

A Practical Cybersecurity Ontology Generator Based On Hierarchical Clustering and Multi-Way Tree

Yixuan Wang¹ , Bo Zhao^{1*} , Xiaofu Song¹ , Jiahui Zhu² 

¹ School of Cyber Science and Engineering, Wuhan University, Wuhan (China)

² School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu (China)

* Corresponding author: zhaobo@whu.edu.cn

Received 3 December 2023 | Accepted 1 October 2025 | Early Access 12 March 2026



ABSTRACT

Cybersecurity ontology development is typically carried out by cybersecurity experts and ontology engineers. While some existing works focus on extracting cybersecurity knowledge from either textual or structured data, few address the challenge of handling both types of data simultaneously. This paper presents Locust, a tool integrating structured data and domain corpus for comprehensive cybersecurity ontology generation. We use open source cybersecurity specifications as structured input to build the skeleton of the ontology, and use the domain corpus to enrich and finalise the ontology. Additionally, we propose a methodology for filtering and simplifying the ontology using hierarchical clustering and multi-way tree. Experimental results demonstrate the effectiveness of our approach in acquiring a cybersecurity ontology from specific domain data sources. Locust is implemented in Java and is available as an open source tool.

KEYWORDS

Cybersecurity,
Hierarchical Clustering,
Multi-Way Tree,
Ontology Engineering,
Ontology Learning.

DOI: 10.9781/ijimai.2026.6499

I. INTRODUCTION

In the realm of cybersecurity, the exchange of cyber threat intelligence information holds paramount importance. Ontology serves as an efficacious means to facilitate automated sharing of threat intelligence. Several existing ontologies or libraries, such as STIX¹, OpenIOC², and CVE³, contribute to this sharing process. However, the construction of high-quality cybersecurity ontologies encounters substantial methodological challenges in contemporary research. The primary limitations manifest in two critical aspects. First, existing ontology engineering methodologies predominantly emphasize the development of domain-independent ontologies, which fail to accommodate the domain-specific characteristics and requirements inherent in cybersecurity knowledge representation. Second, conventional manual approaches to ontology construction exhibit significant limitations regarding efficiency and consistency, as they are inherently susceptible to human subjectivity and require extensive temporal and human resources. Furthermore, the rapid evolution of cyber threats necessitates frequent updates to maintain the temporal validity of ontological representations, exacerbating the aforementioned limitations of current approaches.

Given these methodological constraints, there is a compelling need for an automated, domain-specific approach to cybersecurity ontology learning. Such a methodology must enhance not only the efficiency of the development process but also the rigorous semantic precision and comprehensive knowledge representation. This research presents a novel methodological framework that integrates hierarchical clustering algorithms with multi-way tree structures to facilitate automated cybersecurity ontology construction, thereby addressing the identified limitations in a systematic and scalable manner.

In this study, we propose an approach for learning domain ontology in the field of cybersecurity. Our approach utilizes hierarchical clustering and multi-way trees to automatically convert structured cybersecurity data sources and their corresponding domain corpus into a cybersecurity ontology. The choice of hierarchical clustering and multi-way trees is motivated by the hierarchical nature of concept relationships in ontology and the suitability of multi-way trees for representing ontology content. The domain corpus consists of textual sentences and a list of words indicating the sentence topics. Based on this corpus, we generate an ontology skeleton that incorporates the semantic information from structured data sources using reflection and divisive hierarchical clustering. Subsequently, we employ Natural Language Processing (NLP) tools and agglomerative hierarchical clustering to convert the domain corpus into a set of candidate concepts and their relations. These candidate concepts are

¹ <https://oasis-open.github.io/cti-documentation/stix/intro>

² <https://www.openioc.org/>

³ <https://cve.mitre.org/>

Please cite this article as:

Y. Wang, B. Zhao, X. Song, J. Zhu. A Practical Cybersecurity Ontology Generator Based on Hierarchical Clustering and Multi-Way Tree, International Journal of Interactive Multimedia and Artificial Intelligence, (2026), <http://doi.org/10.9781/ijimai.2026.6499>

then used to enrich the ontology skeleton, which is structured as a comprehensive multi-way tree. The resulting ontology is generated using the Jena⁴ framework, and the output includes a mapping from the multi-way tree. To facilitate this entire process, we have developed an ontology generator called Locust⁵. Locust is implemented in Java and is available as an open-source project under a liberal license. The ontology derived from Locust has the following characteristics: (i) comprehensive concept coverage by integrating diverse cybersecurity datasets and domain corpora; (ii) hierarchical completeness for representation of complex relationships; (iii) extensibility enabled by our automated approach, facilitating efficient addition of new security concepts and relationships. These characteristics enhance the robustness and practicality of our cybersecurity ontology, supporting automated threat intelligence sharing and security application development effectively.

In our experiment, we utilized six freely available domain specifications as structured inputs for the Locust tool: CAPEC⁶, CCE⁷, CEE⁸, CPE⁹, MAEC¹⁰, and STIX. Additionally, we incorporated over 150 textual domain corpora along with their corresponding summary words indicating text's topic as unstructured inputs. We meticulously traced the ontology learning process and fine-tuned the approach's thresholds. The proposed approach combines hierarchical clustering with multi-way trees to process cybersecurity domain knowledge. This combination is designed to capture the hierarchical nature of cybersecurity concepts while maintaining their semantic relationships in the domain. The method aims to generate a comprehensive and consistent ontology that effectively represents cybersecurity knowledge. To the best of our knowledge, this is the inaugural endeavor in developing an ontology learning tool specifically tailored for the field of cybersecurity.

The paper presents the following contributions:

- We propose and develop a method for generating cybersecurity ontologies, which is a flexible and comprehensive approach based on hierarchical clustering and multi-way trees. This method can effectively integrate diverse sources of information and is specifically designed for the cybersecurity domain. The resulting ontology has an enhanced concept coverage, better ontology structure, better scalability and maintainability.
- We have implemented the proposed approach as a tool called Locust and conducted extensive experiments to evaluate its effectiveness. The tool supports automated processing of heterogeneous data sources, and features flexible architecture that supports easy extension and customization.
- Locust has been made freely available as an open-source tool under a liberal license. This is the first initiative specifically focused on cybersecurity ontology learning.

The paper is structured as follows: Section II provides the background and motivation for this proposal, introducing general-purpose methods for ontology learning and specific sub-field ontologies. In Section III, we elaborate on the approach used to automatically construct the target ontology from diverse and heterogeneous data sources. Subsequently, Section IV demonstrates the proposed approach through experiments conducted on six distinct datasets. Lastly, Section V concludes the paper.

⁴ <https://jena.apache.org>

⁵ <https://gitee.com/yixuan94/ontology-generator-tony>

⁶ <https://capec.mitre.org/>

⁷ <https://ncp.nist.gov/cce>

⁸ <https://cee.mitre.org/>

⁹ <https://cpe.mitre.org/>

¹⁰ <https://maecproject.github.io/>

II. RELATED WORKS

In this section, we present the fundamental concepts pertaining to the proposed research. Firstly, we introduce highly qualified cybersecurity ontologies that are specialized in the sub-domain of cybersecurity or serve a general purpose. Subsequently, we provide an overview of the current research status in ontology learning.

A. Cybersecurity Ontology

Cybersecurity ontology can be broadly classified into two categories: (i) general-purpose cybersecurity domain ontology, and (ii) specific sub-domain ontology. The general-purpose cybersecurity ontology aims to encompass all concepts and instances relevant to cybersecurity, resulting in a comprehensive and extensive framework. An exemplary instance of a general-purpose cybersecurity ontology is the Unified Cybersecurity Ontology (UCO) [1]. UCO effectively incorporates and integrates diverse and heterogeneous data and schemas, proving to be a valuable asset in establishing a cybersecurity knowledge graph. It encompasses essential cybersecurity concepts such as "Means," "Consequences," "Attack," "Attacker," "AttackPattern," "Exploit," and "Exploit Target," all of which play pivotal roles in cybersecurity.

There exist numerous ontologies that specialize in specific sub-domains of cybersecurity. For instance, the System Security Assurance Ontology (SSAO) [2] is a modular ontology designed to facilitate network security situation awareness and support situation assessment. Similarly, the Security Core Ontology (SECCO) and an ontological framework named CRATELO [3] have been developed by the cybersecurity research alliance to address network security situation awareness. However, SECCO's concepts and categories lack the necessary granularity to capture detailed domain-specific scenarios. In order to enhance risk information gathering in cyber-physical systems, a more detailed cybersecurity ontology [4] has been proposed. To effectively comprehend malware threat intelligence, researchers have introduced an ontology named MALOnt [5]. MALOnt is derived from numerous annotated malware threat reports and enables the analysis, detection, and classification of threats originating from malware. The Kill Chain ontology [6] presents a model that describes the various phases of network intrusions, aiding in the understanding of the iterative nature of intelligence gathering and forming the basis for intelligence-driven computer network defense. The Insider Threat Indicator Ontology (ITIO) [7] has been developed as a standardized method for expressing potential indicators of malicious insider activity. The Measurement Ontology for IP traffic (MOI) [8] offers a high-level structured description of the interface and data exchange using IP traffic measurement devices. Additionally, SEPSES [9] is a knowledge graph that utilizes modular ontologies to integrate resources for cybersecurity.

B. Ontology Learning

Due to the time-consuming and labor-intensive nature of manual ontology development, there is a growing interest in automating the process. This automated process is known as ontology learning. Typically, ontology development consists of five phases: (i) concept extraction, (ii) concept relationship extraction, (iii) data properties and object properties extraction, (iv) axiom extraction, and (v) instance generation. Among these phases, axiom extraction is particularly challenging to automate and prone to potential errors. Furthermore, as knowledge graphs continue to evolve, ontologies increasingly serve as the schema layer for knowledge graphs, while other technologies like graph embedding are utilized for instance generation. Consequently, existing research primarily focuses on the first three phases of ontology development, such as glossary extraction [10]–[12] and concept relationship extraction [13], [14]. However, there is a lack of studies that specifically address ontology learning within a particular domain and that effectively combine structured data and unstructured corpus simultaneously.

In [15], the authors introduced CyberRel, a comprehensive model for identifying security concepts through joint entity and relation extraction. This research utilizes the Bidirectional Encoder Representation from Transformer (BERT) model to generate word vectors and employs a combination of Bidirectional Gating Recurrent Unit (BiGRU) and an attention mechanism to extract relevant features. The results are then decoded using a combination of BiGRU and Conditional Random Field (CRF). However, it primarily focuses on extracting triples from textual data to construct the desired cybersecurity knowledge graph, overlooking the inclusion of structured data and the ontology schema.

In [16], the authors introduced Chowlk, a tool designed to facilitate the conversion of UML-based ontology conceptualizations into OWL. This tool represents a significant endeavor in the development of ontologies from structured data. The authors present a comprehensive framework for mapping the ontology conceptualization graph to OWL. To validate the accuracy of the results produced by Chowlk, they conducted transformations of the visual OWL constructs depicted in the visual notation. However, similar to prior research, a notable limitation of their proposal is its dependency on high-quality Unified Modelling Language (UML) models as input.

In addition to UML, relational databases (RDB) play a crucial role as a structured input for ontology learning due to their foundation in relational algebra and high level of formalization. Numerous studies [17]–[25] have focused on converting RDB to ontologies without any loss of semantic information. Specifically, studies [17]–[19], [23], [24] involve a semi-automatic approach that requires human involvement in the ontology generation process. Mapping RDB to ontologies has emerged as a well-established field.

Several studies have been conducted on the processing of unstructured documents in the field. In [26], the authors presented an innovative approach to ontology learning from text that allows for the estimation of document similarity and identification of specific words or terms. Another study [27] focused on concept extraction and introduced a domain time relevance metric to identify relevant concepts. Additionally, [28] aimed to enhance the efficiency of ontology algorithms and proposed a partial multi-dividing ontology learning algorithm. The results demonstrated the effectiveness of this approach in optimizing the partial multi-dividing ontology learning model. In [29], the authors proposed two algorithms for learning terminological ontologies and compared them with two existing methods for learning concept hierarchies. In [30], a coreference resolution method is applied to implement data-driven ontology building, which uses a large biomedical corpus to derive coreference chains. Also, study [31] proposes a novel method based on Finite-state transducers to construct multilingual ontologies. In [32], the authors developed an ontology of adversary tactics by analyzing existing high-quality knowledge bases and integrating the parsed information from Cyber Threat Intelligence (CTI) reports.

Based on this review, it has been observed that none of these approaches provides a comprehensive coverage of the entire cybersecurity ontology development process. Additionally, the reviewed approaches focus primarily on structured data or text corpus, rarely taking both into account. Therefore, this paper proposes an automatic construction of the target cybersecurity ontology by combining structured data and text corpus. In this approach, structured data is utilized for cyber threat intelligence sharing to establish the ontology skeleton using a multi-way tree. Subsequently, the ontology skeleton is enriched by incorporating concepts and their relationships. Finally, the resulting target cybersecurity ontology is output in OWL DL format.

This section outlines the primary contributions of our research. Firstly, we provide a comprehensive overview of our ontology learning approach's architecture. Subsequently, we delve into the inner modules of this architecture, namely: (i) the reflection of structured classes; (ii) the methodology for constructing the ontology skeleton using divisive hierarchical clustering and multi-way tree; and (iii) the technique for enhancing the ontology through agglomerative hierarchical clustering.

A. Preliminaries and Problem Definition

In the field of cybersecurity, ontology plays a crucial role as a knowledge representation tool. The terms used in cybersecurity possess precise semantic relationships, which makes it necessary for the ontology to be constructed by domain experts who possess extensive knowledge in this area. However, it is uncommon for cybersecurity experts to also possess expertise in ontology development. Consequently, an automated method for ontology development is needed to expedite the process. Additionally, since ontology development is a time-consuming endeavor, human effort is indispensable in determining the final version of the ontology.

Our approach processes two types of input: structured cybersecurity specifications and a set of unstructured text documents $D = \{d_1, \dots, d_n\}$ that are related to cybersecurity. Each document d_i has a set of related topic words or phrases $W_i = \{W_{i1}, W_{i2}, \dots, W_{ik}\}$, $k \geq 1$. The goal is to construct the target ontology using both specifications and D . Firstly, we use specifications to construct the ontology skeleton and extract relevant concept information from D to update and enrich the ontology skeleton. Our proposed method, Locust, assumes that all the documents serve the cybersecurity topic.

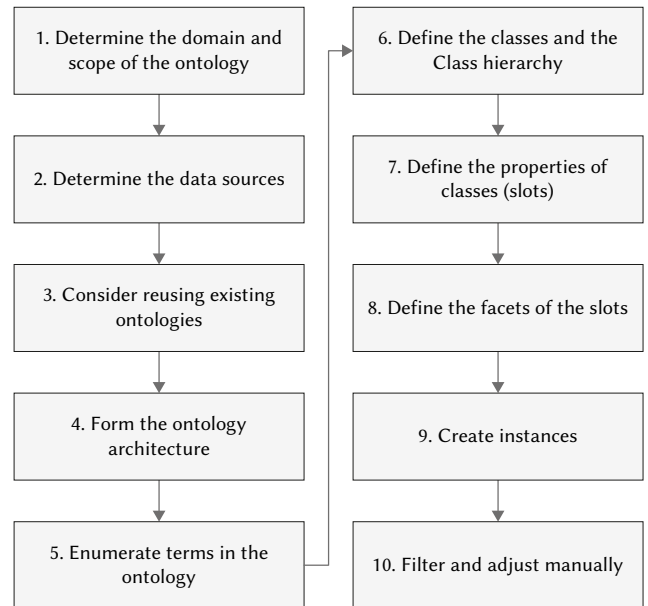


Fig. 1. Customized ontology development process.

To facilitate the automated development of ontologies, we employ a customized development methodology, as depicted in Fig. 1. In contrast to the traditional Stanford seven-step method, we omit the "propose competency questions" process and introduce two new steps: step 4 and step 10. The exclusion of competency questions is justified as it is a labor-intensive step that relies on requirement analysis, making it challenging to implement during ontology learning. Moreover, our focus lies in the ontology's architecture, specifically the ontology skeleton, hence the inclusion of step 4. Recognizing

that the automatically generated ontology output is prone to errors and unsuitable for direct practical application, step 10 becomes indispensable in ensuring the ontology’s quality. Overall, steps 2 to 9 can be executed automatically.

B. Overview

The primary contribution of this research is the development of an ontology learning approach specifically tailored to the field of cybersecurity. Within this context, a significant contribution of this study is the integration of cybersecurity specifications and the fusion of textual corpora with their corresponding topic terms. The outcomes of the automated ontology construction process serve as the foundation for the ultimate ontology. It is important to note that ontology learning is an iterative procedure, requiring multiple iterations before the ontology becomes both practical and applicable.

Fig. 2 illustrates an overview of the workflow employed by Locust. The proposed method employs a multi-way tree structure for ontology representation and consists of two preprocessing phases. The primary objective of the first phase is to initialize the multi-way tree by creating an empty node and subsequently pruning the tree to eliminate duplicate concepts. As part of this initial phase, we employ specific format conversion tools (such as the XJC¹¹ tool) to map structured specifications to Java classes. This approach allows us to preserve the semantic content of the original specifications to the greatest extent possible. By leveraging the class reflection mechanism, the converted classes can be further mapped and utilized for multi-way tree construction. Subsequently, concepts are disambiguated using divisive hierarchical clustering. In the second phase, a set of candidate concepts is filtered from the unstructured corpus, retaining only those that are relevant to the given topic. The remaining candidate concepts serve as contextual information for determining hierarchical relationships among the concepts. These hierarchical relationships among candidate concepts are then integrated into the ontology tree.

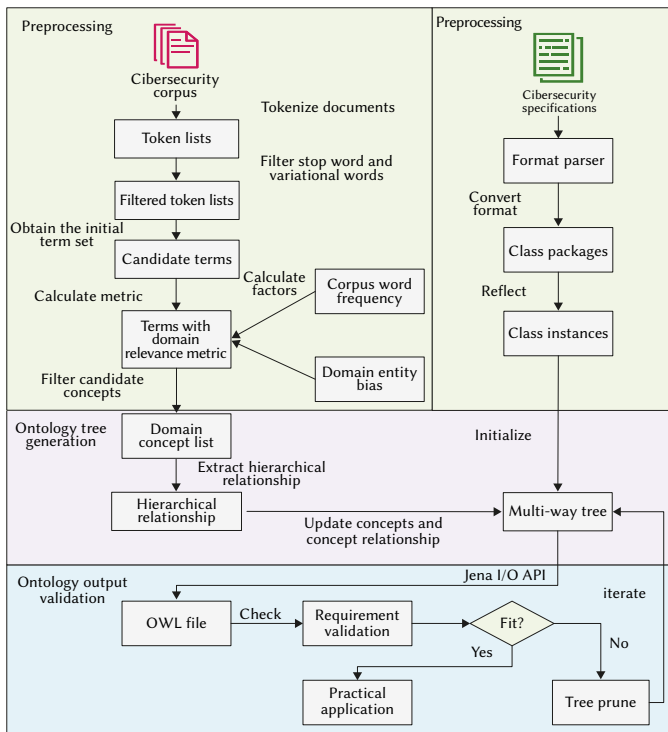


Fig. 2. Workflow of Locust to generate target ontology from a set of cybersecurity corpus and specifications.

¹¹ <https://docs.oracle.com/javase/8/docs/technotes/tools/unix/xjc.html>

C. Preprocessing and Generating Candidate Concepts

For structured cybersecurity specifications, schemas are converted to class files while preserving their complete path names. It is assumed that the class name corresponds to a class in the ontology, and the elements within the class correspond to elements in the target ontology. The inclusion of the full path names for the classes signifies the relative path of the ontology concept. By applying the mapping principle from class files to ontology nodes, we can derive the framework of the target ontology.

The unstructured cybersecurity *D* domain-specific documents *d_i* comprises Each domain document *d_i* has a corresponding summary word set $K_i = \{w_1, w_2, \dots, w_n\}$, indicating its title and content. The process begins with tokenizing the corpus into n-grams, followed by stop-word filtering to generate initial candidate concepts. Then, calculate the domain consistency and domain relevance metric [33] of each candidate concept. Discard those terms whose metric is lower than the predefined threshold. This heuristic assumes that cybersecurity domain concepts typically exhibit domain relevance metrics. high remaining concepts is The relationship between the obtained clustering method. using a hierarchical After this process, all remaining concepts are organized into a whole concept tree. Then take this tree as the material to enrich the ontology tree.

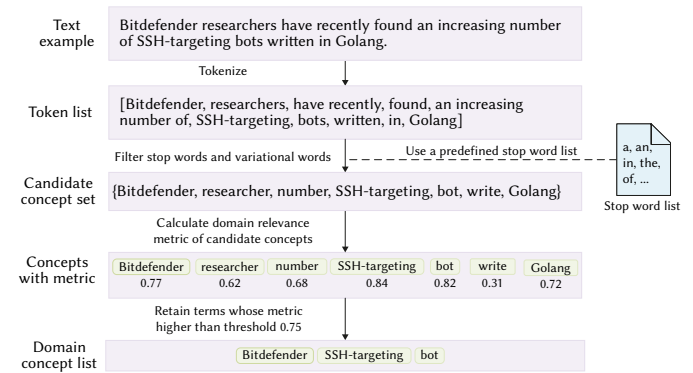


Fig. 3. Filtering process for candidate concepts from an example document.

Fig. 3 illustrates the text filtering process using a botnet-related example. Within the list of candidate concepts, terms with a metric below the threshold of 0.75 are eliminated. The remaining terms form the concept list for this particular example. Several factors influencing the threshold are listed below.

- **Corpus quality:** The quality of the corpus is crucial because low-quality texts may introduce noise, affecting the selection of the threshold. In certain fields, the frequency of specific terminology can be high, influencing the selection of the relevance threshold.
- **Application requirements:** Certain applications may demand higher precision to ensure that selected concepts are relevant to the domain. In the medical field, incorrect concept selection could lead to severe consequences, necessitating higher thresholds. In some applications, it may be essential to cover as many concepts within the domain as possible, which could result in a reduction of the threshold to ensure that more relevant concepts are included. Algorithm 1 Divisive hierarchical clustering for disambiguating concepts.
- **Domain complexity:** There may be an overlap between specific and general terms in some fields, complicating the selection of relevance thresholds. A careful analysis of term usage is necessary to avoid including irrelevant concepts.

The selection process involves a trade-off between coverage of domain concepts and the avoidance of redundant concepts. On one hand, a lower threshold may enhance the coverage of relevant

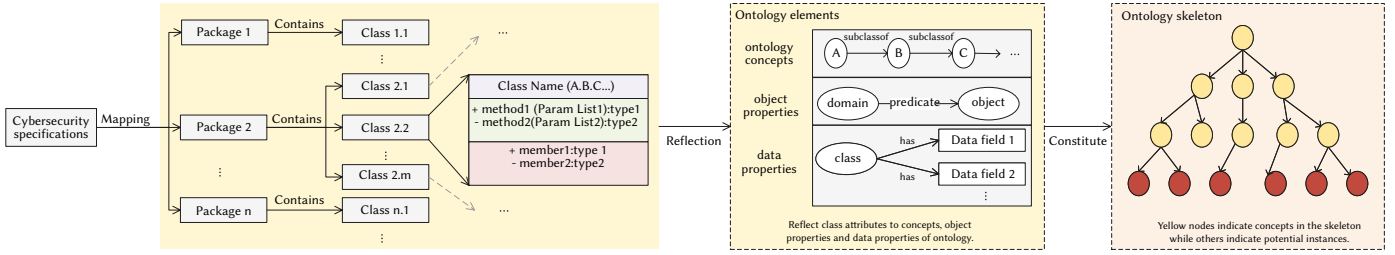


Fig. 4. Constituting the ontology skeleton with structured cybersecurity specifications with Java packages and reflections.

Algorithm 1. Divisive hierarchical clustering for disambiguating concepts

Require: An ontology skeleton multi-way tree T , a concept similarity matrix list L , split threshold ϵ

Ensure: All clusters out of M

```

1: results ← []
2: C ← extractDuplicatedConcepts(T)
3: for all c in C do
4:   Mc ← getMatrix(L, c)
5:   iterator ← Mc.iterator
6:   while iterator.hasNext() do
7:     curVal ← iterator.next()
8:     if curVal < ε then
9:       results.add(record(coordinates))
10:    end if
11:  end while
12: end for
13: return results

```

concepts, ensuring that a broader range of domain-specific knowledge is captured. On the other hand, this approach may lead to the inclusion of redundant or less relevant concepts, which can dilute the overall quality of the results. Therefore, it is essential to find an optimal balance that maximizes the inclusion of relevant concepts while minimizing redundancy, thereby enhancing the effectiveness of the domain analysis. In our approach, we use the valid terms and noise terms to support selecting the threshold. The detailed selection process is discussed in Section IV(C).

D. Constituting Ontology Skeleton

As illustrated in Section III(B), the initialization of the multi-way tree occurs prior to the enrichment process, and it is based on structured cybersecurity specifications. The constitution of the ontology skeleton involves two stages: (i) converting the data format to Java class packages, and (ii) reflecting the class instances of the target ontology. The process of ontology skeleton constitution is elaborated in Fig. 4, providing a detailed depiction.

The first stage involves data format conversion using tools such as xjc. The results of this mapping process are organized into multiple Java packages, each containing several Java files that represent classes. Fig. 4 illustrates these classes as class diagrams, providing their full names, method members, and data members. The class’s full name is represented as "A.B.C..." and can be divided into a node path within the target ontology. We consider these classes to be meaningful concepts within the target ontology. The data members, being attached to the class and having explicit data types, can be converted into data properties. Similarly, the method members, which absorb parameters

and have explicit return values, can be converted into object properties. The parameters’ data types and the return value represent specific Java classes or elementary data types. Therefore, the parameters define the domain, while the return value defines the range of object properties. Each class in a package corresponds to a concept node in the target ontology. By converting all the classes, we establish the basic ontology skeleton, which is represented as a multi-way tree.

The target ontology is depicted as a multi-way tree, as illustrated in Fig. 5 and Fig. 6. The concept nodes of the multi-way tree are of the "OntMultiwayTreeNode" type. This representation encompasses all the attributes of the ontology classes. Moreover, converting the multi-way tree representation into an OWL-formatted ontology file can be easily accomplished using Jena through a recursive approach.

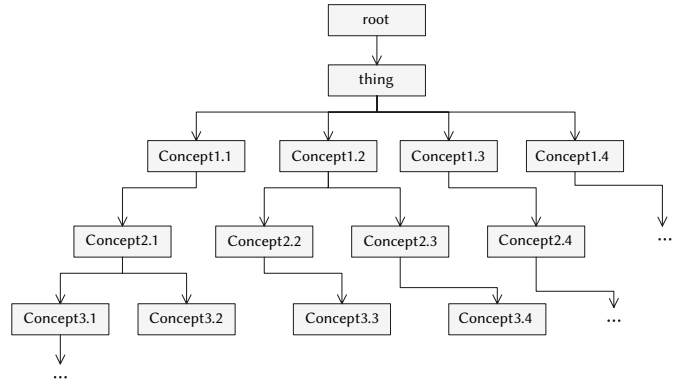


Fig. 5. Ontology multi-way tree example.

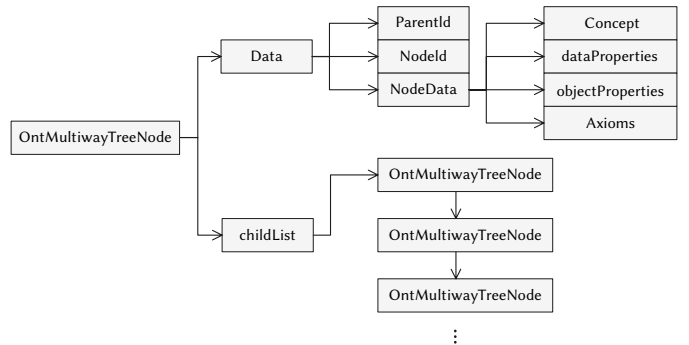


Fig. 6. An example "OntMultiwayTreeNode" instance of multi-way tree.

E. Filtering and Disambiguating Concepts

Since different cybersecurity specifications may have the same concepts, we want to disambiguate the concepts in the target ontology multi-way tree, i.e., to prune the subtrees whose root nodes have the same concept name. For example, the concept “activity” may come from STIX or CEE, but they have different paths and are different nodes in the multi-way tree. In this case, these two concept nodes would be merged. Otherwise, both nodes would be retained. We disambiguate these concept nodes by comparing their semantic attributes, including

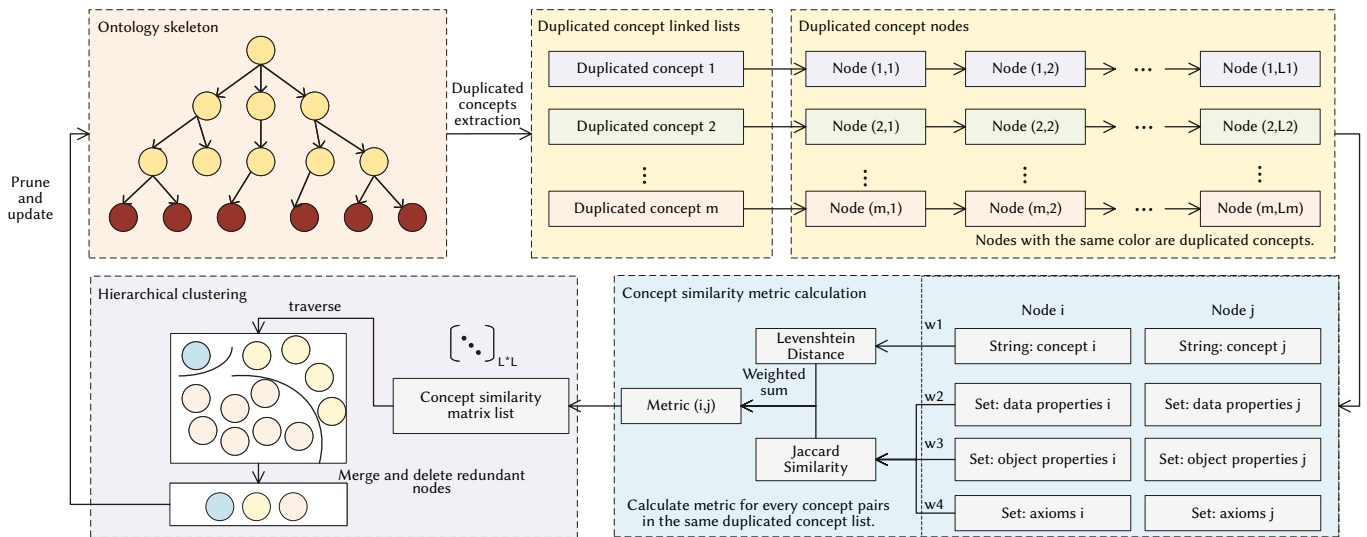


Fig. 7. Disambiguating concepts in the ontology skeleton.

data properties and object properties. Equation 1 refers to the concept similarity score (CSS):

$$CSS_{ij} = \sum_{k=1}^3 w_k \times m_{ij} \quad (1)$$

where w_k is the weight of the k^{th} attribute while m_{ij} indicates the Jaccard similarity of the attributes. As Fig. 7 depicts, during the whole disambiguating period, we first extract all the concept nodes with the same concept string from the ontology skeleton. Then, we initialize a similarity matrix M_t , where t is the number of nodes with the same concept string. After that, calculate and fill the matrix. Set a threshold to filter the concept pairs, only concept pairs whose similarity metric is lower than the threshold are remained. The selection of this threshold is discussed in Section IV(C).

The process of determining which concepts are retained employs a divisive hierarchical clustering method. Our customized method, presented in Algorithm 1, takes the concept similarity matrix as input and produces all concept clusters as output. Subsequently, we merge and eliminate redundant nodes from the ontology skeleton. By applying this method to divide all duplicated concepts, the phase of filtering and disambiguating concepts concludes.

F. Extracting Candidate Concepts and Relationships

Since the initial ontology skeleton has been established, we can now enhance the ontology by incorporating the domain-specific corpus. To achieve this objective, we need to extract potential concepts and their inherent relationships from the corpus. This process consists of two primary phases: (i) generation of candidate concepts and (ii) extraction of hierarchical relationships. In the first phase, the text is processed, resulting in a list of concepts corresponding to each text. The second phase utilizes the previous results as input and employs a customized hierarchical clustering method. Fig. 8 provides a detailed illustration of these two phases. The cybersecurity corpus is utilized to construct a concept map, where the candidate concepts serve as keys and the summary words of each text, representing the article titles, serve as corresponding values. By calculating the semantic distances between each pair of concepts, we can derive a concept correlation matrix from the concept map. Each element in the concept correlation matrix represents the semantic distance between two concepts. This matrix plays a crucial role in the extraction of hierarchical relationships.

Algorithm 2 outlines our customized approach for extracting hierarchical relationships from the corpus. It introduces a concept

class with three attributes: (i) index x , (ii) index y , and (iii) group identity. The algorithm follows an agglomerative policy to determine concept relationships, where the concept distance plays a crucial role in deciding which two concepts should be merged. Whenever two concepts are selected, a new concept node is created to serve as their parent. The algorithm continues until all concepts have been merged and the resulting hierarchical relationship is outputted as a file named "tree.json". This JSON file encompasses the entire clustering process and is subsequently utilized for ontology enrichment.

Algorithm 2. Agglomerative hierarchical clustering for concept relationship extraction.

Require: M : concept map, α : merge ratio
Ensure: Sorted concept relationships

- 1: $L \leftarrow \text{loadConceptMap}(M)$ {concept list}
- 2: $D \leftarrow \text{getConceptDistanceMap}()$
- 3: $C \leftarrow [\text{Concept}() \text{ for } _ \text{ in range}(\text{len}(L))]$
- 4: $U \leftarrow \{i : 1 \text{ for } i \text{ in range}(\text{len}(C))\}$
- 5: $idx \leftarrow 0$
- 6: **for** $(k, _)$ **in** D **do**
- 7: $l, h \leftarrow \text{int}(k.\text{split}(\#))$ {low/high index}
- 8: **if** $C[l].\text{group} \neq C[h].\text{group}$ **then**
- 9: $g_l \leftarrow C[l].\text{group}; g_h \leftarrow C[h].\text{group}$
- 10: $U[g_l] += U[g_h]$
- 11: **for** c **in** C **do**
- 12: **if** $c.\text{group} = g_h$ **then**
- 13: $c.\text{group} \leftarrow g_l$
- 14: **end if**
- 15: **end for**
- 16: **end if**
- 17: **if** $|U| \leq |C| \times \alpha$ **then**
- 18: **break**
- 19: **end if**
- 20: **end for**
- 21: $S \leftarrow \text{sort}(U.\text{items}(), \text{by value, desc})$
- 22: **return** S

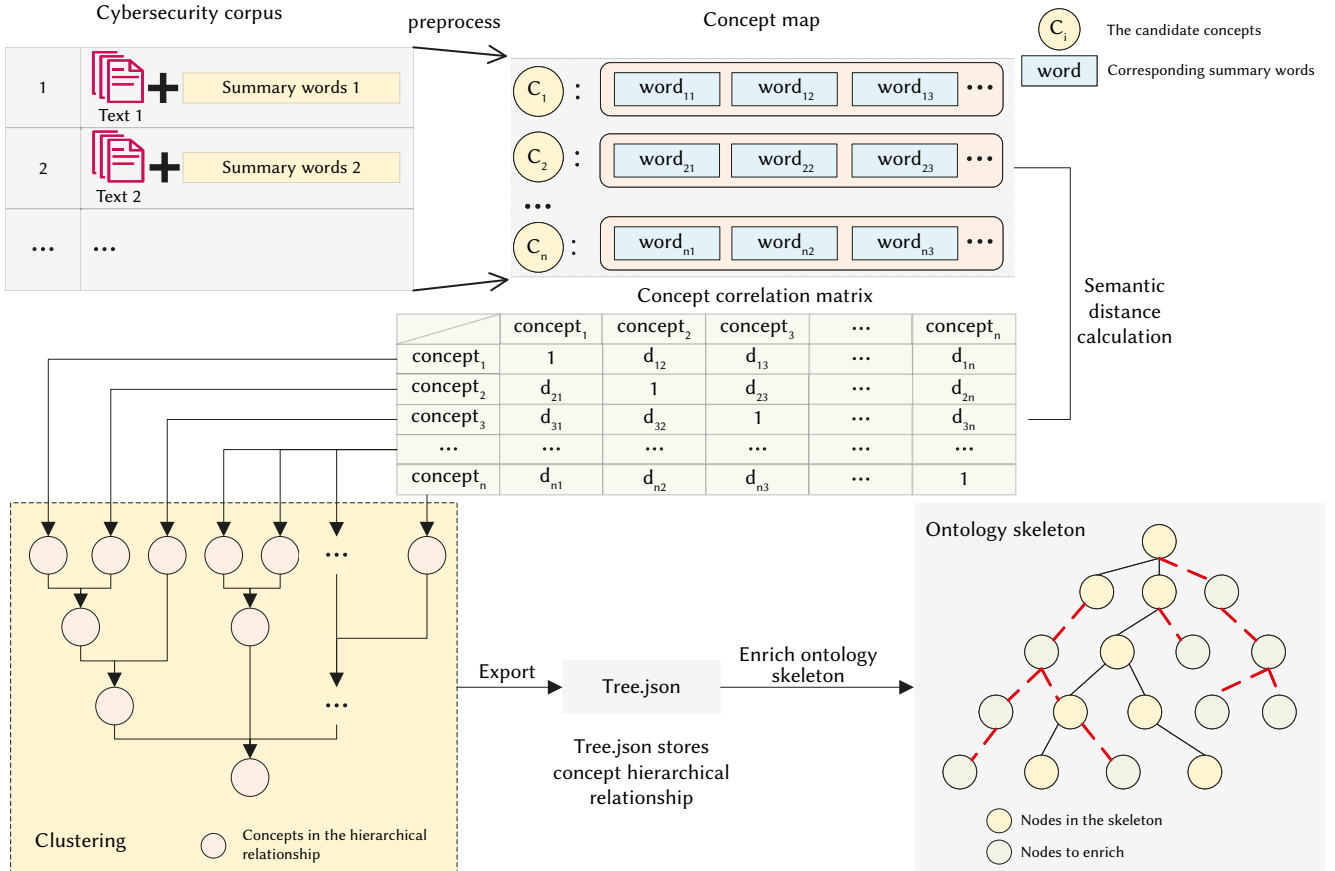


Fig. 8. Extracting concept relationship and enriching ontology skeleton.

G. Enriching and Serializing the Resulting Ontology

In order to enhance the ontology skeleton presented in Section III(D), we utilize the JSON file discussed in Section III(F) as the input data. Given that we assume the concept relationship has been extracted from a corpus in the field of cybersecurity, it plays a crucial role in constructing the target ontology. As depicted in Fig. 9, the enrichment process adheres to the principle of "merge or add". If a concept already exists in the ontology skeleton, it is simply discarded. Conversely, if the concept is new, along with its sub-concepts, it is added as a new branch in the multi-way tree. Additionally, if the concept is new but its parent exists in the ontology skeleton, it is added as a child node of its corresponding parent. This entire process is carried out iteratively by traversing the nodes in the hierarchical structure of concepts. Once all the nodes have been traversed and checked for enrichment in the multi-way tree, all the requirements for serializing the multi-way tree to the target ontology are fulfilled.

In order to obtain the ultimate ontology, it is necessary to parse the multi-way tree and serialize the ontology. To accomplish this, we make use of the model factory mechanism provided by the Jena library. Firstly, an instance of the ontology model is initialized. Subsequently, the tree is traversed and the nodes along with their corresponding relationships are added to the model instance. Lastly, the resulting ontology is outputted as a "*.owl" file, representing the desired target ontology.

H. Manual Filtering and Adjustment

After the automated ontology generation and enrichment process, additional manual review and adjustments are required to ensure the quality of the ontology by experts in the cybersecurity domain. Manual filter and adjustment primarily focus on semantic validation

and structural refinement. To complete semantic validation, we need to review the concept relationships and hierarchical structures. Also, to complete structural refinement, concept organization and relationships may be adjusted and optimized.

The process is completed through the protégé tool [34], which can open the resulting ontology and visualize it. The visualization page can show the class hierarchical relationship clearly. In addition, protégé allows users to complete a further revision of the ontology.

IV. EXPERIMENTS

In our evaluation, we focus on three key aspects. Firstly, we quantify the performance of our proposed method. Subsequently, we assess the structure and practical application of the generated ontology. Lastly, we explore the robustness and adaptiveness of our method by varying both the input corpus and the parameter w , which governs the termination of the clustering process. This investigation holds significance for the practical implementation of our method, as it allows us to establish default parameter values for these crucial algorithms.

A. Datasets

Due to the fact that there is no specific unstructured document dataset on cybersecurity, we evaluate our method on a newly created dataset, SD. This dataset is specifically tailored to our scenario of generating an ontology from a set of unstructured documents. Also, we take CAPEC, CCE, CEE, CPE, MAEC, and STIX as the structured cybersecurity input of our method.

Common Attack Pattern Enumeration and Classification (CAPEC) is a comprehensive collection of attack patterns and techniques. It

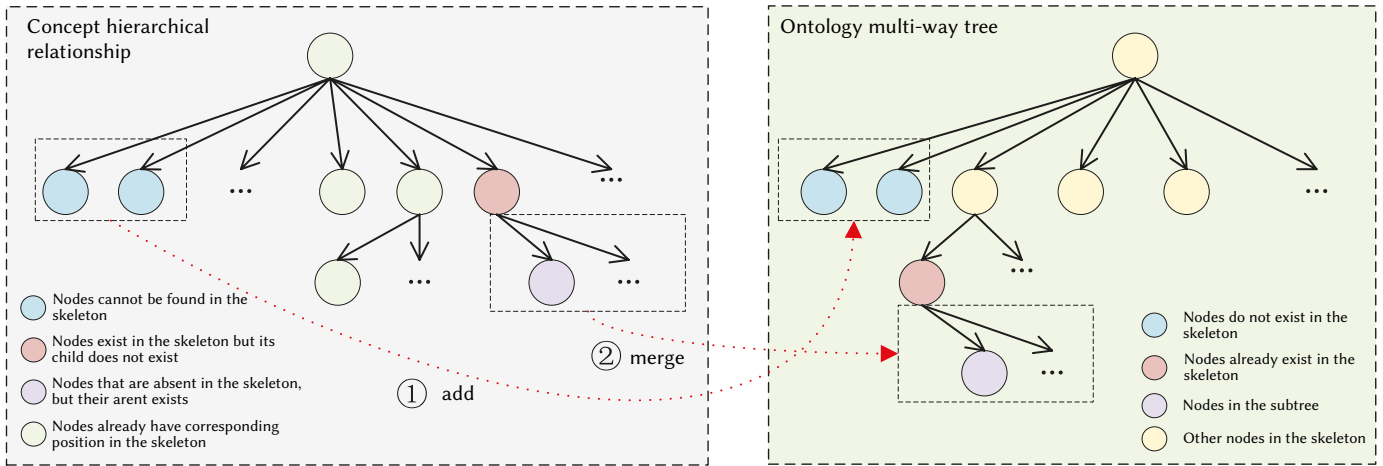


Fig. 9. Enriching ontology multi-way tree.

provides a structured and standardized taxonomy for describing and categorizing common attack methods used by adversaries. The core schema of CAPEC is represented as a set of XSD files and is open source.

Common Configuration Enumeration (CCE) is a standardized schema that assigns unique identifiers to system configuration issues. Its primary function is to facilitate rapid and precise correlation of configuration data across diverse information sources and tools.

Common Event Expression (CEE) is a standardized collection of event log data. Its purpose is to enhance the sharing and analysis of cybersecurity events across different systems. CEE adheres to a uniform format and schema, enabling analysts and researchers to readily compare and correlate events from diverse sources.

Common Platform Enumeration (CPE) is a standardized dataset used to identify software and hardware products. It offers a structured format for describing the attributes of a particular product. This allows for consistent and comprehensive product identification and classification.

Malware Attribute Enumeration and Characterization (MAEC) is a standardized language and schema used for describing and sharing information regarding malware and malicious software. This framework offers a structured approach to capture and represent the attributes, behaviors, and characteristics of malware.

Structured Threat Information eXpression (STIX) is a standardized language and framework utilized for the description and dissemination of cybersecurity threat intelligence. Its purpose is to facilitate the exchange and analysis of cyber threats among organizations in a structured and uniform manner.

We have developed the SD dataset to assess the effectiveness of our approach in the field of cybersecurity. This dataset consists of 150 research articles that are specifically focused on cybersecurity. To construct this dataset, we enlisted the expertise of a dedicated cybersecurity team. Each article in the dataset is accompanied by a set of summary words that capture the main themes addressed in the article. For instance, if an article discusses the workings of computer viruses, its summary words may include "virus" and "attack". It is important to note that all the summary words in the dataset were determined through unanimous agreement among the members of the expert team.

B. Baseline Methods and Evaluation Metrics

We take several state of the art cybersecurity ontologies as baselines in our practical experiments. Two frameworks are used to make the comparison: (i) the ability to extract ontology elements, (ii) the structure of the target ontology.

As the evaluation of ontology learning systems remains an ongoing challenge, we adopt the framework proposed by Shamsfard et al. [35] to assess our approach. The evaluation encompasses two key aspects: (i) the learning methodology employed, and (ii) the resultant ontology.

First, we evaluate the ability of capturing basic ontology elements, which are: (i) concepts/classes, (ii) hierarchical relationships, (iii) non-hierarchical relationships, (iv) axioms, (v) instances, (vi) input type, (vii) output syntax, and (viii) input preprocessing.

Shamsfard's framework highlights the unreliability of evaluating learning methods solely based on accuracy comparisons. This is due to the variations in domains, inputs, and backgrounds across different ontology learning systems. To address this, we employ cross-evaluation techniques to compare domain ontologies, ensuring a more comprehensive and robust evaluation.

Also, we take three metrics [36] to evaluate the structure of the resulting ontology, which are: (i) *RR* (Relationship Richness), (ii) *AR* (Attribute Richness), and (iii) *LR* (Link Richness). Equations 2, 3, and 4 indicate the computation of *RR*, *AR*, and *LR* metrics, respectively.

$$RR = \frac{|P|}{|SC| + |P|} \quad (2)$$

$$AR = \frac{|att|}{|C|} \quad (3)$$

$$LR = \frac{|R|}{|C|} \quad (4)$$

In these equations, $|P|$, $|SC|$, $|att|$, $|C|$, and $|R|$ indicates the number of relations, subclasses, attributes, concepts, and triples, respectively.

C. Dataset Preprocessing and Experimental Settings

In terms of the dataset preprocessing, we extract or download the available schema files and then convert them into Java packages. After that, integrate these packages into the whole program. To extract candidate concepts from unstructured documents, we exclude the common English stop word list from NLTK corpus library. Throughout our experiments, we set the domain relevance threshold mentioned in Section III(C) as 0.75 and the split threshold ϵ in Algorithm 1 as 0.8. As is stated earlier, the split threshold is to measure the semantic distance of duplicated concepts. Choosing an appropriate split threshold can better help distinguishing these concepts. In addition, no matter what merge ratio we determine in Algorithm 2, the concept coverage rate is 100%. We choose the merge ratio as 0.99 since a higher merge ratio contributes to more branches and fewer redundant concepts. The reasons for selecting thresholds are listed below.

TABLE I: DOMAIN RELEVANCE THRESHOLD SELECTION METRICS

Threshold	Precision	Recall	F1-score	Valid terms	Noise terms	Terms remained
0.10	0.6198	1.0000	0.7653	3942	2418	6360
0.15	0.6198	1.0000	0.7653	3942	2418	6360
0.20	0.6198	1.0000	0.7653	3942	2418	6360
0.25	0.6198	1.0000	0.7653	3942	2418	6360
0.30	0.6198	1.0000	0.7653	3942	2418	6360
0.35	0.6198	1.0000	0.7653	3942	2418	6360
0.40	0.6198	1.0000	0.7653	3942	2418	6360
0.45	0.6198	1.0000	0.7653	3942	2418	6360
0.50	0.6200	1.0000	0.7654	3942	2416	6358
0.55	0.6225	0.9949	0.7659	3922	2378	6300
0.60	0.6305	0.9934	0.7714	3916	2295	6211
0.65	0.6415	0.9926	0.7793	3913	2187	6100
0.70	0.6504	0.9881	0.7844	3895	2094	5989
0.75	0.7992	0.9645	0.8741	3802	955	4757
0.80	0.7918	0.4691	0.5831	1849	486	2335
0.85	0.9925	0.1687	0.2884	665	5	670
0.90	0.9944	0.0901	0.1652	355	2	357

TABLE II. HIERARCHICAL CLUSTERING SPLITTING THRESHOLD METRICS

Threshold	Retained clusters	Average similarity	Precision	Recall	F1-score	Clusters split	Wrong split
0.40	1015	0.8963	1.0000	0.0769	0.1429	1	0
0.45	1009	0.8991	1.0000	0.5833	0.7368	7	0
0.50	1005	0.9008	1.0000	0.8462	0.9167	11	0
0.55	1005	0.9008	1.0000	0.8462	0.9167	11	0
0.60	1005	0.9008	1.0000	0.8462	0.9167	11	0
0.65	1005	0.9008	1.0000	0.8462	0.9167	11	0
0.70	1005	0.9008	1.0000	0.8462	0.9167	11	0
0.75	1005	0.9008	1.0000	0.8462	0.9167	11	0
0.80	1004	0.9224	1.0000	0.9231	0.9600	12	1
0.85	823	0.9231	0.0673	1.0000	0.1262	193	180
0.90	474	0.9434	0.0250	1.0000	0.0468	542	529
0.95	173	0.9920	0.0154	1.0000	0.0304	843	830

1. Domain Relevance Threshold Selection

During the period of preprocessing and generating candidate concepts, a domain relevance threshold is used to filter terms. To determine the optimal threshold value, we conducted empirical experiments with different threshold values on our cybersecurity corpus. The data presented in the Table I illustrates the relationship between various threshold values and their corresponding performance metrics, including Precision, Recall, F1-Score, Valid Terms, Noise Terms, and Terms Remained. This table lists a range of threshold values from 0.1 to 0.9. For each threshold, we observe the corresponding Precision, Recall, and F1-Score. These metrics provide insight into the effectiveness of the selected terms in relation to their relevance to the domain. Precision measures the proportion of valid terms among the total terms identified. As the threshold increases, we notice a gradual improvement in Precision, particularly beyond the threshold of 0.7, where it reaches a peak of 0.9944 at 0.9. This indicates that higher thresholds lead to a more accurate selection of relevant terms. Recall, on the other hand, reflects the ability to identify all relevant terms within the corpus. The Recall values fluctuate significantly, with a notable drop at higher thresholds (e.g., 0.85 and 0.9), suggesting that while fewer terms are selected, the proportion of relevant terms decreases. The F1-score, which balances Precision and Recall, is crucial for determining the optimal threshold. The highest F1-score of 0.8741 occurs at a threshold of 0.75, indicating a favorable balance between the number of relevant terms identified and the accuracy of those terms. This suggests that a threshold around 0.75 may be optimal

for maintaining a good trade-off between coverage and redundancy. The table also distinguishes between valid terms and noise terms. As the threshold increases, the number of valid terms decreases, while the number of noise (2) terms also reduces significantly at higher thresholds. This reduction in noise terms is beneficial, as it indicates a cleaner selection of relevant concepts.

Based on the analysis of the items in Table I, we conclude that a threshold of approximately 0.75 offers the best compromise. It maximizes the identification of relevant (4) terms while minimizing redundancy and noise, thus enhancing the overall quality of the concept extraction process.

2. Hierarchical Clustering Split Threshold Selection

To determine the optimal split threshold value in hierarchical clustering, we conducted experiments with threshold values from 0.4 to 0.95. The experiment results are shown in Table II. For each threshold, we observe the number of retained clusters, average similarity, and performance metrics such as precision, recall, and F1-score. These metrics provide insight into the effectiveness of the clustering process at different thresholds. These metrics provide insight into the effectiveness of the clustering process at different thresholds. As the splitting threshold increases, the number of retained clusters generally decreases. These trends indicate that higher thresholds lead to more significant merging of clusters, which can simplify the clustering structure but may also risk losing important distinctions between data points. The average similarity of

TABLE III. ABILITY TO EXTRACT ONTOLOGY ELEMENTS

References	Classes (Concepts)	Hierarchical relations	Non-hierarchical relations	Axioms	Instances
[17]	✓	✓	✓	✓	✓
[18]	✓	✓	✓	✓	✓
[19]	✓	✓	✓	✓	✗
[20]	✓	✓	✓	✓	✓
[21]	✓	✗	✓	✓	✗
[22]	✓	✓	✓	✓	✗
[23]	✓	✗	✓	✓	✓
[24]	✓	✓	✓	✓	✓
[25]	✓	✓	✓	✓	✓
[26]	✓	✓	✗	✓	✓
[27]	✓	✓	✓	✓	✗
[28]	✓	✓	✗	✓	✗
[29]	✓	✓	✗	✓	✗
[30]	✓	✓	✗	✓	✓
[31]	✓	✓	✓	✓	✓
[32]	✓	✓	✓	✓	✗
Our approach	✓	✓	✓	✓	✓

TABLE IV. STARTING POINT AND RESULTS OF ONTOLOGY LEARNING SYSTEMS

References	Prior knowledge	Automation degree	Structured data	Unstructured data	Output
[17]	Base RDB	Semiautomatic	✓	✗	OWL
[18]	Base RDB	Semiautomatic	✓	✗	OWL
[19]	Base RDB	Semiautomatic	✓	✗	OWL
[20]	Base RDB	Automatic	✓	✗	OWL
[21]	Base RDB	Semiautomatic	✓	✗	OWL
[22]	Base RDB	Automatic	✓	✗	OWL
[23]	Base RDB	Semiautomatic	✓	✗	OWL
[24]	Base RDB	Semiautomatic	✓	✗	OWL
[25]	Base RDB	Automatic	✓	✗	OWL
[26]		Semiautomatic	✗	✓	Intermediates
[27]		Semiautomatic	✗	✓	Intermediates
[28]	Base ontology	Semiautomatic	✓	✗	Intermediates
[29]		Semiautomatic	✗	✓	Intermediates
[30]		Automatic	✗	✓	Not mentioned
[31]		Semiautomatic	✗	✓	OWL
[32]	Base ontology	Semiautomatic	✗	✓	OWL
Our approach		Automatic	✓	✓	OWL

the clusters is an essential metric that reflects how closely related the items within each cluster are. The average similarity remains relative high across low thresholds, indicating that clusters are formed with closely related items. However, as the threshold increases, the average similarity also shows slight variations, suggesting that while clusters may become larger, their internal cohesion remains relatively stable. Precision measures the accuracy of the clusters in identifying relevant items. Higher thresholds bring lower precision, indicating that many irrelevant items are included in the clusters. As the threshold decreases, precision improves, reaching a maximum of 1 at thresholds of 0.8 and below, suggesting that the clusters become more accurate in representing relevant items. Recall remains consistently high when the threshold is larger than 0.5, indicating that the clustering process successfully most concept clusters that should be split. The F1-score, which balances Precision and Recall, is crucial for determining the optimal threshold. The highest F1-score of 0.9600 occurs at a threshold of 0.8, indicating an optimal balance between Precision and Recall.

Based on the analysis of items in Table II, we conclude that a threshold of approximately 0.8 offers the best compromise. This value maintains a reasonable number of clusters while maximizing precision, ensuring that most concept clusters are split correctly and

few clusters are split incorrectly. In summary, the selection of the split threshold for hierarchical clustering is a nuanced process that requires careful consideration of various performance metrics.

D. Results and Analysis

We applied our approach to the datasets we stated in Section IV(A). We analyzed the resulting ontology, which we call CSO (Cyber Security Ontology). As stated in Section IV(B), we evaluate our approach from the following two aspects.

1. Learning Method Evaluation

Table III illustrates the effectiveness of various approaches in extracting fundamental ontology elements. Most existing methods focus on specific aspects of ontology learning, such as omitting the extraction of non-hierarchical relationships. However, it is crucial to consider all ontology elements as they contain important semantic information. Therefore, it is essential to employ a comprehensive ontology learning method. Table IV presents the initial conditions and outcomes of these learning methods. Evaluating an ontology learning system requires considering factors such as prior knowledge and the degree of automation. Our proposed method is both independent of prior knowledge and fully automated. While many approaches

TABLE V. ONTOLOGY METRICS

Ontology	Class (Concept)	Class property	Axiom	Logical axiom	Declaration axiom	Subclass axiom
UCO	106	104	635	378	216	126
MALOnt	68	44	1405	654	375	46
Kill chain	13	5	64	37	19	26
ITIO	125	143	1252	643	289	126
SEPSES	52	228	1409	672	413	9
SSAO	217	129	1030	640	357	243
MOI	551	348	4796	3349	1291	977
CSO	8007	5433	207553	194113	13440	24640

TABLE VI. ONTOLOGY STRUCTURE EVALUATION

Ontology	P	SC	att	C	R	RR	AR	LR
UCO	308	126	277	196	755	0.709	1.413	3.852
MALOnt	137	46	985	68	1406	0.749	14.485	20.676
Kill chain	37	26	13	13	121	0.587	1.000	9.308
ITIO	581	126	815	125	1423	0.822	6.520	11.384
SEPSES	78	14	245	59	1461	0.848	4.153	24.763
SSAO	121	977	129	217	653	0.110	0.594	3.009
MOI	87	243	348	551	1624	0.264	0.632	2.947
CSO	27368	24640	169473	8007	207553	0.526	21.166	25.921

accept either structured data or unstructured text as input, existing methods that concentrate on converting RDB schemas can directly generate resulting ontologies in the owl specification. Conversely, methods that process unstructured documents tend to produce intermediate structures rather than the final ontologies. In contrast, our proposed approach encompasses all the listed ontology elements and accommodates both structured and unstructured input data. The output ontology is formatted in OWL 2 DL, which is the ontology language standard recommended by the W3C.

2. Resulting Ontology Evaluation

In this section, we compare our output ontology with several state-of-the-art (SOTA) cybersecurity ontologies, with a focus on evaluating their structure. These cybersecurity ontologies are mentioned in Section II(A), which include UCO, MALOnt, Kill chain, ITIO, SEPSES, SSAO and MOI. Descriptive information about the ontologies is presented in Table V. These statistics have been automatically calculated using the Protégé tool. The results show that our proposed method, which generates the Cybersecurity Ontology (CSO), has significantly larger ontology metrics. This is because CSO incorporates a large number of unstructured documents and captures ontology elements in fine granularity. In contrast, other ontologies have been manually constructed and carefully pruned.

These basic metrics only reflect a facet of the ontology and larger metrics do not equal higher quality, especially the class count. To further evaluate the structure of the ontologies, we perform structure evaluation using three metrics mentioned in Section IV(B): Relation Richness (RR), Attribute Richness (AR), and Linkage Richness (LR). The results of this evaluation are presented in Table VI. RR indicates the diversity of relations within an ontology, with a higher RR value indicating a greater variety of relationships. AR reflects the richness of attributes within an ontology, with more attributes generally indicating a greater amount of knowledge conveyed. LR measures the richness of relevance between concepts, with a low LR indicating sparsity.

The results show that CSO has relatively more attributes and is denser compared to the other ontologies. Compared with the manually constructed ontologies, CSO should be further pruned to exclude redundant concepts, which are generated during the agglomerative hierarchical clustering period. However, CSO exhibits

lower relationship richness, indicating that a higher percentage of its relations are class-subclass relations. Overall, CSO demonstrates a well-structured design.

3. Manual Filter and Adjust

While our proposed approach automatically extracts and constructs a cybersecurity domain ontology from both the structured data and unstructured corpus, ensuring its practical utility requires expert intervention. This ontology refinement is conducted by the protégé tool, where cybersecurity experts perform critical adjustments to enhance the ontology’s practicability. In this period, we invited four experienced cybersecurity experts from network security companies. Here are two main steps to complete filtering and adjusting.

The first step is instance data curation. Automatically generated ontologies contain data extracted from textual sources that may lack representativeness in actual security applications. In this step, expert review involves: (i) Systematically examining and removing automatically generated instance data, (ii) Retaining only conceptual structures with significant practical value; (iii) Ensuring the ontology reflects domain-specific knowledge rather than arbitrary textual artifacts.

The second step is concept hierarchy validation. Since our ontology construction primarily relies on text clustering and concept extraction, the hierarchical relationships require rigorous professional scrutiny. Cybersecurity experts meticulously examine inheritance and containment relationships between concepts.

Since the Locust tool takes the structured cybersecurity datasets as the skeleton of the resulting ontology and uses unstructured corpus to enrich the skeleton, it saves much time and labor. In our experiment, it only takes two days to complete the manual adjustment process. And the resulting ontology is actually utilized in the Security Information and Event Management (SIEM) system of the cybersecurity corporation.

4. Challenges With the Proposed Approach

Using structured and unstructured domain datasets for ontology learning, particularly in the proposed method, faces potential challenges when applied to broader or more diverse cybersecurity datasets. First, the quality of the datasets influences the quality of the resulting ontology. For an unstructured corpus, its domain relevance

directly leads to the result of domain-concept filtering. If the quality of the corpus is low, it may introduce noisy terms that can easily be confused with valid terms, thereby affecting the final concept extraction. Thus, a new fine-tuned domain relevance threshold and split threshold are needed to mitigate this problem. Secondly, as more structured datasets are inputted, the number of coreferential terms increases. These terms can consume a significant amount of time during the clustering process, potentially leading to reduced processing efficiency. Therefore, when applying this approach to a wider range of cybersecurity datasets, special attention must be paid to the quality of the datasets and processing efficiency to ensure the effectiveness and accuracy of ontology learning.

V. CONCLUSION AND FUTURE WORK

This study presents a methodology and introduces Locust, a tool designed for ontology extraction from cybersecurity specifications and corpora. The approach consists of two main components: (1) constructing a multi-way tree to outline the ontology framework based on structured specifications, and (2) enhancing the framework through the incorporation of contextual information extracted from relevant documents. Through experiments conducted on seven datasets, the proposed methodology yields a high-quality ontology that surpasses several well-established cybersecurity ontologies in terms of robustness and modeling precision.

One limitation of the proposed method is that it requires an input corpus consisting of a set of topic words. These topic words can be manually labeled and recorded, or extracted automatically. To address this issue, we intend to integrate our method with Latent Dirichlet Allocation (LDA) models, which facilitate automatic extraction of topic words. Furthermore, our future objective is to develop an interactive platform to refine the resulting ontology. Subsequently, we will incorporate the cybersecurity ontology learning method as the core module of the Network Security Situation Awareness (NSSA) system. Additionally, the cybersecurity ontology will be integrated into the model-driven engineering design of the NSSA system. Our aim is to utilize ontology learning to streamline and optimize the model-driven engineering process, making it more efficient and less labor-intensive.

CREDiT AUTHORSHIP CONTRIBUTION STATEMENT

Yixuan Wang: conceptualization, writing -original draft, writing -review and editing.

Bo Zhao: conceptualization, validation, formal analysis. Xiaofu Song: conceptualization, writing - review and editing.

Jiahui Zhu: investigation.

DATA STATEMENT

The data of this research are not publicly available due to the data protection policies of Beijing TOPSEC Technologies Science and Technology Inc.

DECLARATION OF CONFLICTS OF INTEREST

We have no conflict of interest to declare.

ACKNOWLEDGMENT

We would like to thank anonymous reviewers for their helpful suggestions and comments. This work is supported by the National Natural Science Foundation of China under grant 62572356 and the

Innovation Funding Plan by Beijing TOPSEC Technologies Science and Technology Inc.

REFERENCES

- [1] Z. Syed, A. Padia, T. Finin, M. L. Mathews, A. Joshi, "UCO: A unified cybersecurity ontology," Arizona, USA, 2016. [Online]. Available: <http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12574>
- [2] Y. Wang, B. Zhao, W. Li, L. Zhu, "An ontologycentric approach for network security situation awareness," in *47th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2023*, Torino, Italy, 2023, pp. 777–787, IEEE. <https://doi.org/10.1109/COMPSAC57700.2023.00107>
- [3] A. Oltramari, L. F. Cranor, R. J. Walls, P. D. McDaniel, "Building an ontology of cyber security," Fairfax VA, USA, 2014. [Online]. Available: https://ceur-ws.org/Vol-1304/STIDS2014_T08_OltramariEtAl.pdf
- [4] C. Grigoriadis, A. M. Berzovitis, I. Stelios, P. Kotzaniolaou, "A cybersecurity ontology to support risk information gathering in cyberphysical systems," in *Computer Security. ESORICS 2021 International Workshops - CyberICPS, SECPRE, ADIoT, SPOSE, CPS4CIP, and CDT&SECOMANE, Darmstadt, Germany, October 4-8, 2021, Revised Selected Papers*, vol. 13106 of Lecture Notes in Computer Science, 2021, pp. 23–39, Springer. https://doi.org/10.1007/978-3-030-95484-0_2
- [5] N. Rastogi, S. Dutta, M. J. Zaki, A. Gittens, C. C. Aggarwal, "Malont: An ontology for malware threat intelligence," 2020. [Online]. Available: <https://arxiv.org/abs/2006.11446>
- [6] E. M. Hutchins, M. J. Cloppert, R. M. Amin, et al., "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," 2011. [Online]. Available: https://www.ciosummits.com/media/solution_spotlight/LM_Cyber_Kill_Chain_White_paper_2011.pdf
- [7] D. Costa, M. Collins, S. J. Perl, M. Albrethsen, G. Silowash, D. Spooner, "An ontology for insider threat indicators: Development and application.," 2014. [Online]. Available: https://ceur-ws.org/Vol-1304/STIDS2014_T07_CostaEtAl.pdf
- [8] E. G. Specification, "Measurement ontology for ip traffic," European Telecommunications Standards Institute, 2013. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/moi/001_099/002/01.01.01_60/gs_moi002v010101p.pdf
- [9] E. Kiesling, A. Ekelhart, K. Kurniawan, F. J. Ekaputra, "The SEPSES knowledge graph: An integrated resource for cybersecurity," in *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, vol. 11779 of Lecture Notes in Computer Science, 2019, pp. 198–214, Springer. https://doi.org/10.1007/978-3-030-30796-7_13
- [10] Y. Park, R. J. Byrd, B. Boguraev, "Automatic glossary extraction: Beyond terminology identification," in *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, 2002*, pp. 1–7. <https://aclanthology.org/C02-1142/>
- [11] H. Zhong, Z. Ning, G. Li, Z. Li, "A method of core concept extraction based on semantic-weight ranking," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 1, 2022. <https://doi.org/10.1002/cpe.6504>
- [12] S. Fang, Z. Huang, M. He, S. Tong, X. Huang, Y. Liu, J. Huang, Q. Liu, "Guided attention network for concept extraction," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 2021*, pp. 1449–1455, ijcai.org. <https://doi.org/10.24963/ijcai.2021/200>
- [13] S. Gul, S. Rübiger, Y. Saygin, "Context-based extraction of concepts from unstructured textual documents," *Information Sciences*, vol. 588, pp. 248–264, 2022. <https://doi.org/10.1016/j.ins.2021.12.056>
- [14] A. Lopes, J. L. Carbonera, D. Schmidt, L. F. Garcia, F. H. Rodrigues, M. Abel, "Using terms and informal definitions to classify domain entities into top-level ontology concepts: An approach based on language models," *Knowledge-Based Systems*, vol. 265, p. 110385, 2023. <https://doi.org/10.1016/j.knsys.2023.110385>
- [15] Y. Guo, Z. Liu, C. Huang, J. Liu, W. Jing, Z. Wang, Y. Wang, "Cyberrel: Joint entity and relation extraction for cybersecurity concepts," in *Information and Communications Security - 23rd International Conference, ICICS 2021, Proceedings, Part I*, vol. 12918 of Lecture Notes in Computer Science, Chongqing, China, 2021, pp. 447–463, Springer. https://doi.org/10.1007/978-3-030-64633-0_27

- org/10.1007/978-3-030-86890-1_25
- [16] S. Chávez-Feria, R. García-Castro, M. PovedaVillalón, “Chowlk: from uml-based ontology conceptualizations to OWL,” in *The Semantic Web - 19th International Conference, ESWC 2022, Proceedings*, vol. 13261 of *Lecture Notes in Computer Science*, Hersonissos, Crete, Greece, 2022, pp. 338–352, Springer. https://doi.org/10.1007/978-3-031-06981-9_20
- [17] C.-h. Liao, Y.-f. Wu, G.-h. King, “Research on learning OWL ontology from relational database,” vol. 1176, no. 2, p. 022031, 2019. <https://doi.org/10.1088/1742-6596/1176/2/022031>
- [18] M. A. Hazber, R. Li, X. Gu, G. Xu, “Integration mapping rules: Transforming relational database to semantic web ontology,” *Applied Mathematics Information Sciences*, vol. 10, no. 3, pp. 1–21, 2016. <http://dx.doi.org/10.18576/amis/100307>
- [19] M. Dadjoo, E. Kheirkhah, “An approach for transforming of relational databases to OWL ontology,” 2015. [Online]. Available: <http://arxiv.org/abs/1502.05844>.
- [20] M. A. G. Hazber, R. Li, Y. Zhang, G. Xu, “An approach for mapping relational database into ontology,” in *12th Web Information System and Application Conference, WISA 2015*, Jinan, China, 2015, pp. 120–125, IEEE Computer Society. <https://doi.org/10.1109/WISA.2015.25>
- [21] M. A. G. Hazber, R. Li, X. Gu, G. Xu, Y. Li, “Semantic SPARQL query in a relational database based on ontology construction,” in *11th International Conference on Semantics, Knowledge and Grids, SKG 2015*, Beijing, China, 2015, pp. 25–32, IEEE Computer Society. <https://doi.org/10.1109/SKG.2015.14>
- [22] E. Jiménez-Ruiz, E. Kharlamov, D. Zheleznyakov, I. Horrocks, C. Pinkel, M. G. Skjæveland, E. Thorstensen, J. Mora, “Bootox: Practical mapping of rdbs to OWL 2,” in *The Semantic Web - ISWC 2015 14th International Semantic Web Conference, Proceedings, Part II*, vol. 9367 of *Lecture Notes in Computer Science*, Bethlehem, PA, USA, 2015, pp. 113–132, Springer. https://doi.org/10.1007/978-3-319-25010-6_7
- [23] M. A. G. Hazber, B. Li, G. Xu, M. A. S. Moseh, X. Gu, Y. Li, “An approach for generation of SPARQL query from SQL algebra based transformation rules of RDB to ontology,” *Journal of Software*, vol. 13, no. 11, pp. 573–599, 2018. <https://doi.org/10.17706/jsw.13.11.573-599>.
- [24] H. Tissot, C. A. G. Huve, L. M. Peres, M. D. D. Fabro, “Exploring logical and hierarchical information to map relational databases into ontologies,” *International Journal of Metadata, Semantics and Ontologies*, vol. 13, no. 3, pp. 191–208, 2019. <https://doi.org/10.1504/IJMSO.2019.099834>
- [25] T. Naz, M. Shuja, S. K. Shahzad, M. Atif, “Fully automatic OWL generator from rdb schema,” *International Journal of Advanced and Applied Sciences*, vol. 5, no. 4, pp. 79–86, 2018. <https://doi.org/10.21833/ijaas.2018.04.010>
- [26] A. Tissaoui, S. Sassi, R. Chbeir, A. Mechergui, “A top-down enriching approach for ontology learning from text,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 19, 2022. <https://doi.org/10.1002/cpe.7036>
- [27] F. N. AL-Aswadi, H. Y. Chan, K. H. Gan, et al., “Enhancing relevant concepts extraction for ontology learning using domain time relevance,” *Information Processing and Management*, vol. 60, no. 1, p. 103140, 2023. <https://doi.org/10.1016/j.ipm.2022.103140>
- [28] W. Gao, J. L. G. Guirao, B. Basavanagoud, J. Wu, “Partial multi-dividing ontology learning algorithm,” *Information Sciences*, vol. 467, pp. 35–58, 2018. <https://doi.org/10.1016/j.ins.2018.07.049>
- [29] W. Wang, P. M. Barnaghi, A. Bargiela, “Probabilistic topic models for learning terminological ontologies,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 1028–1040, 2010. <https://doi.org/10.1109/TKDE.2009.122>
- [30] S. Ashury-Tahan, A. D. N. Cohen, N. Cohen, Y. Louzoun, Y. Goldberg, “Data-driven coreferencebased ontology building,” in *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, 2024, pp. 14290–14300, Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.findings-emnlp.834>
- [31] F. B. Mesmia, M. Mouhoub, “Semi-automatic building and learning of a multilingual ontology,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 11, pp. 242:1–242:19, 2023. <https://doi.org/10.1145/3615864>
- [32] C. Huang, P. Huang, Y. Kuo, G. Wong, Y. Huang, Y. S. Sun, M. C. Chen, “Building cybersecurity ontology for understanding and reasoning adversary tactics and techniques,” in *IEEE International Conference on Big Data, Big Data 2022, Osaka, Japan, December 17-20, 2022*, 2022, pp. 4266–4274, IEEE. <https://doi.org/10.1109/BigData55660.2022.10021134>
- [33] P. Velardi, M. Missikoff, R. Basili, “Identification of relevant terms to support the construction of domain ontologies,” 2001. [Online]. Available: <https://aclanthology.org/W01-1005.pdf>
- [34] M. A. Musen, “The protégé project: a look back and a look forward,” *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015. <https://doi.org/10.1145/2757001.2757003>
- [35] M. Shamsfard, A. A. Barforoush, “The state of the art in ontology learning: a framework for comparison,” *Knowledge Engineering Review*, vol. 18, no. 4, pp. 293–316, 2003. <https://doi.org/10.1017/S0269888903000687>
- [36] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth, B. Aleman-Meza, “Ontoqa: Metric-based ontology quality analysis,” 2005. [Online]. Available: <https://corescholar.libraries.wright.edu/knoesis/660>



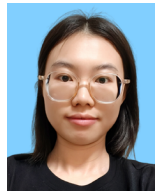
Yixuan Wan

Yixuan Wang received his M.S. degree in Computer Science from the University of Electronic Science and Technology of China. After graduation, he worked as a software engineer in Nanjing Research Institute of Electronic Engineering. Now, he is a Ph.D candidate in the School of CyberScience and Engineering, Wuhan University. His research interests include semantic web and network security.



Bo Zhao

Dr. Zhao received his PhD degree in Computer Science from Wuhan University, China. Now he is a professor in School of Cyber Science and Engineering, a member of China Cryptography Society and a senior member of China Computer Society. His current research interests include trusted computing, system security and network security. As the project team leader, he has successfully completed many research projects of high quality, including the projects sponsored by the National Science Fund of China. He has published over 100 journal and conference papers as the first or corresponding author. He is authorized more than 30 patents by the State Intellectual Property Office of China. Also, he has published the book titled Trusted Computing, and it has been adopted as a textbook by many universities.



Xiaofu Song

Xiaofu Song received her M.S. degree in Computer Science from the Central China Normal University and worked as a security engineer in China Telecommunication Corporation after graduation. Now, she is currently a Ph.D candidate in the the School of Cyber Science and Engineering, Wuhan University. Her research interests include intrusion detection and network security.



Jiahui Zhu

Jiahui Zhu received the M.S. degree in Computer Science from the University of Electronic Science and Technology of China. He is currently pursuing the Ph.D. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include big data and data mining.