CSKB: A Cyber Security Knowledge Base Based on Knowledge Graph

Kun Li, Huachun Zhou(✉), Zhe Tu, and Bohao Feng

School of Electronic and Information Engineering,
Beijing Jiaotong University, Beijing 100044, China
{kun_li,hchzhou,zhe_tu,bhfeng}@bjtu.edu.cn

Abstract. The access of massive terminal devices has brought new security risks to the existing Internet, so traditional cybersecurity data sets are difficult to reflect the modern and complex network attack environment. Therefore, how to realize the standardization and integration of cybersecurity data, so as to continuously store and update malicious traffic information under massively connected terminals, has become a critical issue to be solved urgently. Therefore, based on the knowledge graph, we built a standardized cybersecurity ontology, and introduced the implementation process of the cybersecurity knowledge base (CSKB) from five stages of knowledge acquisition, knowledge fusion/extraction, know-ledge storage, knowledge inference, and knowledge update, aiming at providing a reliable basis for real-time cybersecurity protection solutions. Experiments prove that the knowledge stored in CSKB can effectively realize the specification and integration of security data.

Keywords: Cyber security data · Knowledge graph · Security ontology · Cyber security knowledge base

1 Introduction

With the rapid development of 5G communication technology, the access of massive terminal devices has brought new security risks to the existing Internet, which in turn threatens user's privacy protection and impacts the security of critical information infrastructure [1, 2]. In the field of cybersecurity, although a series of cybersecurity data sets have been designed, such as KDDCup99 [3], NSL-KDD [4], UNSW-NB15 [5], and CICDDoS2019 [6], etc. They are stored in a CSV file in the form of a two-dimensional table, designed to reflect modern and complex attack environments by designing a comprehensive data set containing normal and abnormal behavior, but they still have some shortcomings: Firstly, cybersecurity data sets capture and analyze traffic in the form of data packets, and put all the characteristics of traffic into data rows, so that they lose the clear relationship between cyber entities and various features. It is difficult to achieve logical preservation of existing data only through data sets; Secondly, each security data set uses its own rules to count traffic and design feature values, resulting

S. Yu et al. (Eds.): SPDE 2020, CCIS 1268, pp. 100–113, 2020.
https://doi.org/10.1007/978-981-15-9129-7_8

in a lack of effective correlation with each other, which hinders data mining and knowledge extraction; Finally, the security data set is collected and analyzed under a specific network environment. When faced with traffic information from multiple sources, the data set cannot be updated and expanded regarding the original rules. Therefore, how to effectively use a large amount of existing knowledge and historical accumulation in the field of cybersecurity to achieve the specification and integration of security data, to continuously store and update malicious traffic information under massively connected terminals, has become a critical issue to be solved urgently [7].

On the other hand, in the past decade, research on the construction of knowledge graphs has developed rapidly. As a new knowledge representation method, the knowledge graph represents the relationship between entities in the form of nodes and edges. The efficient query ability, flexible storage mechanism, and update ability of knowledge graph are favored by security researchers [8]. The endless network of threats and the great progress of knowledge graphs have prompted academia to consider how to use knowledge graphs to describe network attack traffic. Among them, related work mainly focuses on attack source traceback [9, 10], which can effectively query and find the evidence and location left by the attack to attribute the source of the attack. However, attack graphs based on specific network environments do not always take into account the dynamic nature of the modern network, especially Distributed Denial of Service (DDoS) attacks [11], so they always lack awareness and classification of malicious attacks on the network. These methods are difficult to meet the needs of the attack and defense parties to quickly and accurately assess the attack success rate and attack revenue [12].

Therefore, we focus on how to build a cybersecurity knowledge base (CSKB) based on the knowledge graph to reflect the modern complex attack environment. The CSKB continuously updates the cybersecurity knowledge through the real-time monitoring system, so as to continuously store and update malicious traffic information under massively connected terminals and achieve network situation awareness and dynamic defense. Specifically, we designed a standardized cybersecurity ontology regarding multi-source security data sets and cybersecurity knowledge, which uniformly describes security element information and implements the function of integrating multi-source and heterogeneous network threat data. Then, we propose a CSKB construction framework based on knowledge graphs and introduce the implementation process of the CSKB from five stages: knowledge acquisition, knowledge fusion/extraction, knowledge storage, knowledge inference, and knowledge update. In particular, we propose a path ranking algorithm *TransFeature* combined with deep learning to achieve knowledge reasoning. Finally, we used the graph database Neo4j to store knowledge in the field of cybersecurity based on the cybersecurity ontology, thereby constructing the CSKB, and showing the comparative analysis between the knowledge in CSKB and various data sets.

The rest of the paper is organized as follows: In Sect. 2, we discuss the related work. In Sect. 3, we explain the construction of cybersecurity ontology. Then, we introduce the construction framework of CSKB based on the knowledge graph and use the graph database Neo4j [13] to store knowledge in the field of cybersecurity in Sect. 4. In Sect. 5, We show a comparative analysis between the knowledge in CSKB and various data sets. At last, Sect. 6 summarizes the paper and future work.

2 Related Work

Recently, many studies have focused on the construction of cybersecurity ontology. Feng et al. [14] focus on Loc/ID split network architectures and provide a related comprehensive survey on their principles, mechanisms, and characteristics. In order to solve the problem of mining and evaluating security information in multi-source heterogeneous networks existing in the Internet of Things (IoT), Xu et al. [15] proposed an IoT cybersecurity situation awareness model based on semantic ontology and user-defined rules. Ontology technology can provide a unified and formal description to solve the problem of semantic heterogeneity in the field of IoT security. Islam et al. [16] analyzed the complexity of integrating safety software systems into safety coordination platforms, and then proposed an ontology-driven method for safety orchestration platforms to automate safety system integration processes. However, the above works only build a general framework for security entities and do not give detailed and standardized cybersecurity ontology construction. This paper refers to various types and characteristics of network attacks to establish a cybersecurity ontology that manages cybersecurity entities at a semantic level.

In addition, several works use knowledge graphs to describe and store modern network attack traffic information. Based on only a limited number of computers and routers involved in the attack session, Yu et al. [9] propose a novel mark-on-demand (MOD) traceability scheme based on the DPM mechanism. Zhu et al. [10] proposed a network attack attribution framework and constructed an air-ground cybersecurity knowledge graph for tracking the source of attacks in the air-ground integrated information network. However, the schemes of these attack graphs do not always consider the dynamic nature of modern networks, so it is difficult to reflect the modern complex attack environment. Our CSKB provides a basis for network situational awareness and dynamic defense by integrating multi-source and heterogeneous security data.

This paper designs a standardized cybersecurity ontology based on multi-source and heterogeneous security data and implements a CSKB by combining knowledge graphs. The CSKB continuously updates the cybersecurity knowledge through the real-time monitoring system, so as to continuously store and update malicious traffic information under massively connected terminals and achieve network situation awareness and dynamic defense.

3 Cyber Security Ontology Construction

Ontology is a set of terms used to describe a field. Its organizational structure is hierarchically structured and can be used as the skeleton and foundation of a knowledge base. Therefore, the goal of building a cybersecurity ontology is to acquire, describe, and represent knowledge in the field of cybersecurity, and to provide a common understanding of cybersecurity knowledge. By determining the commonly recognized terms, we finally give a clear definition of the relationship between concepts or entities from different levels of formal models.

3.1 Process of Cyber Security Ontology Construction

The ontology construction method can be roughly classified as top-down and bottom-up ones. According to the knowledge structure of cyber malicious attacks, we propose a top-down approach to constructing a cybersecurity ontology, as shown in Fig. 1, which aims to describe the types and characteristics of modern network attacks as comprehensively as possible at the semantic level.

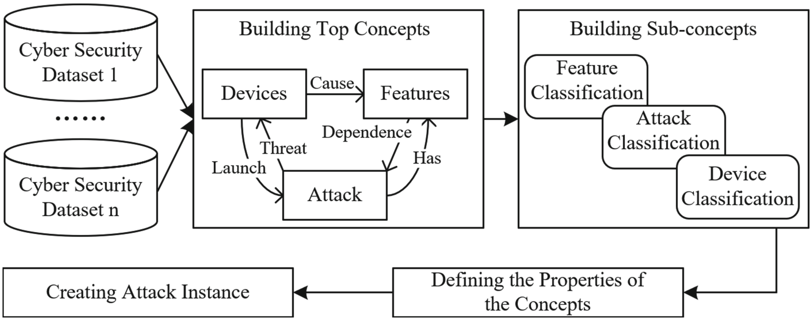

Fig. 1. The process of cybersecurity ontology construction

In order to solve the problem of discrete and independent multi-source heterogeneous cybersecurity data, we collect security data sets that have been widely used in the field of cybersecurity and refer to their experimental environment. First, we constructed the top concepts required by the network attack knowledge conceptual model, including three ontology: device, attack, and feature, and established the relationship between the top concepts. Then, based on the top concepts, we further construct sub-concepts of network attack knowledge. We classify each top concept and provide a detailed description, and describe the internal relationship between each concept by defining object attributes. Finally, by mapping the formatted security data to the ontology model, we add the generated instance to the network attack ontology to describe the relationship between cybersecurity entities.

3.2 Description of Cyber Security Ontology

In this section, we outline the concepts of each layer in the cybersecurity ontology and define the relationship between them.

Top Concepts. As shown in Fig. 1, the top concept contains three ontology: Device, Attack and Feature, and five relationships: Launch, Cause, Threat, Has, and Dependence.

The concept of "Device" represents various physical entities or hardware/software/operating systems from a modern network environment. It may be the source and target of a network attack, as well as the source of attack features.

$$Device \subseteq \forall launch\ Attack\ \cup\ \forall cause\ Feature \tag{1}$$

"Attack" is the core concept in cybersecurity ontology and represents a variety of malicious behaviors in modern networks. It poses a serious threat to devices in the modern network environment. At the same time, different network attacks also have their traffic features.

$$Attack \subseteq \forall threat\ Device \cup \forall has\ Feature \tag{2}$$

The concept of "Feature" represents the essential features that attack traffic must possess. Different attack types determine different feature values.

$$Feature \subseteq \forall dependence\ Attack \tag{3}$$

Sub-concepts and Entities. Based on the above three top-level concepts, we expand the category of each concept to further expand the scale of the cybersecurity ontology and achieve a comprehensive and detailed cyber security ontology construction. Since the concepts of upper and lower layers are subordinate relations, we define the relationship attribute as "Has".

We first introduce the classification of device-based sub-concept, as shown in Table 1. Device may be the source and target of modern network attacks, so it should include all hardware, software, and operating systems that may be subjected to or launched attacks. Since the device is a mature ontology, its instantiation has been uniformly described. Therefore, we directly give the cyber security entity corresponding to the device sub-concept based on the experience from the field of cyber security.

Table 1. Device-based sub-concepts

Id	Sub-concepts	Entities
1	Hardware	PC, Mobile device, IXIA etc.
2	Software	Malicious software
3	Operating System	Win7, Win8, Win10, Linux etc.

Next, we analyze the sub-concept classification based on network attacks. The purpose of building the knowledge base is to reflect the modern complex and changing attack environment. Therefore, a comprehensive and meticulous classification of the concept of network attacks, so as to deal with malicious attacks in a targeted manner, is of great significance for achieving network situation awareness and dynamic defense. Unlike device-based sub-concept classification, some network attacks can achieve deeper classification according to their characteristics, especially DDoS attack. As shown in Table 1, referring to multi-source and heterogeneous network attack data, we divide modern network attacks into 8 seed concepts, and also classify each sub-concept in detail, so as to cover the various attack types that appear in modern networks as much as possible.

Table 2. Attack-based sub-concepts

Id	Sub-concepts-1	Sub-concepts-2 (Entities)
1	Fuzzers	FTP Fuzz, Web Fuzz
2	Backdoors	Add root, Sniff user passwords
3	Exploits	SQL injection, Cross-site scripting, Weak password
4	Analysis	Port scan, Spam, Html files penetrations
5	Worms	E-mail, P2P, Vulnerability, Search engine
6	Shellcode	None
7	Reconnaissance	Data collation attack, Sniffing/scanning
8	DDoS	PortMap, NetBIOS, LDAP, MSSQL, UDP, SYN, UDP-Lag, NTP, DNS, SNMP, SSDP, Web, TFTP

By further classifying the sub-concepts of attacks, the types of attacks contained in the underlying concepts directly correspond to the attack entities (Table 2).

Finally, we introduce the classification of feature-based sub-concepts. Different security data sets use different feature extraction tools and lack effective correlation with each other. Therefore, we define and classify feature ontology to realize the specification and integration of multi-source and heterogeneous security data. In order to maintain the scale of the cyber security ontology and ensure a high efficiency of querying the knowledge base, we use the Pearson coefficient to calculate the correlation of each feature value in the data sets, such as NSL-KDD, UNSW-NB15, and CICDDoS2019. Finally, we selected the basic five-tuple features and the five most relevant features as sub-concepts, and explained each feature sub-concept, as shown in Table 3.

Table 3. Feature-based sub-concepts

Id	Sub-concepts	Introduction
1	srcip	Source IP address
2	sport	Source port number
3	dstip	Destination IP address
4	dsport	Destination port number
5	proto	Transaction protocol
6	sbytes	Source to destination bytes
7	sttl	Source to destination time to live
8	sloss	Source packets retransmitted or dropped
9	service	http, ftp, ssh, dns, etc.
10	spkts	Source to destination packet count

3.3 Cyber Security Ontology Implementation

Through the construction of a top-down cyber security ontology, we have established a cyber security ontology that can reflect the types and characteristics of modern network attacks, as shown in Fig. 2. Each node in the ontology represents a concept or entity in cyber security. When the level of the node becomes deeper, the semantics of the entity becomes more specific, but the abstraction of the entity also decreases.

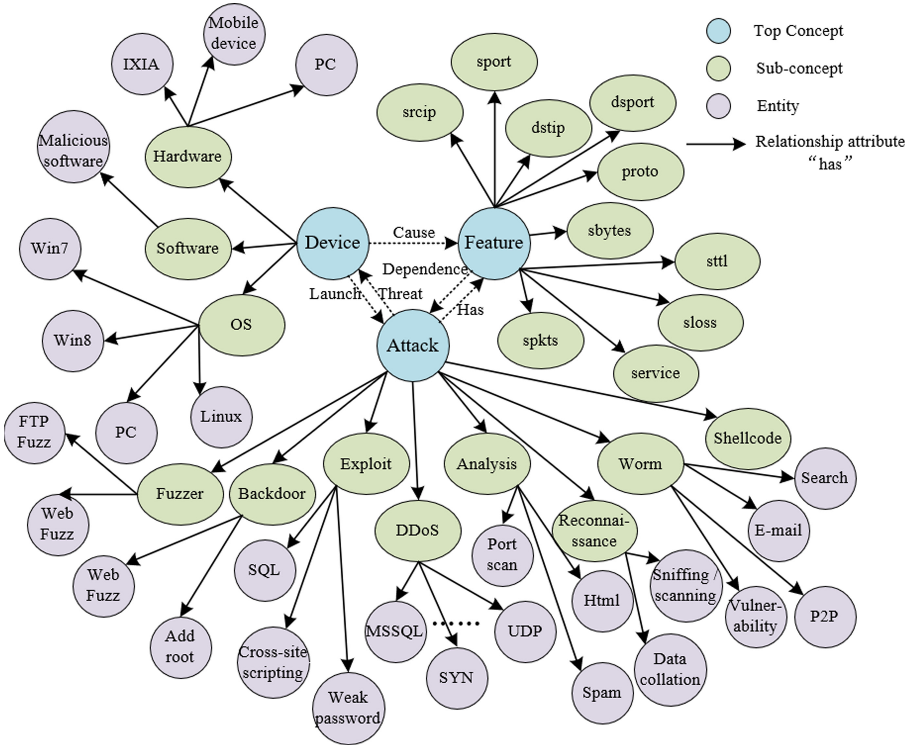

Fig. 2. Cybersecurity ontology

4 Implementation of CSKB

In this section, we will discuss the realization of the CSKB based on the knowledge graph. The implementation process is based on the cyber security ontology in Sect. 3. The implementation process of the CSKB based on knowledge graph we proposed is shown in Fig. 3. It includes five stages: knowledge acquisition, knowledge fusion/extraction, knowledge storage, knowledge inference and knowledge update. Each stage will be explained in the following subsections.

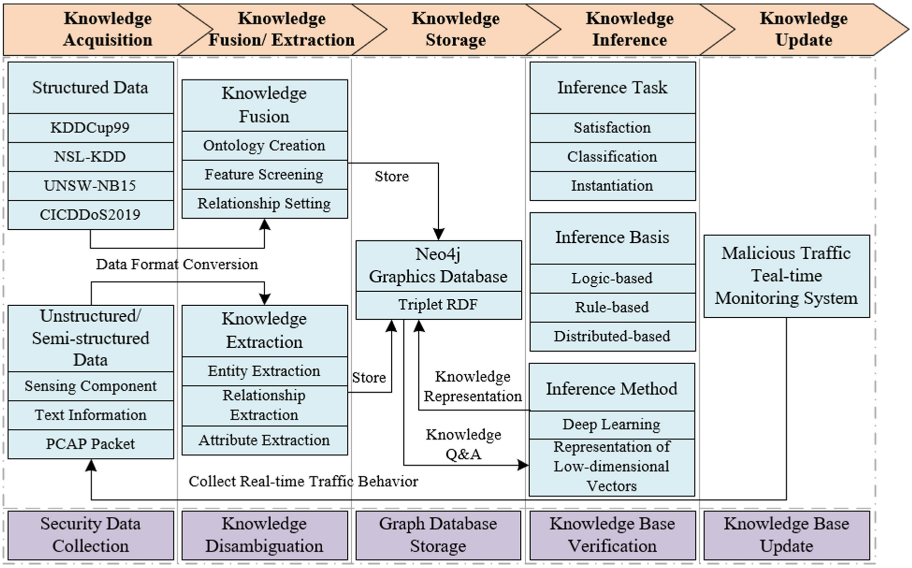

Fig. 3. The solution of constructing CSKB based on knowledge graph

4.1 Knowledge Acquisition and Knowledge Fusion/Extraction

In the previous section, we have constructed a complete cyber security ontology for the CSKB. The process of ontology construction has reflected how to obtain and integrate multi-source security data, and it describes the relationship between cyber security entities and entities. Therefore, we combine the two stages of knowledge acquisition and knowledge fusion/extraction. Based on the cyber security ontology, we can describe how to obtain useful information from multi-source and heterogeneous massive security data and convert it to a triple Resource Description Framework (RDF) format that the graph database can store.

Attack data sources in the field of cyber security are distributed discretely in security databases, PCAP files, security documents, Internet drafts and other media. As shown in Fig. 3, the security data can be divided into structured data, semi-structured data, and unstructured data according to the data type. Above all, structured security data is generally stored in the format of a security data set, and its confidence is usually high. They can be mapped into the cyber security ontology model, and the redundant data can be used for knowledge disambiguation through knowledge fusion technology (such as feature selection). These structured data are an important part of the initial construction of the CSKB. Secondly, part of the attack data is contained in the PCAP format data packets obtained in real-time network attacks. They are called semi-structured data and need to be analyzed using knowledge extraction tools. In turn, they are converted into structured data and stored in a triple RDF format suitable for graph database storage through data fusion. In this paper, we use the CICFlowMeter [17] tool to create reliable features from PCAP files and save them as structured security data sets. Finally, some security documents or Internet drafts usually contain security data without any structure, which can provide a basis for the expansion of cyber security ontology and CSKB. All in all, in the field of cyber security, the main security data comes from structured or semi-structured data. We use CICFlowMeter tool and knowledge fusion technology to complete the mapping of data to cyber security ontology.

4.2 Knowledge Storage

In order to effectively express the relationship between entities, we usually use a graph database to store knowledge graphs, rather than a conventional table database. The form database usually has a fixed data structure, but the knowledge stored in the knowledge graph always changes dynamically. Therefore, we use the Neo4j graph database, which is a NoSQL database with a graph engine as the core, as the storage carrier of the CSKB. It can effectively solve the problem that the table database has insufficient processing capacity when coping with dynamic data changes. The concepts or entities in the Neo4j graph database are stored in the form of nodes, and the directed edges connecting the nodes represent the relationships between the entities. When the cybersecurity data structure changes, we need to add or delete the corresponding nodes and edges; when the data content changes, we only need to modify the attributes of the nodes or edges.

4.3 Knowledge Inference

After the above stages, we have integrated a multi-source and heterogeneous cybersecurity data and used a unified semantic data structure (such as the triple RDF) to store the data in the Neo4j graph database. Finally, a preliminary CSKB was successfully constructed. However, when we collect a large amount of heterogeneous cybersecurity data through knowledge acquisition methods and transform it into the CSKB, the reliability of the data cannot be guaranteed, so we need to complete the classification and recommendation of the data through inference algorithms. Knowledge reasoning can generally be divided into logic-based inference, rule-based inference, and algorithm-based inference. Since cybersecurity data has distinct data features and is more restrictive in logic and rules, we focus on algorithm-based knowledge inference methods to ensure the reliability of security data in the CSKB.

Deep learning can effectively identify the types of network attacks based on input features. Therefore, based on the high-confidence cybersecurity data stored in the CSKB, deep learning can construct a reliable neural network model to identify new types of knowledge. However, the process of deep learning is a black-box model. Therefore, in order to ensure that the knowledge inference process is recognizable, we designed a path sorting algorithm, *TransFeature*, to further verify the reliability of the input safety data content. The process of knowledge inference is shown in Fig. 4.

Deep learning is not the focus of this article, so we choose a Convolutional Neural Network (CNN) consisting of two convolutional layers and two maximum pooling layers as a model for identifying the type of network attack data. We take all the feature entities in the CSKB as the input of CNN and set the corresponding attack entities as labels. After training, we can obtain the trained model to determine the type of attack to which the input security data belongs. According to the judgment of the model, the data that cannot be mapped as the attacking entity in the cybersecurity database is discarded;

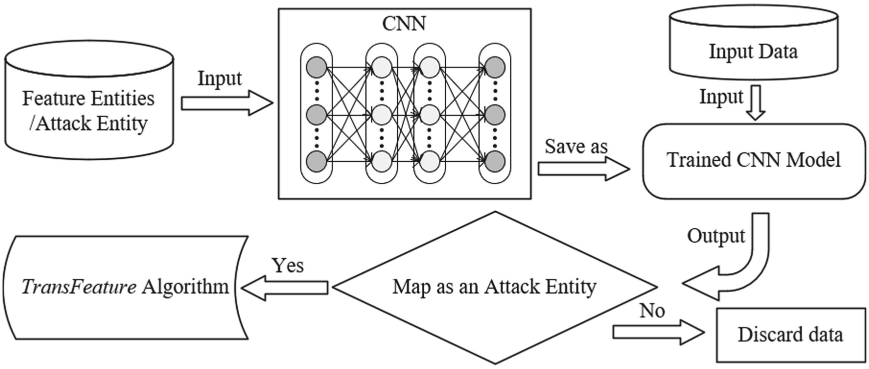

Fig. 4. The process of knowledge inference

The data that can be mapped to a certain attack entity inputs the feature entity into the *TransFeature* algorithm to determine whether the security knowledge is reliable.

TransFeature is the process of learning the low-dimensional vector representation of entities and relationships and comparing entities to achieve the goal of optimization. Due to the particularity of network attack features, it is impossible to describe the type of network attack through a single feature vector, so we define the feature vectors and related calculations as shown in Eqs. (4)–(8).

$$\overrightarrow{F_i} = \left(\overrightarrow{b_i}, \overrightarrow{t_i}, \overrightarrow{l_i}, \overrightarrow{e_i}, \overrightarrow{p_i}\right) \tag{4}$$

$$\overrightarrow{F'} = \left(\overrightarrow{b'}, \overrightarrow{t'}, \overrightarrow{l'}, \overrightarrow{e'}, \overrightarrow{p'}\right) \tag{5}$$

$$\tau_k = \left\|\overrightarrow{k}\right\|_{max} - \left\|\overrightarrow{k'}\right\| k = b, t, l, e, p \tag{6}$$

$$\overrightarrow{\mathrm{T}} = \left(\tau_b, \tau_t, \tau_l, \tau_e, \tau_p\right) \tag{7}$$

$$d_i = \left\|\overrightarrow{F_i} - \overrightarrow{F'}\right\| \tag{8}$$

Among them, the vectors $\overrightarrow{b_i}$, $\overrightarrow{t_i}$, $\overrightarrow{l_i}$, $\overrightarrow{e_i}$, $\overrightarrow{p_i}$ in Eq. (4) represent the two-dimensional vector representations of feature entities sbytes, sttl, loss, service, and spkts in the *i*-th data packet. We only select the last five feature entities because they represent the features of the attack packet itself, $\overrightarrow{F_i}$ represents the set of feature vectors. The vectors $\overrightarrow{b'}$, $\overrightarrow{t'}$, $\overrightarrow{l'}$, $\overrightarrow{e'}$, $\overrightarrow{p'}$ in Eq. (5) respectively correspond to the mean vector of each feature vector stored in the CSKB, $\overrightarrow{F'}$ represents the mean feature vector set. τ_k in Eq. (6) represents the maximum difference of each feature scalar, and is stored in the threshold vector $\overrightarrow{\mathrm{T}}$ in Eq. (7). Equation (8) calculates the total distance d_i from the feature vector of the *i*-th packet to the mean feature vector. Finally, we determine whether the input safety knowledge is reliable by determining the size of d_i and the threshold $\vec{T}$. If d_i is greater than $\vec{T}$, we determine that the security data is unreliable, so we implement the option of discarding knowledge; otherwise, we determine that knowledge is reliable, and then store the security data in the CSKB.

4.4 Knowledge Update

Finally, based on the cybersecurity ontology and knowledge graph, we constructed a CSKB that can reflect the dynamic attributes and types of network attacks and ensured the reliability of cybersecurity knowledge through knowledge inference. In order to ensure that the CSKB can keep up with the development of modern attacks, we propose a new stage, knowledge update. As the infrastructure for generating and undertaking attacks, the network has a large scale, high complexity, and strong uncertainty. Therefore, we should pay attention to the problem of malicious traffic caused by terminal devices in massive connections, and build a large-scale network scenario that can reflect the malicious behavior of modern networks. By establishing a corresponding prototype system, we can monitor malicious traffic in real-time and update knowledge to continuously ensure the real-time and reliability of cybersecurity knowledge.

5 Performance Evaluation

In this section, we show the comparative analysis between the knowledge in CSKB and various data sets. Then, different machine learning techniques have been utilized to compare the classification performance of CSKB with other datasets.

Table 4 shows the comparative analysis between CSKB, NSL-KDD, UNSW-NB15, and CICDDoS2019 datasets. We compared six typical parameters, namely attack families, DDoS attack families, feature extraction tools, number of features, storage format, and data update capability. It can be observed that CSKB has the most attack types compared to the other three datasets. In particular, CSKB also covers 13 DDoS attacks, which can reflect modern attack types to a certain extent. However, by comparison, we found that CSKB has the least number of features. This is because we hope to achieve the specification and integration of multi-source and heterogeneous security data by defining a unified feature ontology classification. What's more, CSKB can continuously filter and update data through the stage of knowledge inference and knowledge update, so it has higher flexibility and scalability.

Since exploring classification methods on the datasets discussed is not the focus of this work, we use Tensorflow [18] to implement five machine learning models for performance analysis. Each model is described as follows:

Logistic Regression (LR): we use the default L2 Regularization to prevent the model from overfitting.
Naive Bayes (NB): the default of Gaussian NB is used.
K-Nearest Neighbor (KNN): we use a cross-validation method to select the optimal K = 6 to balance processing time and classification accuracy.
Decision Tree (DT): Entropy is used as a splitting criterion. In addition, we limit the tree depth to 20 to prevent overfitting.

Table 4. Comparison of datasets

Parameters	NSL-KDD	UNSW-NB15	CICDDoS2019	CSKB
Attack families	4	9	13	30
DDoS attack families	0	0	13	13
Feature extraction tools	Bro-IDS	Bro-IDS, Arugs	CICFlowMeter	CICFlowMeter
Number of features	42	49	80	15
Storage format	CSV	CSV	CSV	Knowledge graph
Data update capability	No	No	No	Yes

Random Forest (RF): the number of base evaluators has a monotonic effect on the accuracy of the RF. The greater the number of evaluators, the better the effect of RF. Therefore, we set the number of evaluators to 100.

When analyzing the performance of the classifier on each data set, the commonly considered indicators are Accuracy, Recall, and F1-Score. Among them, F1-Score is defined as the harmonic mean of Precision and Recall. Finally, we selected the F1-Score as the evaluation indicator. If the F1-Score is larger, the classification performance of the data set is better in the machine learning model.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \in [0, 1] \tag{9}$$

In order to quantitatively analyze the performance of different data sets in each classifier, we set each data set to randomly select 80,000 pieces as the training set and 20,000 pieces as the testing set. The F1-Score obtained for each data set is shown in Fig. 5.

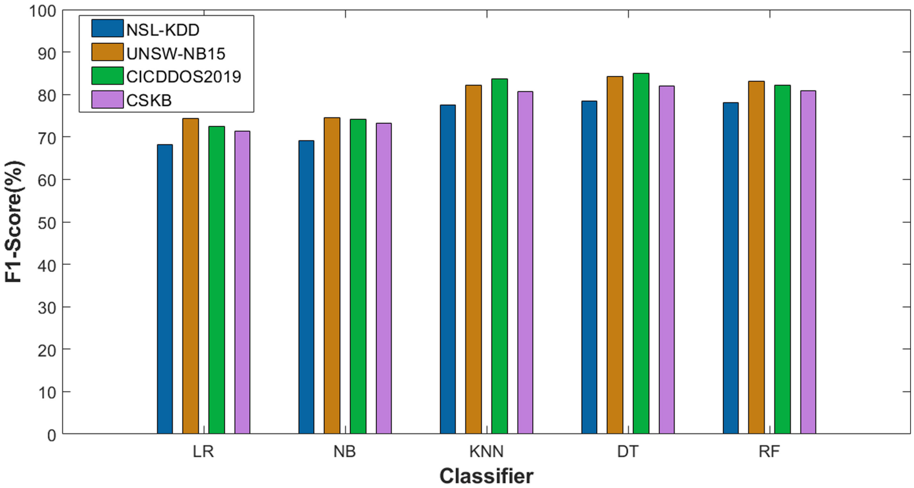

Fig. 5. Comparison of F1-score for NSL-KDD, UNSW-NB15, CICDDoS2019, and CSKB

From Fig. 5, each data set has obtained a high F1-Score in the process of classifying malicious attacks. In addition, it is evident that CSKB equals or betters NSL-KDD on all learning models implemented. Therefore, the experiment proves that the knowledge stored in CSKB effectively realizes the specification and integration of security data.

6 Conclusion

This article is dedicated to solving the problem that traditional security data sets are difficult to reflect the modern and complex network attack environment. We built a standardized cybersecurity ontology based on the knowledge graph and realized CSKB from five stages: knowledge acquisition, knowledge fusion/extraction, knowledge storage, knowledge reasoning, and knowledge update, aiming at fully reflecting the dynamic nature of modern network attacks and providing a reliable basis for real-time cybersecurity protection solutions. Experiments prove that the knowledge stored in CSKB can effectively realize the specification and integration of security data. In future work, we consider expanding CSKB as a communication behavior knowledge base and then establish an intelligent and trusted platform for adaptive memory communication behavior.

Acknowledgement. This paper is supported by National Key R&D Program of China under Grant No. 2018YFA0701604, NSFC under Grant No. 61802014, No. U1530118, and National High Technology of China ("863 program") under Grant No. 2015AA015702.

References

1. Yu, S., Liu, M., Dou, W., Liu, X., Zhou, S.: Networking for big data: a survey. IEEE Commun. Surv. Tutor. **19**(1), 531–549 (2017)
2. Feng, B., Zhou, H., Zhang, H., et al.: HetNet: a flexible architecture for heterogeneous satellite-terrestrial networks. IEEE Network **31**(6), 86–92 (2017)
3. KDD99 (2007). kdd.ics.uci.edu/databases/
4. NSLKDD (2009). nsl.cs.unb.ca/NSLKDD/
5. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, pp. 1–6 (2015)
6. Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A.: Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, pp. 1–8 (2019)
7. Yu, S.: Big privacy: challenges and opportunities of privacy study in the age of big data. IEEE Access **4**, 2751–2763 (2016)
8. Song, Q., Wu, Y., Lin, P., Dong, L.X., Sun, H.: Mining summaries for knowledge graph search. IEEE Trans. Knowl. Data Eng. **30**(10), 1887–1900 (2018)
9. Yu, S., Zhou, W., Guo, S., Guo, M.: A feasible IP traceback framework through dynamic deterministic packet marking. IEEE Trans. Comput. **65**(5), 1418–1427 (2016)
10. Zhu, Z., Jiang, R., Jia, Y., Xu, J., Li, A.: Cyber security knowledge graph based cyber attack attribution framework for space-ground integration information network. In: 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, pp. 870–874 (2018)

11. Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y., Tang, F.: Discriminating DDoS attacks from flash crowds using flow correlation coefficient. IEEE Trans. Parallel Distrib. Syst. **23**(6), 1073–1080 (2012)
12. Yu, S., Wang, G., Zhou, W.: Modeling malicious activities in cyber space. IEEE Network **29**(6), 83–87 (2015)
13. Neo4j (2020). neo4j.com/
14. Feng, B., Zhang, H., Zhou, H., Yu, S.: Locator/identifier split networking: a promising future internet architecture. IEEE Commun. Surv. Tutor. **19**(4), 2927–2948 (2017)
15. Xu, G., Cao, Y., Ren, Y., Li, X., Feng, Z.: Network security situation awareness based on semantic ontology and user-defined rules for internet of things. IEEE Access **5**, 21046–21056 (2017)
16. Islam, C., Babar, M.A., Nepal, S.: An ontology-driven approach to automating the process of integrating security software systems. In: 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP), Montreal, QC, Canada, pp. 54–63 (2019)
17. CICFlowMeter (2017). www.github.com/ISCX/
18. TensorFlow (2020). https://tensorflow.google.cn/