Construction of a Cybersecurity Behavior Knowledge Base for Malicious Behavior Analysis

Keke Feng, Huachun Zhou, Weilin Wang, Jingfu Yan and Xiaojing Fan

School of Electronic and Information Engineering, Beijing Jiaotong University, China

Email: 22120046@bjtu.edu.cn

Received: September 24, 2024; Revised: November 20, 2024; Accepted: December 10, 2024; Published: December 30, 2024

Abstract

Facing the surge in malicious behaviors in the network environment, the existing cybersecurity knowledge graph suffers from fragmented security knowledge and limited application scenarios, making it challenging to collaborative malicious behavior analysis. To address this, we propose a cybersecurity behavior knowledge base (CSBKB) framework for comprehensive malicious behavior analysis. Based on knowledge of user behavior, attack traffic, and attack paths, we construct six types of knowledge graphs to characterize malicious behavior, including user behavior perception, user behavior mapping, malicious behavior association, malicious behavior category, domain attack, and malicious behavior path traceability graph. These graphs characterize malicious behaviors and form a comprehensive security behavior knowledge base. To fully utilize the graph structure information, we design a reasoning module based on the graph neural network further to explore the relationship between entities in the graph. Using DDoS attacks as a case study, we demonstrate this framework's construction and knowledge-reasoning capabilities. Experimental results demonstrate that the proposed CSBKB framework effectively realizes a comprehensive malicious behavior analysis mechanism encompassing "malicious user behavior monitoring, malicious behavior type detection, and malicious behavior path tracing." It can effectively analyze malicious behaviors, with an accuracy of more than 0.97 in detecting abnormal users, more than 0.97 in inferring DDoS attack types, and an identification rate of more than 0.92 for malicious behavior paths.

Keywords: Cybersecurity knowledge base, Knowledge reasoning, Malicious behavior analysis, Graph neural networks.

1 Introduction

With the development of new information technologies and applications, the scale of the network continues to expand, resulting in an explosive growth of network security data generated in cyberspace. Consequently, network security managers find it increasingly difficult to quickly locate information related to malicious behavior and utilize it effectively [1]. Faced with complex network environments and a surge in malicious behaviors, existing network security protection solutions based on statistical methods or machine learning have problems such as complex data structure, harrowing feature extraction, poor generalization ability, and slow update of security detection knowledge [2].

As a specific knowledge graph in the security field, the Cyber Security Knowledge Graph (CSKG) is a large-scale semantic network composed of nodes and edges. It provides an intuitive modeling method for various attack scenarios in network security. In most related studies, it is synonymous with the cyber security Knowledge Base [3]. Cybersecurity knowledge graph data mainly comes from multi-source heterogeneous data such as security databases, reports, social media, and blogs [4]. The constructed graphs typically target single application scenarios like situation awareness, attack prediction, and attack path analysis [5]. The scattered data sources and single application scenarios lead to issues in the current security knowledge graphs,

such as difficulty in intercommunication of security knowledge and inconvenient collaboration in malicious behavior analysis. Additionally, these security knowledge graphs often lack a focus on network layer behavior analysis and reasoning.

The reasoning technology of network security knowledge graphs serves as the core technological foundation for cognitive intelligence. Knowledge reasoning based on Graph neural networks can simultaneously consider both the semantic and structural information of the knowledge graph [6], Compared to traditional models based on logic rules and representation learning, which suffer from issues of low efficiency and limited rule coverage [7,8], reasoning using graph neural networks offers superior interpretability. It is urgent to build a knowledge base that effectively manages cybersecurity data and enables practical analysis and intelligent reasoning of malicious behaviors in communication networks.

Therefore, this paper proposes a cyber security behavior knowledge base (CSBKB) framework for malicious behavior analysis to address the issues above. The main contributions of this paper are as follows:

(1) The proposed framework extracts feature from user behavior data based on identity trust protocols and five types of mainstream DDoS attack traffic malicious communication behaviors, completing data preprocessing and structuring. The five types of attacks are network layer DDoS, application layer DDoS, low-rate DDoS, DRDoS, and botnet DDoS.

(2) Based on user features, DDoS attack traffic features, and path tracing features, we build a CSBKB, forming six types of knowledge graphs: user behavior perception graph, user behavior mapping graph, malicious behavior association graph, malicious behavior category graph, domain attack graph, and malicious behavior path traceability graph.

(3) We design Graph neural network algorithms for reasoning across three inference graphs: the user behavior perception graph, the malicious behavior association graph, and the domain attack graph realizing a comprehensive malicious behavior analysis mechanism of "malicious user behavior monitoring-malicious behavior type detection-malicious behavior path tracing".

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 gives a detailed introduction to the proposed framework, Section 4 gives the detailed results and analysis of the experiments, and Section 5 concludes this study and provides some suggestions for future work.

2 Related Work

This section reviewed the related work on cybersecurity knowledge graph(CSKG) construction scenarios, malicious behavior analysis reasoning technology, and graph neural network technology in malicious behavior analysis.

The cybersecurity knowledge graph can effectively analyze, mine, and associate massive amounts of data and information in the cybersecurity field. The literature [9] generated an extended attack graph to obtain the most probable vulnerability path and provide the success rate and loss of power grid attacks. The literature [10] specifically introduced an external dictionary, compiled from sources such as cybersecurity-related blogs, vulnerability databases, and Wikipedia, when generating word embedding vectors. This approach reflects the patterns and nuances of the cybersecurity domain. The literature [11] proposed a large-scale analysis and defense framework using aggregated CTI. It used knowledge graphs to extract information and store it in a structured format, retaining the semantics of threat intelligence. The literature [12] proposed an APT threat knowledge extraction algorithm that leverages deep learning and expert knowledge to complete and update the knowledge graph. This approach facilitates a defense method that integrates much-fragmented intelligence and can actively adjust defense strategies. However, the above literature focused on a single scenario when constructing knowledge graphs without paying attention to modeling malicious behaviors such as network

layer behaviors and user behaviors. The malicious behavior knowledge base built in the literature [13] is relatively complete. Still, its work focuses on implementing the distributed knowledge base and constructing and designing malicious behavior graphs. There is a gap in the research of malicious behavior reasoning based on graphs.

Regarding the analysis and reasoning of malicious behavior, the literature [14] proposed a feature construction method to simulate user access behavior. It used the UCI machine learning repository database to train and test the behavior to detect abnormal users. It did not use knowledge graphs but provided a new perspective for malicious behavior analysis. The literature [15] assigned a weight to the nodes and edges in the traceability graph to express the threat level and used a greedy algorithm to find the attack path based on the threat level. Researchers have continually explored reasoning based on cybersecurity knowledge graphs to investigate malicious behaviors further. The literature [16] proposed a defense strategy reasoning model comprising a knowledge graph embedding algorithm (CTI-KGE) and reasoning rules. This model can automatically infer tail entities with any relationship with the head entity and complete threat information. The literature [17] integrated vulnerabilities, weaknesses, affected platforms, tactics, attack techniques, and patterns into coherent links. It employed reasoning rules to perceive threats in complex heterogeneous environments. The literature [18] pioneered a novel approach based on knowledge graph reasoning and knowledge embedding. This innovative method automatically detects potential attack patterns, significantly enhancing the efficiency and accuracy of network threat identification. However, the graph structure of the knowledge graph is not fully considered and applied during reasoning, making it impossible to achieve comprehensive reasoning about malicious behavior.

In the field of malicious behavior analysis utilizing graph neural network technology, the literature [19] proposed a novel Recursive Evolution Network (RE-GCN) based on the Graph Convolutional Network (GCN). This model leverages GCN to learn the evolutionary representations of entities and relations at each timestamp, providing a dynamic and comprehensive approach to understanding malicious behaviors over time. The literature [20] proposed E-GraphSAGE based on GraphSAGE, which can capture the edge features of graphs in flow-based data and the topological information for network intrusion detection in the network. The above work has not integrated the knowledge graph. Graph neural networks use graph data representation methods. They can learn based on the graph structure's topological connections and node attributes to meet the knowledge graph's reasoning needs. The literature [21] employed a Graph Attention Network (GAT) model to simultaneously learn nodes' feature information and structural relationships within the malicious behavior structure graph. This approach transforms the task of malicious behavior detection into a node classification problem. The literature [22] applied GCN to DDoS attack path tracing, which determines the attacker's identity and location by restoring the DDoS attack's complete path. The literature [23] proposed a relation-aware graph attention network (QRGAT), which encodes the reasoning process into a reasoning graph and collaboratively captures the dependencies of different relation paths of each entity. Even so, applying Graph Neural Network reasoning methods for in-depth analysis of relationships and entities within malicious behavior knowledge graphs remains an area that requires further exploration.

3 Cyber Security Behavior Knowledge Base Framework

In this section, we first propose the construction requirements of the CSBKB for malicious behavior analysis, design a construction framework based on the requirements, and then introduce the overall process of the framework and the design and implementation rules of each module in detail.

3.1 Demand Analysis

Malicious Behavior Analysis (MBA) aims to identify, understand, and respond to malicious behaviors or attacks by monitoring, detecting, and analyzing abnormal and suspicious activities in the network. The CSBKB is an essential reference for formulating defense strategies. It not only needs to extract comprehensive and practical knowledge from multi-source heterogeneous data but also can reason and analyze behaviors. On the

defensive side, the knowledge base for malicious behavior analysis should be capable of answering critical questions such as who attacked, how the attack was carried out, and what the attack path was [24]. This capability enables the development of customized defense measures and countermeasures. Considering the above requirements, we design a CSBKB integration framework, including user behavior base, traffic behavior base, and attack behavior base. At the same time, we design behavioral reasoning modules in combination with the graph neural network algorithm to conduct a collaborative analysis of malicious user behavior, malicious behavior types, and malicious behavior paths. This module aims to address the questions: "Who attacked me?", "In what way?", and "What is the attack path?"

3.2 Framework Overview

To achieve intelligent analysis of malicious behaviors, we designed a CSBKB framework as shown in Figure. 1. The overall framework includes a data layer, a knowledge layer, and a reasoning layer

The knowledge base system is deployed at the gateway entrance. When DDoS attacks occur in the domain, the behavior data collection module of the data layer collects the user behavior data sent by the identity-based trusted protocol module and the DDoS attack traffic data occurring in the domain. Then, according to the behavior definition rules, data passes through the data processing module to form a structure that can be used to generate a knowledge spectrum in the cyber security behavior knowledge base. The knowledge layer stores the constructed CSBKB. After receiving data through the data acquisition interface, the CSBKB constructs corresponding knowledge maps in the user, traffic, and attack behavior bases. The reasoning layer obtains the graph dataset from the knowledge layer graph through the knowledge reasoning interface, which provides feature and connection information for training. The three reasoning modules use graph convolutional network, E-GraphSAGE, and graph attention network technologies, in turn, to train and save the optimal model and use the optimal model for online reasoning to achieve malicious user behavior discovery, malicious behavior type detection, and malicious behavior path tracing. Finally, after reasoning, the reasoning results of each module are updated to the corresponding knowledge graph of the knowledge layer through the knowledge update interface, and the content in the third-party bases is linked to analyze malicious behavior.

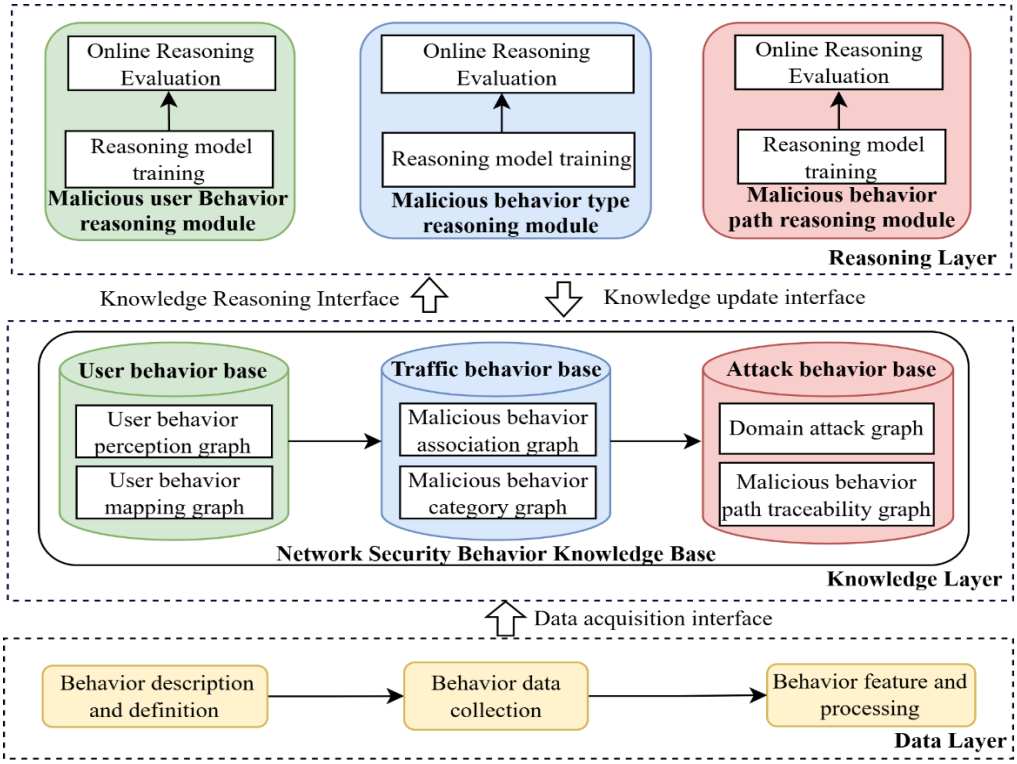

Figure. 1. Framework for building CSBKB

3.3 Data Layer

3.3.1 Behavior Description and Definition

The essence of a communication network is traffic communication between users. The user is the initiator of communication behavior. Abnormal users initiate malicious behavior in the network. The behavior initiated by the user is transmitted in the form of traffic. Attack behavior includes the attack's type, source, and forwarding path. At the same time, to facilitate the reasoning of the attack path, the attack behavior in this article also covers network topology information. Therefore, the cybersecurity behavior (CSB) defined in this paper as shown in (1) :

$$\mathrm{CSB} =< \text{User behavior}, \text{Traffic behavior}, \text{Attack behavior} > \quad (1)$$

The information transmitted in the network can be extracted into security behaviors that the knowledge base can understand. Based on the above analysis, we define each type of network security behavior in the cybersecurity behavior knowledge base in detail.

User behavior is defined based on the information record of the identity-based trusted protocol Combined with the method of viewing behavior from the perspective of statistical characteristics in the literature [14], as shown in (2).The definition includes the user's identity identifier UID, user attribute UA, user EID address, number of successful identity authentications $\mathrm{IA_S}$, number of failed identity authentications $\mathrm{IA_F}$, number of successful access control $\mathrm{AC_S}$, number of failed access control $\mathrm{AC_F}$, number of successful separation mappings $\mathrm{SM_S}$, number of failed separation mappings $\mathrm{SM_F}$, type of failed access resource ARTF.

$$\text{User behavior} =< \mathrm{UID}, \mathrm{UA}, \mathrm{EID}, \mathrm{IA_S}, \mathrm{IA_F}, \mathrm{AC_S}, \mathrm{AC_F}, \mathrm{SM_S}, \mathrm{SM_F}, \mathrm{ARTF} \ldots > \quad (2)$$

Traffic behavior is a collection of traffic characteristics in a certain period in the network, as shown in (3), including source port SP, destination port DP, source address SI, destination address DI, flow duration FD, flow rate FS, flow number FN, flow type FT, flow flag FF, etc.

$$\text{Traffic behavior} =< \mathrm{SP}, \mathrm{DP}, \mathrm{SI}, \mathrm{DI}, \mathrm{FD}, \mathrm{FS}, \mathrm{FN}, \mathrm{FT}, \mathrm{FF}, \ldots > \quad (3)$$

Attack behavior includes the information inferred from user behavior and traffic behavior, as well as the information reflecting the attack environment state of the network topology node, as shown in (4). These include network topology NT, source address SI, destination address DI, attack start time ST, attack end time ET, malicious behavior type TP, entropy feature EC, packet rate PR, byte rate BR, etc.

$$\text{Attack behavior} =< \mathrm{NT}, \mathrm{SI}, \mathrm{DI}, \mathrm{ST}, \mathrm{ET}, \mathrm{TP}, \mathrm{EC}, \mathrm{PR}, \mathrm{BR} \ldots > \quad (4)$$

3.3.2 Behavioral Data Collection

The collected behavioral data comes from two aspects: one is the user historical behavior data of the identity-based trusted protocol module [25]; the other is the traffic data of five major types of attacks, including network-layer DDoS attacks, application-layer DDoS attacks, low-rate DDoS attacks, distributed reflection denial of service attacks (DRDoS), and botnet DDoS attack [21,26].

We utilize the written Python script to collect statistics on data reflecting the user status within the network during the four processes of user authentication, access control, separation mapping, and reputation evaluation in the identity-based trusted protocol. We collect this data as user behavior data. To convert traffic data and extract 84 features in CSV text format as traffic behavior data, we utilize the traffic feature extraction tool CICFlowMeter [27]. Additionally, we use the network packet analysis tool Scapy [28] to extract entropy feature data related to IP addresses and forwarding ports, as well as packet rate and byte rate data from the traffic data, categorizing this information as attack behavior data.

3.3.3 Behavioral Feature and Processing

After collecting the behavior data, some abnormal data will be output, so we need to clean the collected malicious behavior data. We directly deleted the missing values in the collected data and determined 12 user behavior data features based on the user behavior description to distinguish abnormal users, as shown in Table 1.

In addition, considering the 84-dimensional flow features in the traffic behavior data, utilizing all of them would consume significant storage and computational resources. Moreover, not all features in the attack behavior data hold equal importance when selecting attack paths. Therefore, this paper employs the Random Forest Algorithm in conjunction with Recursive Feature Elimination (RFE) [29] to perform feature selection on the original feature set. Initially, we identify and optimize essential features to select the optimal feature set. Ultimately, we choose 26 features from the traffic behavior data and 11 path state features from the attack behavior data. The specific features are in Table 2 and Table 3.

Table 1. User Behavior Data Features

Feature	Description	Feature	Description	Feature	Description
IA	Number of user identity authentications	IA_F	Number of failed user identity authentications	ARIAT	The average interval between authentications
AC	Number of user resource accesses	AC_F	Number of failed user resource accesses	NHS	Number of accesses to highly sensitive resources
NLS	Number of accesses to lowly sensitivity resources	AC_{FA}	Type of failed access actions	NAC	Number of user permission changes
SM	Number of separation mappings	SM_F	Number of failed separation mappings	CW	User reputation score

Table 2. Traffic Behavior Feature Set

Type	Feature	Type	Feature
Flow Feature	Flow Bytes/s	Time Feature	Flow IAT Mean
	Flow Packets/s		Flow IAT Std
	Down/Up Ratio		Fwd IAT Min
Packet Feature	Fwd Packet Length Max		Idle Max
	Fwd Packet Length Mean		Idle Mean
	Fwd Packet Length Std		Bwd IAT Total
	Average Packet Size	Other Feature	Fwd Act Data Pkts
	Packet Length Max		Fwd Header Length
	Packet Length Variance		Fwd Packets/s
Signature Feature	ACK Flag Count		Fwd Seg Size Min
	FIN Flag Count		Fwd Segment Size Avg
Time Feature	Flow Duration		Bwd Init Win Bytes
	Flow IAT Max		Bwd Packets/s

Table 3. Path State Feature Set

Feature	Description
Packet rate	Average number of packets forwarded per second
Byte rate	Average number of bytes forwarded per second
Source IP entropy	Entropy of source IP addresses
Destination IP entropy	Entropy of destination IP addresses
TCP source port entropy	Entropy of TCP source ports
TCP destination port entropy	Entropy of TCP destination ports
UDP source port entropy	Entropy of UDP source ports
UDP destination port entropy	Entropy of UDP destination ports
H(Sip|Dip)	The conditional entropy of the source IP given the destination IP
H(Sip|Dport)	The conditional entropy of the source IP given the destination port
H(Dport|Dip)	The conditional entropy of the destination port given the destination IP

3.4 Construction of CSBKB

3.4.1 Overview of Knowledge Graph Construction

Figure. 2 shows the main contents and uses of each Graph. The graphs constructed in this section are divided into two categories. The reasoning graph provides the graph structure and data required for reasoning for the reasoning module, including the user behavior perception graph, malicious behavior association graph, and domain attack graph. The other is the display graph, updated by the reasoning module based on the results, which stores and displays the analysis results of malicious behaviors, including the user behavior mapping graph, malicious behavior category graph, and malicious behavior path traceability graph.

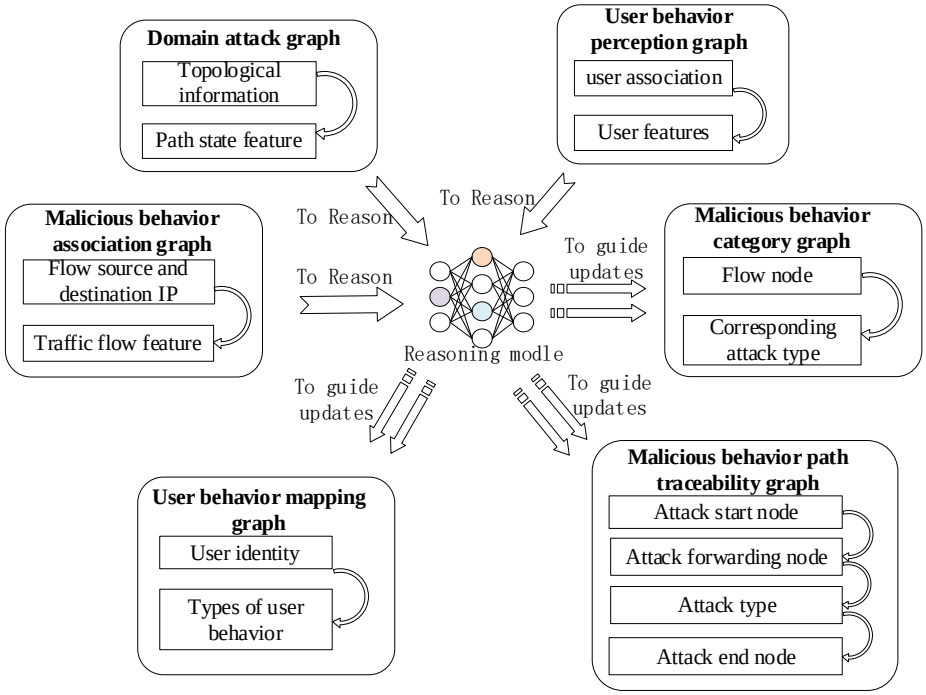

Figure. 2. Knowledge Graph Overview

3.4.2 Construction of User Behavior Base

Table 4. Composition of user behavior perception graph

Entity	Attributes	Relation
Unknown user node	12 user behavior features	Associated users with unknown behavior
Normal user node	12 user behavior features	Associated users with normal behavior
Abnormal user node	12 user behavior features	Associate users with abnormal behavior

We construct the user behavior perception graph and the user behavior mapping graph in the user behavior base. The user behavior perception graph reflects users' behaviors within the network, storing the behavioral features of users and the connections between them. Table 4 presents the structure of the triples in this graph. The 12 user behavior features in the attributes come from Table 1.

The construction of the user behavior perception graph begins with the user node as the primary entity. The connections between users are determined based on their communication times, linking users interacting within a thirty-minute window. This Graph serves as the input for reasoning about malicious user behavior.

The user behavior mapping graph illustrates user activities, with the triple composition of the map detailed in Table 5. The entities involved are user nodes, user behavior nodes, and behavior type nodes. The user

behavior type nodes encompass three categories: abnormal users, normal users, and unknown users. We establish relationships between user nodes and user behavior nodes based on the collected user types and the behavior types inferred by the malicious user behavior reasoning module.

Table 5. Composition of user behavior mapping graph

Entity	Attributes	Relation
User behavior type	User behavior on the network	Associate 3 user behaviors
Normal user	Normal user behavior	Associate normal user node
Abnormal user	Abnormal user behavior	Associate abnormal user nodes
Unknown user	Undetermined nature user behavior	Associate undetermined nature user node
User Node	12 user behavior features	Associate 1 user behavior

3.4.3 Construction of Traffic Behavior Base

We construct malicious behavior association graphs and malicious behavior category graphs in the traffic behavior base.

Table 6. Composition of malicious behavior association graph

Entity	Attributes	Relation
Malicious communication flow	26-dimensional flow features	Associate User Node
Network layer DDoS attack nodes	IP address and port	Correlate network layer DDoS attack flows
Application layer DDoS attack nodes	IP address and port	Correlate application layer DDoS attack flows
Low-rate DDoS attack node	IP address and port	Correlate low-rate DDoS attack flows
DrDoS attack node	IP address and port	Correlate DrDoS attack flows
Botnet attack node	IP address and port	Correlating botnet attack flows

The malicious behavior association graph illustrates the communication flow relationships between user nodes within the network. It is input for the E-GraphSAGE model within the malicious behavior type reasoning module. Table 6 presents the graph triples. Nodes represent combinations of IP addresses and port numbers. A flow from one node to another is generated when a communication relationship exists between devices. At this point, an edge is established between the two nodes to represent this flow, with all features and information of the flow added to the corresponding edge as attributes.

Table 7. Malicious behavior category graph composition

Entity	Attributes	Relation
Communication behavior	Communication behavior in the network	Associate flow type node
Flow type node	Description of each attack type	Associate flow node
Flow node	26-dimensional traffic feature description	Associate flow type node

The malicious behavior category graph is mainly used to reflect the malicious behavior type to which the traffic node belongs and to provide query results for the traffic type. It can guide the formulation of attack response measures, including but not limited to the description of DDoS attacks. Table 7 shows the triplet structure of the graph.

Each flow is defined as a communication behavior in the constructed malicious behavior category graph and is represented as a flow behavior node in the Figure. Each flow node corresponds to a specific communication behavior and points to a DDoS attack type node or a normal flow node, indicating the type of communication behavior.

We can build a malicious behavior category graph for labeled flow nodes based on their type. For unlabeled

flow, we can combine the current information in the knowledge base and use the E-GraphSAGE model to reason, complete the relationship between traffic nodes and attack types, and update the graph.

3.4.4 Construction of Attack Behavior Base

The attack behavior base constructs a domain attack graph and malicious behavior path traceability graph. The domain attack graph records the routing information in the network and the connection relationship between devices. Its connection status is the actual topological connection status in the network, which can provide the device's path features used by malicious behaviors and their spatial distribution.

Table 8 is the triple information of the domain attack graph. The specific content of the 11 features involved in the attributes is in Table 3. The domain topology graph provides input for the malicious behavior path reasoning module.

Table 8. Composition of the domain attack graph

Entity	Attributes	Relation
router	IP address, 11 features	Associate other entities in the network
Host	IP address, 11 features	Associate entities in the network

Table 9. Composition of malicious behavior path traceability graph

Entity	Attributes	Relation
Router	IP address, attack path type	Associate nodes on the same attack path
Host	IP address, attack path type	Associate nodes on the same attack path

The malicious behavior path tracing Graph is a collection of all malicious paths obtained by the malicious behavior path reasoning module, which records and displays all possible paths where malicious behaviors occur. Table 9 is the triple information of the malicious behavior path tracing graph.

The malicious behavior path traceability graph can reflect the attack path and attack method. Based on the known attack path and attack method, it can evaluate the security risk of the system or network and identify the lack or insufficiency of security control.

3.5 Reasoning Based on Knowledge Base

This section proposes relevant graph neural network algorithms based on three reasoning graphs: user behavior perception graph, malicious behavior association graph, and domain attack graph. It introduces the design principles of each reasoning module.

3.5.1 Malicious User Behavior Reasoning

Graph convolutional networks [22] are relatively straightforward to implement. When the network scale increases and large-scale users are connected, GCNs can quickly scale to accommodate reasoning over large-scale user graphs. Therefore, this section employs the GCN model to infer user behavior perception graphs. Using a two-layer graph convolutional network as an example, we apply ReLU and Softmax as the activation functions. Figure. 3 shows the schematic diagram of the process.

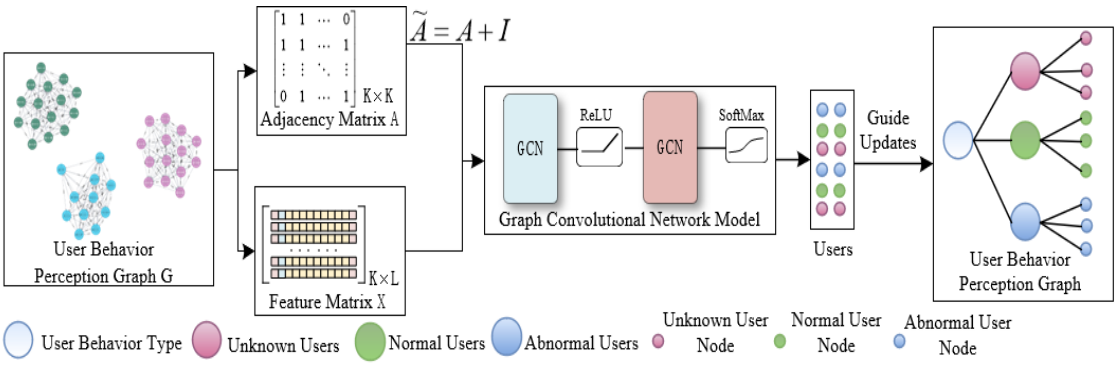

Figure. 3. Schematic diagram of malicious user behavior reasoning

We define the user behavior perception graph generated in the user behavior base as G, and an undirected graph G=(E, R) represents the overall topological structure encompassing all users. We consider each user as an entity e, forming the entity set $E_h = \{e_1, e_2, \ldots, e_K\}$.The connections between users in the Graph constitute the relationship set $R_h = \{r_1, r_2, \ldots, r_l\}$. Additionally, each user behavior feature is regarded as an attribute of the user, forming the attribute set $P_h = \{p_1, p_2, \ldots, p_l\}$.

$$\text{Global Profile} = \langle E_h, R_h, P_h \rangle \tag{5}$$

We derive the adjacency matrix A and the feature matrix X for the input of the graph convolution model from the user behavior perception graph. Assuming the number of users is K, the adjacency matrix A is initially formed by generating a K×K zero matrix. Based on the connections in the user behavior perception graph, if a connection exists between two users, the corresponding elements in the matrix are set to 1. The resulting matrix serves as the adjacency matrix A for input. The feature matrix X is a K×L matrix constructed from the attribute set P_h of each user, which characterizes the behavioral profiles of all users. Here, L denotes the number of features for each user, and as specified in Section 3.3.3, L=12.

We adjust the adjacency matrix A to $\tilde{A} = A + I$ to preserve the user's inherent features during the operation, where I is the identity matrix. The formula for user feature propagation using this adjusted model is as follow (6).

$$f(A, X) = \text{softmax}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\text{ReLU}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW^{(0)}\right)W^{(1)}\right) \tag{6}$$

In this formula, $W^{()}$ is the fully connected weight matrix of this layer, and $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ represents the Laplace normalization of the adjacency matrix, which aims to aggregate the behavioral features of surrounding users.

As the information propagates through each layer of the Graph Convolutional Network (GCN), the feature information of each user node is simultaneously transformed and transmitted to its connected neighboring nodes. Concurrently, we aggregate the feature information of neighboring users to facilitate the fusion of knowledge surrounding the node. Ultimately, we convert aggregated information into class probabilities through a Softmax transformation. We identify the category with the highest probability as the inferred type of abnormal user behavior.

We train the model using the stochastic gradient descent method to determine the optimal parameters for predicting the behavior category of user entities in the user behavior perception graph. Once trained, we deploy the optimal model in the user behavior base. We can then use this model for online inference to identify abnormal user behavior. Subsequently, we use the inference results to generate a user behavior map in the Neo4j graph database.

3.5.2 Malicious Behavior Type Reasoning

E-GraphSAGE [20] is a method for learning inductive representations of graph edges. In the malicious behavior association graph, edges denote flow communications between nodes, storing the flow features. Therefore, we employ the E-GraphSAGE model to infer the type of attack traffic. This approach transforms the task of inferring the type of malicious behavior to which the traffic belongs into the classification of edges within the malicious behavior association graph. Figure. 4 illustrates this process.

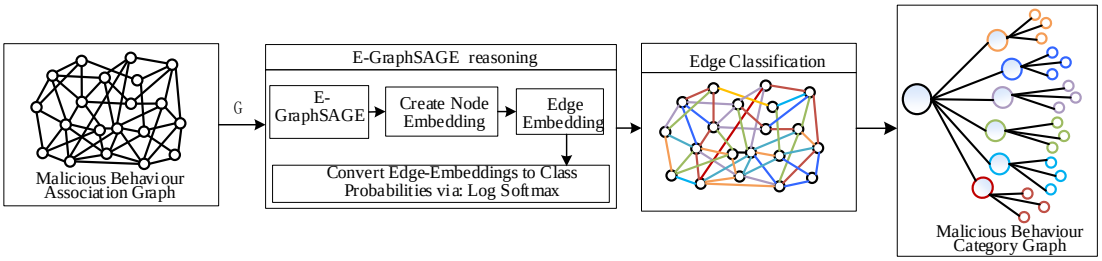

Figure. 4. Malicious behavior type reasoning diagram

The constructed malicious behavior association graph is represented as $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. This Graph provides edge features $\{e_{uv}, \forall uv \in \mathcal{E}\}$. Since the Graph lacks inherent node features, we initialize the node features using vector $x_v = \{1, \ldots, 1\}$(initial node embeddings). The dimension of each node vector is equal to the number of edge features in the flow.

The malicious behavior association graph G and its feature data are then input into the E-GraphSAGE model for inference. The process first aggregates the information of surrounding edges to generate node embeddings and creates an aggregate embedding $h^k_{\mathcal{N}(v)}$ of the sampled neighborhood edges at the kth layer, as shown in (7).

$$h^k_{\mathcal{N}(v)} = AGG_k\left(\left\{e^{k-1}_{uv}, \forall u \in \mathcal{N}(v), uv \in \mathcal{E}\right\}\right) \tag{7}$$

e^{k-1}_{uv} denotes $\mathcal{N}(v)$ the edges in the sampled domain of node u in the k-1 layer, and uv denotes the edge $\{\forall u \in \mathcal{N}(v), uv \in \mathcal{E}\}$, $\mathcal{N}(v)$ in the sampled domain of node v. AGG_k is the feature aggregation function.

Next, we concatenate the node's embedding from the previous layer with the aggregate embedding of the sampled neighborhood. This combined embedding is then processed using a trainable weight matrix (W^k), which serves as a trainable parameter of the model. The result is passed through the ReLU activation function to obtain the node embedding in the kth layer. Equation is as (8).

$$h^k_v = \sigma\left(W^k \cdot CONCAT\left(h^{k-1}_v, h^k_{N(v)}\right)\right) \tag{8}$$

The final node embedding depth is K, $Z_v = h^K_v$ And the node embeddings of u and v are concatenated to produce the edge embedding z^K_{uv}, as shown in (9).

$$z^K_{uv} = CONCAT\left(z^K_u, z^K_v\right), uv \in \mathcal{E} \tag{9}$$

We train the E-GraphSAGE model following the process above. We utilize the mean function for the aggregation function, which computes the element-wise mean of the edge features in the sampled neighborhood. We define the mean aggregator function in E-GraphSAGE in (10).

$$h^k_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v), uv \in \mathcal{E}} \frac{e^{k-1}_{uv}}{|N(v)|_e} \tag{10}$$

$|N(v)|_e$ denotes the number of edges in the sampled neighborhood. When implementing, we select complete neighborhood sampling, meaning that we aggregate the edges in the neighborhood of informative nodes from the whole set.

During the training process, we utilize the cross-entropy loss function and the Adam optimizer to perform gradient descent during the backpropagation phase. We pass the edge embedding through a Softmax layer, facilitating the training and optimization of the model parameters. After adjusting the model parameters to achieve the optimal model, we deploy the model in the traffic behavior base as a malicious behavior type reasoning model.

We perform type inference on the edges representing flows in the malicious behavior association graph using this inference model. We calculate edge embeddings and convert them into class probabilities using the final Softmax layer. We determine the flow type by comparing these probabilities, thus completing the inference process. Following the completion of the reasoning, we update the results in the malicious behavior category graph within the Neo4j graph database by the generation rules of the malicious behavior category graph.

3.5.3 Malicious Behavior Path Reasoning

Reference [30] analyzes 13 packet features widely applicable to classifying DDoS attack behaviors. Among these features, IP entropy can reflect the node status within a specific timeframe. By continuously monitoring changes in entropy values, it is possible to track the forwarding nodes involved in the DDoS attack accurately. Building on this concept, this section employs the 11 path state features described in Table 3 to characterize the DDoS attack path where each node is located. By connecting the nodes on the same attack path, we can reconstruct the actual path experienced by the attack.

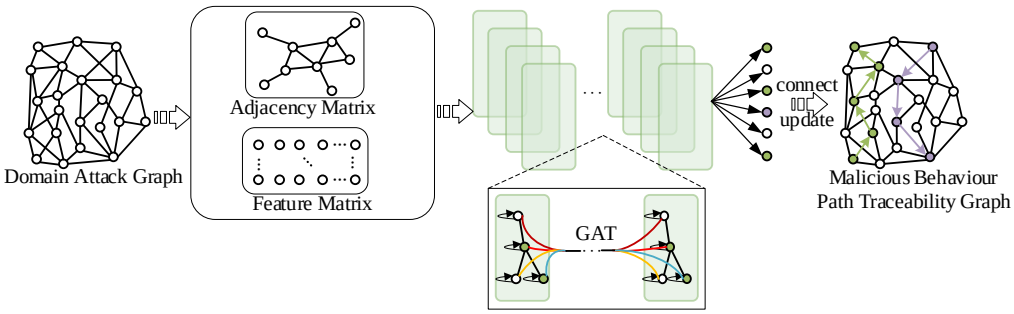

Figure. 5. Malicious behavior path reasoning process

Since each node in the network topology may play a distinct role in the path of malicious behavior, we cannot generalize their roles in the reasoning process. The relationship weight information between network nodes warrants careful consideration. Therefore, this section employs the Graph Attention Network (GAT) model [21] to infer the type of attack path experienced by each node. Figure. 5 illustrates the overall process.

The path state feature set of the nodes in the domain attack topology graph is denoted as $h = \{h_1, h_2, \dots, h_n\}, h_i \in \mathbb{R}^F$. Here, n represents the total number of attacks forwarding nodes, and F represents each node's path state feature dimension, which is 11 in this context. Following a similar method to the malicious user behavior reasoning module, we obtain the adjacency matrix and feature matrix from the domain attack graph as the input for the graph attention model. Then, we use the model for reasoning.

Within the model, for the connected nodes i and j, the calculation formula of the unnormalized attention correlation coefficient $e_{i,j}$ is as (11).

$$e_{i,j} = \text{LeakReLu}\left(a\left(Wh_i \parallel Wh_j\right)\right) \tag{11}$$

In the above formula, $W \in \mathbb{R}^{F' \times F}$ is a learnable shared linear transformation matrix. The symbol ‖ represents concatenation, indicating that we need to concatenate the features of the current node and its neighboring nodes. $a \in \mathbb{R}^{2F' \times 1}$ is a single-layer feedforward neural network used to calculate the similarity coefficient of the concatenated features. Finally, an activation function is applied to compute the normalized correlation coefficient.

We calculate the attention correlation coefficient for each topological node and its related neighbor nodes. The normalized mutual attention correlation coefficient $\alpha_{i,j}$ s obtained using the softmax function. The equation is as (12).

$$\alpha_{i,j} = \mathrm{softmax}(e_{i,j}) = \frac{\exp(e_{i,j})}{\sum_{K \in N_i}(e_{i,k})} = \frac{\exp\left(\mathrm{LeakReLu}\left(a(Wh_i \parallel Wh_j)\right)\right)}{\sum_{K \in N_i}\left(\mathrm{LeakReLu}(a(Wh_i \parallel Wh_k))\right)} \tag{12}$$

In this context, the mutual attention correlation coefficient $\alpha_{i,j}$represents the importance of node i to node j.

After we obtain the mutual correlation coefficients of all forwarding nodes in the global topology, we calculate the feature representation of the current node in the new dimensional space, as shown in (13).

$$h_i^{l+1} = \sigma\left(\sum \alpha_{i,j} W h_j^l\right) \tag{13}$$

Where $\sigma(\cdot)$ represents the activation function used by the current layer. This paper uses the Relu function as the activation function for feature forward propagation during the forward propagation process. The specific category of the node is output through (14).

$$y = \mathrm{argmax}(\mathrm{softmax}(H)) \tag{14}$$

H represents the node embeddings from the final layer of the model, which indicate the predicted DDoS path type, signifying that the node is part of this attack path.

Based on the principles above, we train the optimal model. Then, we use this optimal model for online reasoning of the node path state types. Subsequently, by referencing the topological structure of the domain attack graph, nodes on similar attack paths are connected in a directed manner to form a malicious behavior path traceability graph, as illustrated in Figure 5. This approach enables the complete description of the attack path experienced by each attack, thereby facilitating the reasoning of the malicious behavior path.

3.5.4 Reasoning Module Related Applications

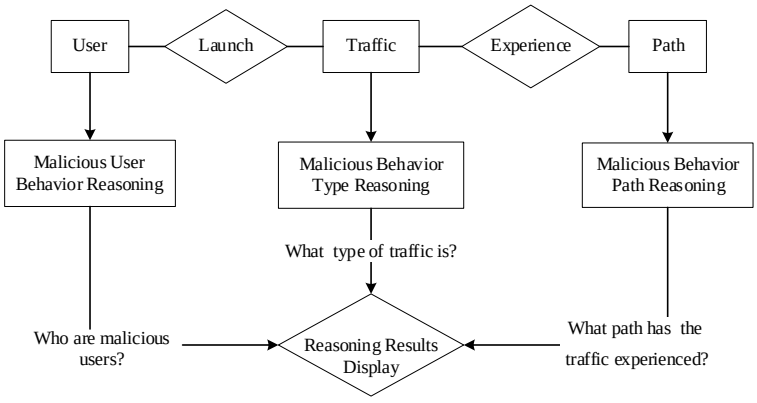

Figure. 6. Analysis of the application of reasoning modules

Figure. 6 illustrates the analytical diagram of the associated application of the reasoning module. The results

from the malicious user behavior reasoning module form a user behavior mapping graph, identifying the malicious user. The malicious behavior type reasoning module generates a malicious behavior category graph, indicating the type of traffic generated by the malicious user. The malicious behavior path reasoning module produces a path traceability graph, reconstructing the path the malicious user takes.

We establish a comprehensive malicious behavior analysis mechanism by analyzing the user behavior, traffic behavior, and attack path behavior data of the same user using these three reasoning modules and correlating the reasoning results. This mechanism encompasses "malicious user behavior monitoring," "malicious behavior type detection," and "malicious behavior path tracing."

4 Experimental Results

This section describes the experimental environment, presents the evaluation metrics, and uses the launch of network layer DDoS attacks as an example to verify the implementation of the knowledge base for malicious behavior analysis mechanisms. Subsequently, the effectiveness of the three reasoning models is evaluated and compared with reasoning methods in similar scenarios.

4.1 Experimental environment

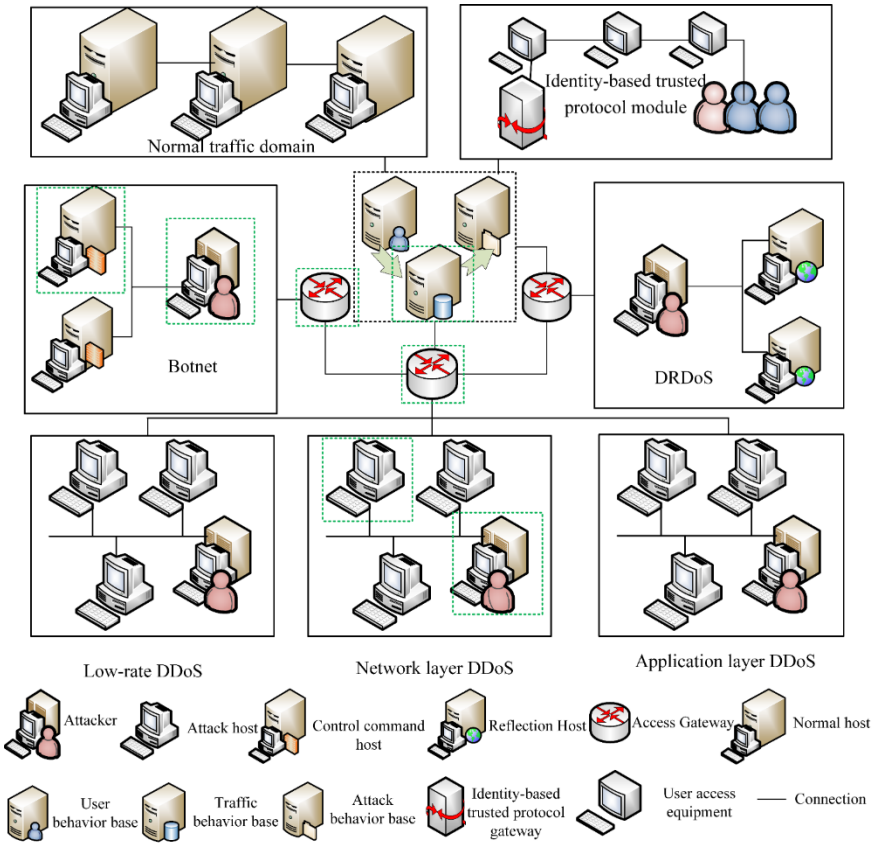

Figure. 7. system topology

As shown in Figure. 7, we build the experimental environment on the virtual platform of VMware vSphere. We installed 24 hosts with identical conFigureurations to form the prototype system. Each host has a 40GB disk, 8GB of memory, and an Ubuntu 18.04 operating system. Among these, we utilized 21 hosts to create five different types of DDoS attack domains and normal traffic domains, including low-rate DDoS, application layer DDoS, network layer DDoS, botnet, and DRDoS [31]. Three hosts served as knowledge bases, each

deploying a Neo4j graph database to generate a knowledge graph. The user behavior knowledge base is linked to the identity-based trusted protocol module and receives user behavior knowledge from this module. Additionally, Scapy and CICFlowMeter tools were deployed in each attack domain, monitored by the traffic behavior knowledge base and the attack behavior knowledge base.

4.2 Evaluation indicators

In evaluating the reasoning model effect, precision, recall, and F1 score are used to evaluate the inference performance of the three models.

In comparing methods, we use accuracy, malicious user recognition rate, and recognition time as evaluation metrics to assess the monitoring effectiveness of the malicious user behavior reasoning module on abnormal users. Accuracy represents the proportion of samples correctly classified by the model out of the total number of samples. The malicious user recognition rate indicates the proportion of malicious users accurately monitored compared to the network's total number of malicious users. Recognition time measures the system's duration to associate and infer all abnormal users.

To verify the detection effectiveness of the malicious behavior reasoning module on traffic types, we consider additional metrics beyond accuracy, namely the malicious traffic detection rate (DR) and the malicious traffic detection capability (DC).

The malicious traffic detection rate (DR) represents the proportion of correctly detected malicious traffic to the total malicious traffic and is defined as follows (15).

$$\mathrm{DR} = \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} A_i} \tag{15}$$

Where i represents the category of malicious traffic，n represents the total number of，T_i is the number of correctly detected traffic instances in the malicious traffic category，A_i is the total number of traffic instances in the malicious traffic category.

Malicious traffic detection capability (DC) is the ratio of the total number of malicious traffic instances to the number of undetected malicious traffic instances. This metric reflects the magnitude of detected malicious traffic, and its definition is shown in (16).

$$\mathrm{DC} = \frac{1}{1 - \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} A_i}} \tag{16}$$

When comparing the malicious behavior path reasoning effect, we introduce the malicious path recognition rate (MPRR) as an evaluation index, which is the ratio of the number of correct paths connected by the inferred path nodes to the total number of malicious paths in the topology, it is expressed as follows (17).

$$\mathrm{MPRR} = \frac{\sum_{i=1}^{n} P_i}{\sum_{i=1}^{n} TM_i} \tag{17}$$

Where i represents the category of malicious paths,n represents the total number of malicious path categories，P_i is the number of correct paths connected according to the inference results, TM_i is the total number of malicious paths.

4.3 Online verification of reasoning functions based on knowledge base

To verify the malicious behavior analysis capabilities of the constructed CSBKB framework, we simulate a scenario where a host with an IP address of 23.1.0.13 in the network layer DDoS attack domain launches

malicious user behavior and network layer DDoS attacks. Additionally, we select a host in the network layer DDoS attack domain as a forwarding node, a host in the botnet attack domain as a forwarding node, and another host as the victim. We also utilize two routers connected to these two domains and the traffic behavior base host as forwarding nodes. These nodes are highlighted explicitly with green boxes in Figure. 7.

We test the knowledge base's graph construction, malicious behavior reasoning, and graph update functions. During online verification, the experimental environment provides a limited number of user nodes and network topology nodes. We only use a small amount of data for testing to demonstrate more precise results.

4.3.1 Reasoning knowledge graph construction

When malicious behavior is initiated, the data layer sends the collected data to the knowledge layer. The various knowledge bases in the knowledge layer construct three types of reasoning graphs based on the data: user behavior perception graph, malicious behavior association graph, and intra-domain attack graph.

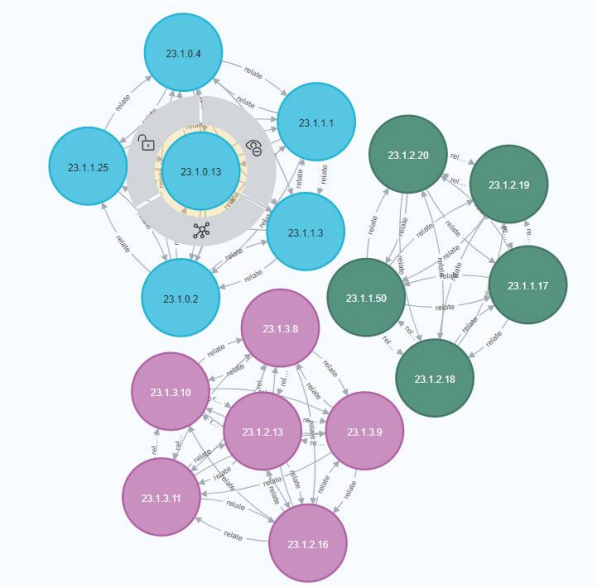

Figure. 8. Partial update of the user behavior perception graph

The user behavior base receives user behavior data from the data layer and establishes connections between the monitored user (23.1.0.13) and other nodes. It stores the features of each user as attributes on the nodes, forming the user behavior perception graph illustrated in Figure. 8. This Graph provides the structure and user features required for reasoning in the malicious user behavior reasoning module.

The traffic behavior base constructs each flow as an edge between two nodes based on the traffic behavior data sent by the data layer. Here, the flow of the host 23.1.0.13 has no label, so the name of the edge is " unknown. " Figure. 9 shows the malicious behavior association graph.

The initial topological information is stored in the attack behavior base. When malicious behavior is detected, the data layer transmits the node features with updated path state features to the attack behavior base. The attack behavior base then updates these features as attributes of the corresponding nodes, forming the intra-domain attack graph depicted in Figure. 10. The right side of the Figure represents the path state features.

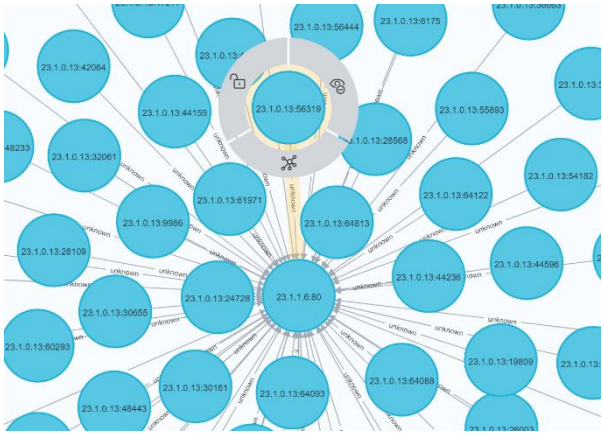

Figure. 9. Partial update of the malicious behavior association graph

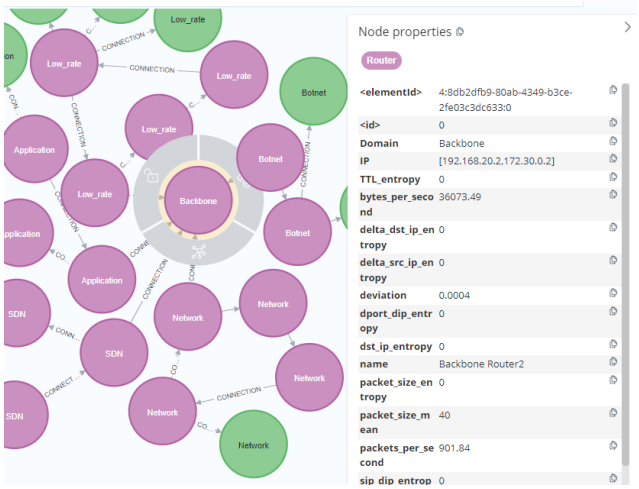

Figure. 10. Partial update of the domain attack graph

4.3.2 Malicious behavior reasoning module runs

```
(mytorch) root@ip707:/home/lddos_rl/user_GCN# python eval.py
loss:0.0036828566808253527    acc:1.0
                  precision    recall  f1-score   support

      normal_user     1.0000    1.0000    1.0000         5
unrecognised_user     1.0000    0.8000    0.8889         5
    abnormal_user     0.8571    1.0000    0.9231         6

         accuracy                         0.9375        16
        macro avg     0.9524    0.9333    0.9373        16
     weighted avg     0.9464    0.9375    0.9364        16

混淆矩阵:
[[5 0 0]
 [0 4 1]
 [0 0 6]]
The inference results about users have been saved to the file 'user_collection'....
User behaviour Mapping Graph being updated...
```

Figure. 11. Malicious user behavior inference results

The reasoning module employs the trained graph neural network model to perform inference, saving the predicted users, traffic types, and node path status labels into a "collection" file. The module subsequently updates the corresponding display class graph based on this file.

Figure. 11 presents the results of the malicious user behavior reasoning. The confusion matrix indicates that

six malicious users were detected. Figure. 12 shows the results of the malicious behavior type reasoning, with the confusion matrix revealing that one application layer attack flow and fifty-four network-layer attack flow instances were correctly identified. Figure. 13 depicts the results of the malicious behavior path reasoning, where the confusion matrix indicates that we identified seven nodes participating in network layer DDoS attacks.

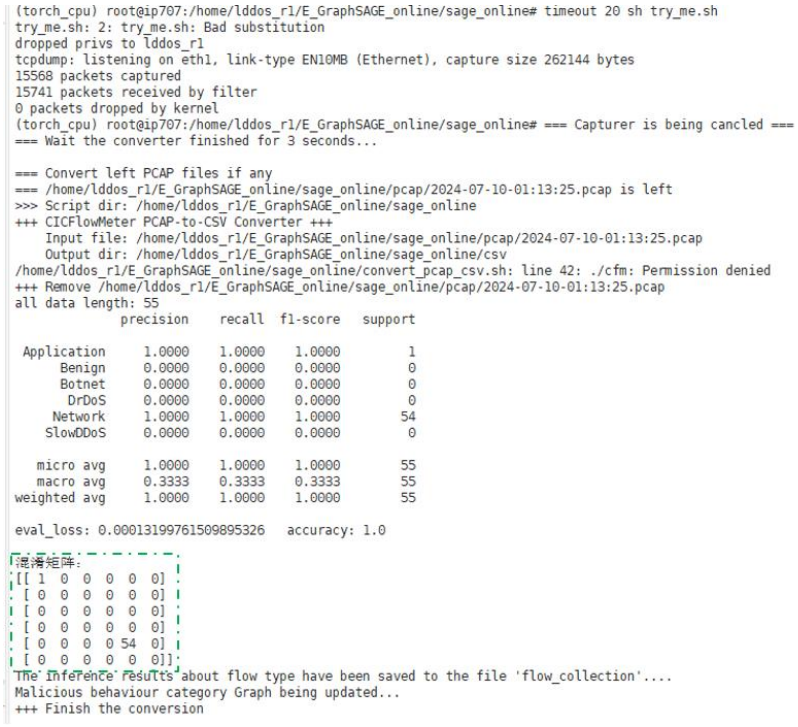

Figure. 12. Malicious behavior type inference results

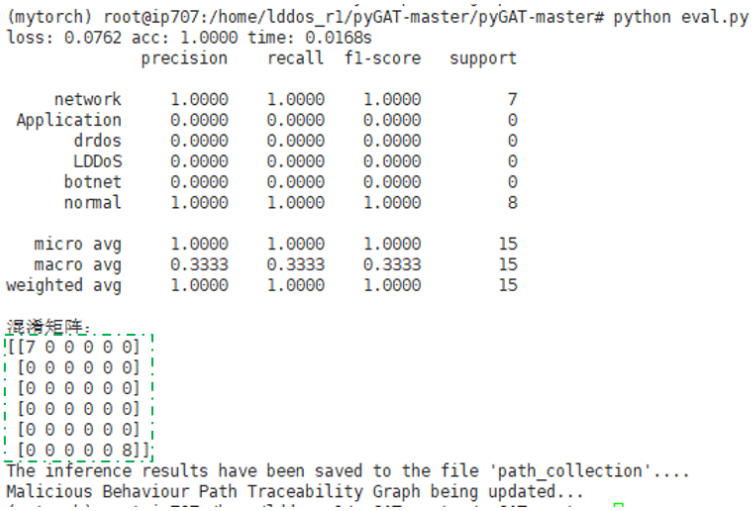

Figure. 13. Malicious behavior path inference results

4.3.3 Display knowledge graph update

The reasoning results from the previous section are transmitted to the cyber security behavior knowledge base via the knowledge update interface and subsequently updated in the display class map of the knowledge base. Figure. 14 provides a partial view of the user behavior map, where the user with IP address 23.1.0.13 is

identified as an abnormal user by the reasoning module and is associated with the "abnormal" node, indicating the user's abnormal status.

Figure. 15 illustrates a partial update of the malicious behavior category graph. The relationship between the flow node and the attack type is established based on the inferred flow type. As depicted in the Figure, the flow node associated with the IP address 23.1.0.13 points to the "NetworkDDoS" type, indicating that the verified traffic type belongs to a network layer DDoS attack.

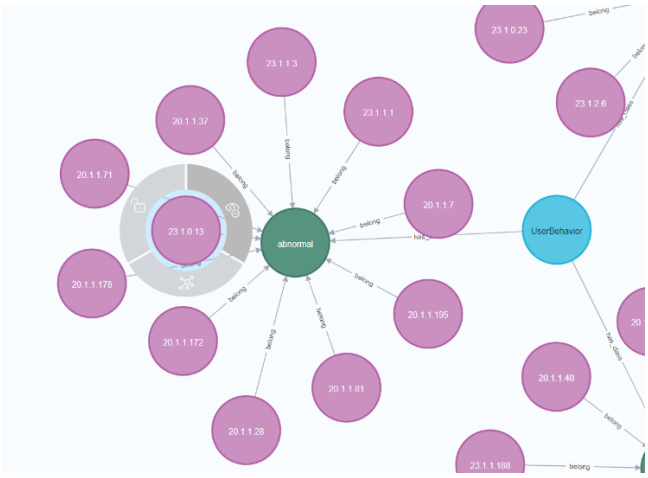

Figure. 14. Partial update of the user behavior mapping graph

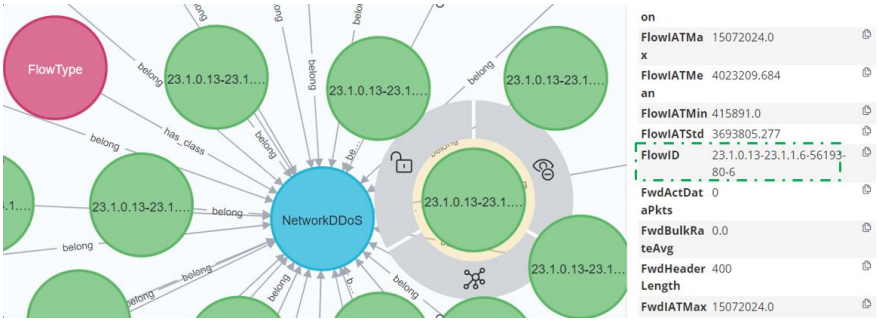

Figure. 15. Partial update of the malicious behavior category graph

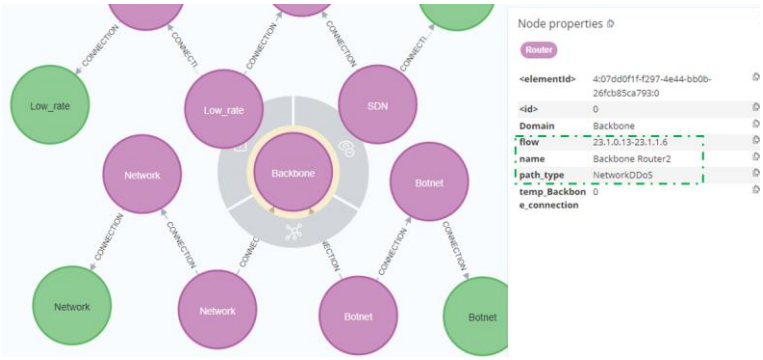

Figure. 16. Partial update of the malicious behavior path traceability graph

Figure. 16 is a partial update of the malicious behavior path traceability graph. Below is a path formed by seven network layer DDoS path nodes, as indicated in the malicious behavior path inference results. The properties of a node in the path are queried, with the "flow" and "path_type" columns in the property bar on

the right displaying the source IP address, destination IP address, and path type of this attack path. This information indicates that the host with IP address 23.1.0.13 experienced the attack path.

Based on the process outlined in Section 4.3 and the query results of the class graph, we have determined that the user with IP address 23.1.0.13 is abnormal. The attack launched by this user is identified as a network layer DDoS attack, and the path it traversed has been mapped. The entire process outlined above demonstrates the implementation of a comprehensive malicious behavior analysis mechanism, encompassing "malicious user behavior monitoring, malicious behavior type detection, and malicious behavior path tracing."

4.4 Reasoning Model and Reasoning Effect Analysis

4.4.1 Reasoning model parameter settings

This study selected 10,000 user data records from the user behavior library, 20,000 traffic data records from the traffic behavior library, and 15,000 attack behavior data records from the attack behavior library for the experiments. The dataset was divided into training and validation sets in an 8:2 ratio. We employed a warm-up strategy during the training process to adjust the learning rate dynamically, with the maximum learning rate set to 0.01. The other parameter settings of the training model [20,21,22] are in Table 10.

Table 10. Parameter settings of the model

Model	Dropout	Batch	Rounds
Malicious user behavior reasoning model	0.3	10	500
Malicious behavior type reasoning model	0.2	20	500
Malicious behavior path reasoning model	0.3	18	400

4.4.2 Reasoning Model Effect

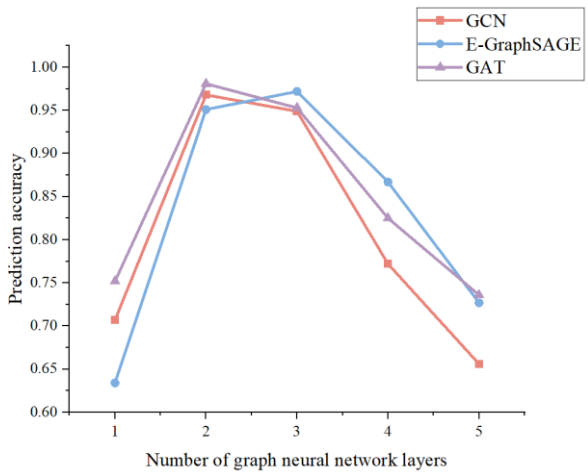

Figure. 17. Prediction accuracy of network models with different numbers of layers

The number of layers in a graph neural network is a crucial factor influencing the model's performance. To determine the optimal number of layers for ensuring efficient model reasoning, we maintained all other conditions constant except for the number of layers. We evaluated the prediction accuracy of the three reasoning models within the prototype system. As depicted in Figure. 17, the results indicate that the graph convolution model and the graph attention model achieve their highest prediction accuracy of 0.97 and 0.98, respectively, with a network layer count of 2. The E-GraphSAGE model has the highest prediction accuracy of 0.97 when the network layer number is 3.

Additionally, it was observed that increasing the number of layers results in a decline in accuracy beyond this optimal point. We may attribute this decline to the exponential increase in the number of neighbor nodes associated with each node, which interferes with the correlation between local nodes. Consequently, in

subsequent experiments, the chosen number of layers for the graph convolution, E-GraphSAGE, and graph attention models are 2, 3, and 2, respectively.

Table 11. Model results

Model	Category	Precision	Recall	F1 score
Malicious Users Behavioral Reasoning Model	Normal user	0.9735	0.9778	0.9756
	Unknown user	0.9510	0.9489	0.9499
	Abnormal user	0.9755	0.9733	0.9744
Malicious behavior Type Reasoning Model	Low-rate DDoS	0.9745	0.919	0.9460
	Normal traffic	0.9388	0.9537	0.9449
	Botnet DDoS	0.9418	0.9893	0.9650
	DRDoS	0.9901	1	0.9950
	Network layer DDoS	0.9990	1	0.9995
	Application layer DDoS	0.9798	0.9598	0.9697
Malicious Behavior Path Reasoning Model	Application layer DDoS	0.9775	0.9649	0.9712
	Normal traffic	0.9478	0.9562	0.9520
	Botnet DDoS	0.9513	0.9716	0.9613
	DRDoS	0.9867	0.9783	0.9825
	Network layer DDoS	0.9880	1	0.9940
	Low-rate DDoS	0.9734	0.9848	0.9791

Based on the above model, we test the reasoning effect. Table 11 shows the specific results.

The three indicators for various types of users in the malicious user behavior reasoning model all exceed 0.94. However, the indicators for unknown types of users are slightly lower than those for the other two types. We may attribute the cause to the feature patterns of unknown users, which lie between normal and abnormal users. When the boundaries between these patterns are not distinct, unknown users are more likely to be misclassified as normal or abnormal users.

The three indicators in the malicious behavior type inference model are above 0.92. The precision and recall of normal traffic and low-rate DDoS are lower than other types. This discrepancy may be due to the relatively slow attack rate of low-rate DDoS, whose attack characteristics closely resemble normal traffic, leading to mutual misclassification.

In the malicious behavior path reasoning model, the overall indicators for using the model to infer the attack path type to which a node in the topology belongs are all above 0.94, with no significant differences among the individual indicators.

4.4.3 Comparison of Reasoning Methods

In this section, we evaluated the reasoning methods of the three reasoning modules in the system and set different evaluation indicators based on the scenarios. We obtained the following results by taking the average after ten tests.

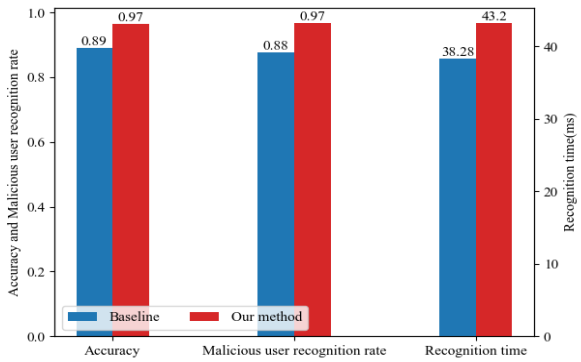

Figure. 18. Comparison of Malicious User Behavior Reasoning Methods

To evaluate the monitoring effectiveness of the malicious user behavior reasoning module for detecting abnormal users, we compared our reasoning model with the method proposed in the literature [14], which uses a random forest (RF) approach to analyze user behavior and detect anomalies. Figure. 18 illustrates the results. Compared to the benchmark solution (Baseline), our approach demonstrates improvements in accuracy, malicious user recognition rate, and recognition time by 9%, 10%, and 12.7%, respectively. It proves that our solution is more effective in identifying potential abnormal users through reasoning based on user associations within the knowledge graph.

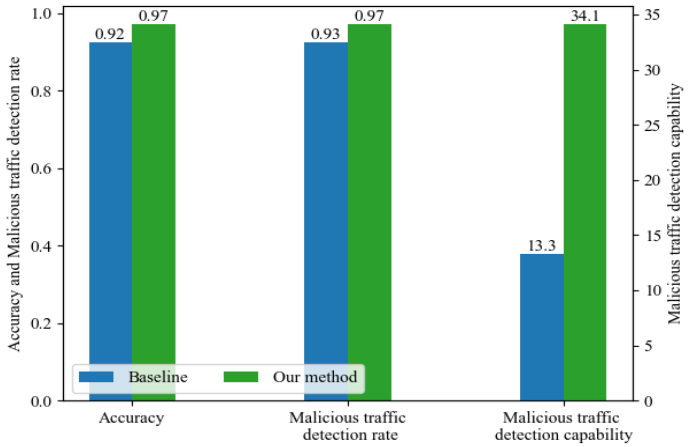

Figure. 19. Comparison of malicious behavior type reasoning methods

To verify the detection effectiveness of the malicious behavior inference module on traffic types, we evaluated the accuracy and two additional indicators: malicious traffic detection rate (DR) and (DC). Figure. 19 compares this paper's malicious behavior type inference scheme and the detection method utilizing E-GraphSAGE as proposed in the literature [20]. The quantitative results demonstrate that our method improves accuracy and malicious traffic detection rate by 5.4% and 4.3%, respectively, compared to the benchmark literature (Baseline). Moreover, our method's malicious traffic detection capability is more than 2.5 times that of the benchmark. We attribute these results to our method's ability to fully leverage the structured information and feature knowledge within the knowledge graph, thereby enhancing the model's reasoning capabilities.

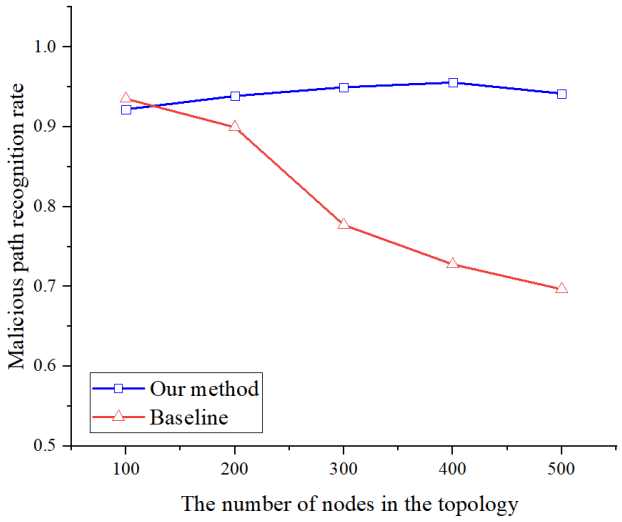

Figure. 20. Comparison of Malicious Behavior Path Reasoning Methods

When comparing the effectiveness of malicious behavior path reasoning, we constructed domain attack

graphs with 100, 200, 300, 400, and 500 nodes in the attack behavior library for testing. The number of malicious paths in these five graphs was set to 4, 8, 12, 16, and 20, respectively, with each path involving six participating nodes. We compared our method with finding the attack path based on the node's threat degree using a greedy algorithm, as described in the literature [15] (Baseline).

The results, shown in Figure. 20, indicate that our method consistently achieves a malicious path recognition rate above 0.92, demonstrating a higher and more stable recognition rate. Our method benefits from a broader knowledge scope as the topology size increases, resulting in improved malicious path recognition rates within topologies of 100 to 400 nodes.

5 Conclusion

This paper proposes a CSBKB framework for malicious behavior analysis, using DDoS attacks as a case study to demonstrate the construction of the CSBKB. The framework designs reasoning modules for the knowledge base using graph neural networks to address questions in malicious behavior analysis, such as "Who attacked me, in what way, and what is the attack path?" Experimental results indicate that the proposed knowledge base integration framework can effectively analyze malicious behavior, achieving an accuracy of over 0.97 in identifying abnormal users, over 0.97 in inferring DDoS attack types, and a malicious behavior path recognition rate above 0.92.

We select several appropriate graph neural network models to design reasoning modules based on research needs and scenarios. It is important to note that these graph neural network models are not the only options available. As the research progresses, we will explore more efficient reasoning models and employ a combination of multiple algorithms to enhance system robustness. Additionally, improving the responsiveness and fluency of the knowledge base system in malicious behavior analysis is a vital issue to be addressed in future work.

Acknowledgement

This paper was supported by the National Key R&D Program of China under Grant No.2018YFA0701604, and NSFC under Grant No. 62341102

References

[1] El Kafhali S, El Mir I, Hanini M, “Security threats, defense mechanisms, challenges, and future directions in cloud computing,” Archives of Computational Methods in Engineering, vol. 29, no.1, pp. 223-246, January. 2022. Article (CrossRef Link)

[2] Guo Q, Wang X, Wu Y, “Online knowledge distillation via collaborative learning,” IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp. 11020-11029, June. 2020. Article (CrossRef Link)

[3] S. Ji, S. Pan, E. Cambria, P. Marttinen and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 2, pp. 494-514, Feb. 2022. Article (CrossRef Link)

[4] Sikos, Leslie F, "Cybersecurity knowledge graphs," Knowledge and Information Systems 65, vol. 65, pp. 3511-3531. April. 2023. Article (CrossRef Link)

[5] Liu, Kai, Fei Wang, Zhaoyun Ding, Sheng Liang, Zhengfei Yu and Yun Zhou, "A review of knowledge graph application scenarios in cyber security," ArXiv, vol. abs/2204.04769, 2022. Article (CrossRef Link)

[6] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song and J. Wang, "A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs," IEEE Access, vol. 10, pp. 75729-75741, 2022.

[7] Article (CrossRef Link)

[8] WANG Q, LIU J, LUO Y F, “Knowledge base completion via coupled path ranking,” Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1308-1318, Aug. 2016. Article (CrossRef Link)

[9] ZHANG W, PAUDEL B, WANG L, “Iteratively learning embeddings and rules for knowledge graph

reasoning," Proceedings of the World Wide Web Conference, pp. 2366-2377, May 2019. Article (CrossRef Link)
[10] Chen Z, Dong N, Zhong S, "Research on the power network security vulnerability expansion attack graph based on knowledge map," Information Technology, vol. 46, no. 02, pp. 30-35, July 2024. Article (CrossRef Link)
[11] GAO Chen, ZHANG Xuan, LIU Hui, "Data and Knowledge-Driven Named Entity Recognition for Cyber Security," Cybersecurity, vol. 4, no. 1, pp. 1-13, May. 2021. Article (CrossRef Link)
[12] Rastogi, Nidhi, Sharmishtha Dutta, Mohammad Zaki, Alex Gittens and Charu Aggarwal, "TINKER: A framework for Open source Cyberthreat Intelligence," 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1569-1574, 2021. Article (CrossRef Link)
[13] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang and Z. Tian, "CSKG4APT: A Cybersecurity Knowledge Graph for Advanced Persistent Threat Organization Attribution," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 6, pp. 5695-5709, 1 June 2023. Article (CrossRef Link)
[14] Liu F, Li K, Song F, "Distributed DDoS attacks malicious behavior knowledge base construction," Telecommunications Science, vol. 37, no. 11, pp. 17-32, 20 November 2021.
[15] Article (CrossRef Link)
[16] M. Afshar, S. Samet and H. Usefi, "Incorporating Behavior in Attribute Based Access Control Model Using Machine Learning," 2021 IEEE International Systems Conference (SysCon), pp. 1-8, 2021. Article (CrossRef Link)
[17] Practicing Security Knowledge Graph, Together Advancing into Cognitive Intelligence, NSFOCUS, Beijing, China, 2021. Article (CrossRef Link)
[18] Zhang S, Li S, Chen P, Wang S, Zhao C, "Generating Network Security Defense Strategy Based on Cyber Threat Intelligence Knowledge Graph," In: Quan, W. (eds) Emerging Networking Architecture and Technologies, vol. 1696, pp. 507–519, 2022. Article (CrossRef Link)
[19] Zhang, Shuqin, Guangyao Bai, Hong Li, Peipei Liu, Minzhi Zhang, and Shujun Li. 2021. "Multi-Source Knowledge Reasoning for Data-Driven IoT Security," Sensors, vol. 21, no. 22, pp. 7579, 2021. Article (CrossRef Link)
[20] Gilliard E, Liu J, Aliyu AA, "Knowledge graph reasoning for cyber attack detection," IET Communications, vol. 18, no. 4, pp. 297-308, 26 February 2024. Article (CrossRef Link)
[21] LI Zixuan, JIN Xiaolong, LI Wei, "Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning," ACM. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 408-417, July 2021.
[22] Article (CrossRef Link)
[23] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher and M. Portmann, "E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT," NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1-9, 2022.Article (CrossRef Link)
[24] O. Liu, K. Li, Z. Yin, D. Gao and H. Zhou, "Multi-domain malicious behavior knowledge base framework for multi-type ddos behavior detection," Intelligent Automation & Soft Computing, vol. 37, no.3, pp. 2955–2977, 2023. Article (CrossRef Link)
[25] Li K, Zhou H, Tu Z, "AT-GCN: A DDoS attack path tracing system based on attack traceability knowledge base and GCN," Computer Networks, vol. 236, no. 5 , pp. 110036, September 2023.
[26] Article (CrossRef Link)
[27] G. Zhang, J. Liu, G. Zhou, K. Zhao, Z. Xie and B. Huang, "Question-Directed Reasoning With Relation-Aware Graph Attention Network for Complex Question Answering Over Knowledge Graph," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 32, pp. 1915-1927, 2024. Article (CrossRef Link)
[28] Sharma, A., Gupta, B.B., Singh, A.K., "Advanced Persistent Threats (APT): evolution, anatomy, attribution and countermeasures," J Ambient Intell Human Comput 14, vol. pp. 9355–9381, 06 May 2023. Article (CrossRef Link)
[29] Song H, Tu Z, Qin Y, "Blockchain-Based Access Control and Behavior Regulation System for IoT," Sensors, vol. 22, no. 21, pp. 8339-8367, October 2022. Article (CrossRef Link)
[30] S. Deng, M. Li and H. Zhou, "Dynamic security sfc branching path selection using deep reinforcement learning," Intelligent Automation & Soft Computing, vol. 37, no.3, pp. 2919–2939, 2023. Article (CrossRef Link)
[31] Lashkari A H. CICFlowMeter[EB/OL]. [2022-01-05]. Article (CrossRef Link)
[32] Scapy. [EB/OL]. [2023.01.30]. Article (CrossRef Link)

[33] Wei P, Zhu W, Zhao Y, Fang P, Zhang X, Yan N, Zhao H, “Extraction of Kenyan Grassland Information Using PROBA-V Based on RFE-RF Algorithm,” Remote Sens. 2021, vol. 13, no. 23, pp. 4762. Article (CrossRef Link)
[34] Gu Y, Li K, Guo Z, “Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm,” IEEE Access, vol. 7, pp. 64351-64365, May 2019.Article (CrossRef Link)
[35] Yin Z, Li K, Bi H, “Trusted Multi-Domain DDoS Detection Based on Federated Learning,” Sensors, vol. 22, no. 20, pp. 7753, 2022. Article (CrossRef Link)

Author’s Biography

Keke Feng received the B.S. degree in Communication Engineering from Beijing Jiaotong University (BJTU), Beijing, China in 2022, and she is pursuing a master's degree at the National Engineering Research Center of Advanced Network Technologies, BJTU. Her research interests include network security and artificial intelligence.

Huachun Zhou received a B.S. degree from the People’s Police Officer University of China in 1986. He received his M.S. in telecommunication automation and Ph.D. degrees in telecommunications and information systems from Beijing Jiaotong University in 1989 and 2008, respectively. Now, he is a professor at the National Engineering Research Center for Advanced Network Technologies at BJTU. He has authored more than 40 peer-reviewed papers and he is the holder of 17 patents. His main research interests are in the areas of Internet architecture, space-air-ground integrated networks, intelligent communication network, mobility management, mobile and secure computing, and routing protocols.

Weilin Wang received the B.S. degree in telecommunications engineering from Beijing Jiaotong University (BJTU), China, in 2020, and she is currently studying for the Ph.D. degree in information and telecommunications engineering. She joined the National Engineering Research Center for Advanced Network Technologies, BJTU. Her research interest includes the architecture of next generation internet, network security, and artificial intelligence.

Jingfu Yan received his B.Eng. degree from Anhui University in June 2018 and M.Eng. degree from Beijing Jiaotong University in June 2022. He is currently a Ph.D. candidate at the National Engineering Research Center of Advanced Network Technologies, Beijing Jiaotong University, Beijing, China. His research interests include network security, the next generation internet and intelligent network.

Xiaojing Fan received the B.S. degree in electronic information from Beijing Jiaotong University (BJTU), China, in 2023, and she is currently studying for the Ph.D. degree in information and communication engineering. She joined the National Engineering Research Center for Advanced Network Technologies, BJTU. Her research interest includes the architecture of next generation internet, satellite communications, and artificial intelligence.