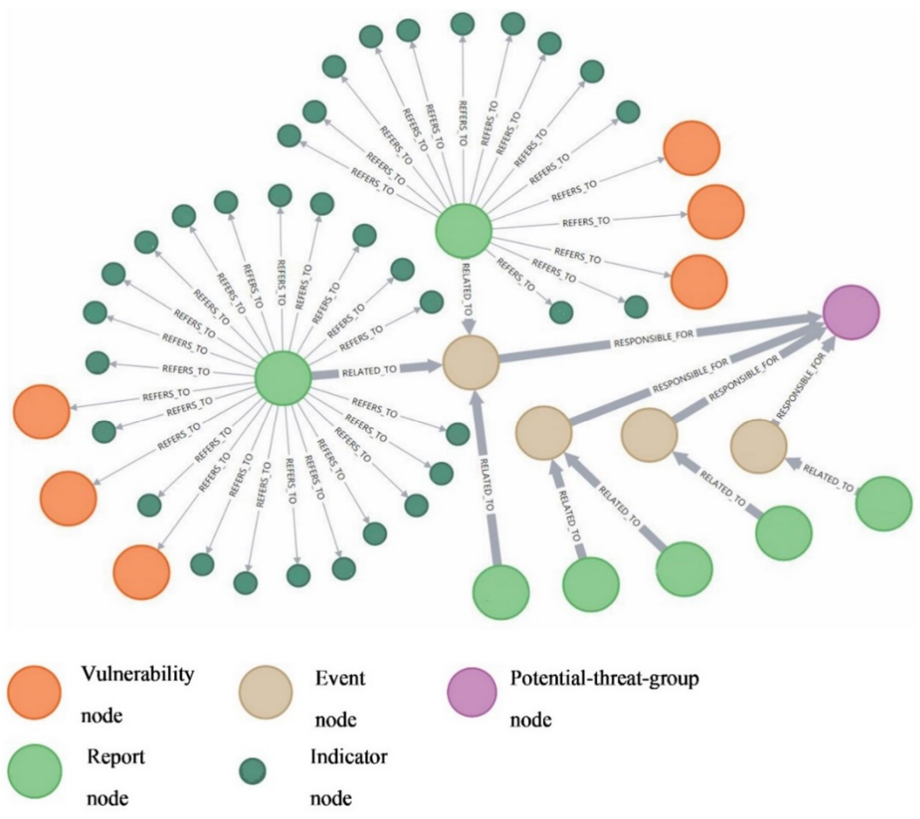


Figure 9. A generated Potential-threat-group node.

The experiment results proved the CTI database can help security staff efficiently query and analyze the data, and the query results can be visualized directly. The CTI records obtained through GCO can effectively help security staff perform more efficient work, including CTI sorting and further analysis. We hope that GCO can provide more information to assist security personnel in making decisions, and summarize the attack methods and tools commonly used by APT groups. When new security threats appear, the GCO can help security staff detect and deal with them in time, which can also minimize losses, predict the source of the cyber-attack, and locate the potential APT group.

6. Conclusions

In this paper, an automatic generation approach of the CTI records based on multi-source information fusion—called 'GCO'—is presented, which is designed to collect articles published by multi-type OSTIPs, classify the threat-articles and non-threat-articles, generate machine-readable STIX 2.0 records and construct a CTI database based on Neo4j database. With the progressive deterioration of cyber threats, the GCO automatically processes a large amount of threat intelligence, which greatly improves the work efficiency of network security practitioners. On the other hand, the proposed approach can help information security personnel grasp the public opinions with specific pertinence, handle the emergency events, even confront the advanced persistent threats.

With an experiment on common article classification algorithms and classification features, we choose a multi-layer perceptron algorithm to train the article classifier for classifying the articles published by multi-type OSTIPs with feature combination of CSI-candidate number, topic word, and dictionary-words density. The result shows our article classification model is over-performed than the state-of-the-art article classification method, which reached an accuracy of 94.29%. With a combination of the HBKSL method and some technologies in the NLP community, GCO can efficiently validate the CSI-candidates in threat-articles and generate the STIX 2.0 records with the extracted details. The GCOs extraction achieved high performance on company OSTIPs with a precision of 98.28% and a recall of 99.13%. The evaluation of the datasets demonstrates that our method has good performance on article classification and CSIs extraction, and the analysis based on the CTI database reveals intrinsic connections among generated STIX 2.0 records and provides a good method to analyze CTI, which highlights the significance of our method. In future work, we plan to collect more cyber threat intelligence and analyze indicators of compromise (IoC) such as attack methods and tools in the CTI records to help security personnel build a more comprehensive and systematic knowledge graph, and provide assistance in tracing the source of cybersecurity attacks.

Author Contributions: Conceptualization, P.Y. and T.S.; Methodology, M.L., P.Y., and T.S.; Validation, T.S. and M.L.; Formal analysis, P.Y. and S.L.; Investigation, P.Y. and T.S.; Resources, T.S. and M.L.; Data curation, T.S.; Writing—original draft preparation, P.Y.; Writing—review and editing, T.S., P.Y., and S.L.; Visualization, T.S. and S.L.; Supervision, P.Y.; Project administration, P.Y. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research and Development Program of Sichuan Province of China (grant no. 2020YFG0076).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the sensitivity of the confidential project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Definition: Threat Intelligence. Available online: https://www.gartner.com/en/documents/2487216 (accessed on 21 June 2020).
2. CleanMX. Available online: https://support.clean-mx.com/clean-mx/viruses.php (accessed on 21 June 2020).
3. Phish Tank. Available online: https://www.phishtank.com/ (accessed on 21 June 2020).
4. Liao, X.; Yuan, K.; Wang, X.; Li, Z.; Xing, L.; Beyah, R. 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, Vienna, Austria, 24–28 October 2016; pp. 755–766.
5. Fire Eye Blog. Available online: https://www.fireeye.com/blog.html. (accessed on 21 June 2020).
6. Husari, G.; Al-Shaer, E.; Ahmed, M.; Chu, B.; Niu, X. TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources. In Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, 4–8 December 2017; pp. 103–115.
7. Khan, L.; Luo, F. Ontology construction for information selection. In Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002), Washington, DC, USA, 4–6 November 2002; pp. 122–127.
8. Nunes, E.; Diab, A.; Gunn, A.; Marin, E.; Mishra, V.; Paliath, V.; Robertson, J.; Shakarian, J.; Thart, A.; Shakarian, P. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics: Cybersecurity and Big Data (ISI 2016), Tucson, AZ, USA, 28–30 September 2016; pp. 7–12.
9. Zheng, R.; Liu, J.; Li, K.; Liao, S.; Liu, L. Detecting Malicious TLS Network Traffic Based on Communication Channel Features. In Proceedings of the 2020 IEEE 8th International Conference on Information, Communication and Networks (ICICN), Xi'an, China, 22–25 August 2020; pp. 14–19. [CrossRef]
10. Chen, Y.; Zheng, R.; Zhou, A.; Liao, S.; Liu, L. Automatic Detection of Pornographic and Gambling Websites Based on Visual and Textual Content Using a Decision Mechanism. *Sensors* **2020**, *20*, 3989. [CrossRef] [PubMed]
11. Threat Vector. Available online: https://threatvector.cylance.com/en_us/home.html (accessed on 21 June 2020).
12. Malware Breakdown. Available online: https://malwarebreakdown.com/ (accessed on 21 June 2020).
13. Team Unit 42. Available online: https://unit42.paloaltonetworks.com/ (accessed on 21 June 2020).
14. OTX. Available online: https://otx.alienvault.com/ (accessed on 21 June 2020).
15. Wilson, D. The History of Openioc. Available online: https://www.fireeye.com/blog/threat-research/2013/09/history-openioc.html (accessed on 21 June 2020).
16. Casey, E.; Back, G.; Barnum, S. Leveraging CybOXTM to Standardize Representation and Exchange of Digital Forensic Information. Available online: https://www.dfrws.org/sites/default/files/session-files/pres-leveraging_cybox_to_standardize_representation_and_exchange_of_digital_forensic_information.pdf (accessed on 21 June 2020).

17. MITRE. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX TM). Available online: http://www.standardscoordination.org/sites/default/files/docs/STIX_Whitepaper_v1.1.pdf (accessed on 21 June 2020).
18. MITRE. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX). Available online: https://oasis-open.github.io/cti-documentation/ (accessed on 21 June 2020).
19. Iqbal, Z.; Anwar, Z.; Mumtaz, R. STIXGEN-A novel framework for automatic generation of structured cyber threat information. In Proceedings of the 2018 International Conference on Frontiers of Information Technology (FIT 2018), Islamabad, Pakistan, 17–19 December 2018; pp. 241–246.
20. Porter, K. Analyzing the DarkNetMarkets subreddit for evolutions of tools and trends using LDA topic modeling. *Digit. Investig.* **2018**, *26*, S87–S97. [CrossRef]
21. Li, K.; Wen, H.; Li, H.; Zhu, H.; Sun, L. Security OSIF: Toward automatic discovery and analysis of event based cyber threat intelligence. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovations (SmartWorld/UIC/ATC/ScalCom/CBDCo), Guangzhou, China, 8–12 October 2018; pp. 741–747.
22. Wang, J.; Zhu, T.-C.; Kang, J.-W.; Li, B. Text Classification and Threat Intelligence Generation for Industrial Control System Security. In *DEStech Transactions on Computer Science and Engineering*; DEStech Publications: Lancaster, PA, USA, 2018; pp. 575–581.
23. Ayoade, G.; Chandra, S.; Khan, L.; Hamlen, K.; Thuraisingham, B. Automated threat report classification over multi-source data. In Proceedings of the 4th IEEE International Conference on Collaboration and Internet Computing (CIC 2018), Philadelphia, PA, USA, 18–20 October 2018; pp. 236–245.
24. Yang, W.; Lam, K.Y. Automated Cyber Threat Intelligence Reports Classification for Early Warning of Cyber Attacks in Next Generation SOC[C]. In Proceedings of the International Conference on Information and Communications Security, Beijing, China, 15–17 December 2019; Springer: Cham, Switzerland, 2019; pp. 145–164.
25. Perry, L.; Shapira, B.; Puzis, R. NO-DOUBT: Attack Attribution Based On Threat Intelligence Reports. In Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 1–3 July 2019; pp. 80–85.
26. Alves, F.; Bettini, A.; Ferreira, P.M.; Bessani, A. Processing tweets for cybersecurity threat awareness. *arXiv* **2019**, arXiv:1904.02072. [CrossRef]
27. Mavroeidis, V.; Bromander, S. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC 2017), Athens, Greece, 11–13 September 2017; pp. 91–98.
28. Iqbal, Z.; Anwar, Z. Ontology Generation of Advanced Persistent Threats and their Automated Analysis. *NUST J. Eng. Sci.* **2016**, *9*, 68–75.
29. Hebert, A.J. Compressing the kill chain. *Air Force Mag.* **2003**, *86*, 50–55.
30. Sager, T. Killing Advanced Threats in Their Tasks: An Intelligence Approach to Attack Prevention. SANS. 29 July. Available online: https://www.sans.org/reading-room/whitepapers/detection/paper/35302 (accessed on 21 June 2020).
31. Hwangbo, H.; Lee, H. Reusing of information constructed in HTML documents: A conversion of HTML into OWL. In Proceedings of the 2008 International Conference on Control, Automation and Systems (ICCAS 2008), Seoul, Korea, 14–17 October 2008; pp. 871–875.
32. pdf2htmlEX. Available online: https://github.com/coolwanglu/pdf2htmlEX (accessed on 21 June 2020).
33. Sillaber, C.; Sauerwein, C.; Mussmann, A.; Breu, R. Data Quality Challenges and Future Research Directions in Threat Intelligence Sharing Practice. In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, Vienna, Austria, 24 October 2016; pp. 65–70.
34. Percuku, A.; Minkovska, D.; Stoyanova, L. Modeling and Processing Big Data of Power Transmission Grid Substation Using Neo4j. *Procedia Comput. Sci.* **2017**, *113*, 9–16. [CrossRef]
35. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
36. Pedregosa, F. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.