Intelligent Automation
& Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.039995
Article

Multi-Domain Malicious Behavior Knowledge Base Framework for Multi-Type DDoS Behavior Detection

Ouyang Liu, Kun Li*, Ziwei Yin, Deyun Gao and Huachun Zhou

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China
*Corresponding Author: Kun Li. Email: kun_li@bjtu.edu.cn
Received: 28 February 2023; Accepted: 29 May 2023; Published: 11 September 2023

Abstract: Due to the many types of distributed denial-of-service attacks (DDoS) attacks and the large amount of data generated, it becomes a challenge to manage and apply the malicious behavior knowledge generated by DDoS attacks. We propose a malicious behavior knowledge base framework for DDoS attacks, which completes the construction and application of a multi-domain malicious behavior knowledge base. First, we collected malicious behavior traffic generated by five mainstream DDoS attacks. At the same time, we completed the knowledge collection mechanism through data pre-processing and dataset design. Then, we designed a malicious behavior category graph and malicious behavior structure graph for the characteristic information and spatial structure of DDoS attacks and completed the knowledge learning mechanism using a graph neural network model. To protect the data privacy of multiple multi-domain malicious behavior knowledge bases, we implement the knowledge-sharing mechanism based on federated learning. Finally, we store the constructed knowledge graphs, graph neural network model, and Federated model into the malicious behavior knowledge base to complete the knowledge management mechanism. The experimental results show that our proposed system architecture can effectively construct and apply the malicious behavior knowledge base, and the detection capability of multiple DDoS attacks occurring in the network reaches above 0.95, while there exists a certain anti-interference capability for data poisoning cases.

Keywords: DDoS attack; knowledge graph; multi-domain knowledge base; graph neural network; federated learning

1 Introduction

With the rapid development in the field of 5G technology, more and more terminal devices are connected to the Internet, and the unrestricted communication between massive terminal devices leads to the network security issue becoming a factor that cannot be ignored [1]. Due to its concealment and high efficiency, DDoS has become the most common attack method used by network attackers, which seriously endangers the security of the Internet. DDoS attacks can be classified into multiple subtypes in terms of protocol level and traffic characteristics, for example, there are network layer DDoS attacks [2], application layer DDoS attacks [3], low-rate DDoS (LDDoS) [4], distributed reflection denial of service attacks (DRDoS) [5] and botnet DDoS attacks [6]. DDoS attacks tend to send more useless traffic, so the management and application of the generated malicious behavior knowledge become a challenge.

Knowledge bases can present data in the form of knowledge graphs [7], which can fully reflect the potential structural relationships among data. To effectively manage the knowledge of malicious behaviors generated by multiple types of DDoS attacks, we can use distributed knowledge base techniques. Structurally, knowledge graphs exist in the form of directed attribute graphs and organize data in the triad of "entity-relationship-entity" [8]. How to apply malicious behavior knowledge to achieve DDoS detection and complete knowledge transfer between multiple knowledge bases becomes a difficult point in applying malicious behavior knowledge.

Traditional DDoS detection is mainly based on statistical and machine learning methods [9], which learn and identify traffic characteristics to achieve DDoS behavior detection. To evade detection, attackers continuously update their attack strategies to make the characteristics of traffic more similar to normal traffic [10]. Graph neural network is a generalized deep learning model based on graph structure. During training, it requires the input of feature matrix and adjacency matrix of sample nodes. So it can take into account the feature information and structural information of the sample nodes in the network space. Therefore graph neural networks are more effective in detecting malicious communication behaviors occurring in the network. Meanwhile, the proposed federated learning is a good solution to the knowledge-sharing problem among multiple malicious behavior knowledge bases [11]. As a distributed machine learning technique, federated learning can ensure that each participant completes multi-party joint modeling without sharing its dataset, which can greatly protect the security of data [12]. Each participant only needs to interact with the locally trained gradients or models obtained from the aggregation server to be able to update the global model during each communication. Combining graph neural networks and federated learning becomes a novel approach to applying knowledge of malicious behavior.

Therefore, this paper proposes a multi-domain malicious behavior knowledge base framework for DDoS behavior detection to solve the above problems, and the contributions of this paper are as follows:

1. We collected malicious behavior traffic generated by five mainstream DDoS attacks, including network layer DDoS, application layer DDoS, low-rate DDoS, DrDoS, and botnet. Then we completed the knowledge collection mechanism by data preprocessing and malicious behavior dataset design.
2. We designed a malicious behavior category graph and malicious behavior structure graph for the traffic feature information and spatial structure information of DDoS attacks and completed the knowledge learning mechanism based on a graph neural network model.
3. We build a multi-domain malicious behavior knowledge base detection system. We complete the knowledge-sharing mechanism of malicious behavior knowledge bases in multiple network domains through federated learning, and the process protects the data privacy of each knowledge base.
4. We build a malicious behavior repository to store the constructed knowledge graphs and a detection model repository to store the local graph neural network models and global federated learning models to complete the knowledge management mechanism.

The study in this paper consists of five sections. Section 2 discusses the related work. Section 3 gives a detailed description of the framework proposed in the text. Section 4 gives the detailed results and analysis of the experiments. Section 5 concludes the study of this paper and provides some suggestions for future work.

2 Related Work

In this section, we review related work on knowledge graphs, graph neural networks, and federated learning for DDoS detection and defense.

Knowledge graphs allow effective management and deep mining of graph-structured data. The literature [13] provided an overview of the basic concepts and definitions of knowledge inference and inference methods for knowledge graphs. The authors classified inference methods into three categories: rule-based inference, distributed representation-based inference, and neural network-based inference. Also, the authors review relevant applications of knowledge graph inference and discuss future challenges of knowledge graph inference. The literature [14] proposed a context-aware approach based on a knowledge base to handle intrusions generated by malicious nodes. The knowledge base is located at the base station and is mainly used to store malicious events generated by nodes in the network and to prevent the generation of malicious events by acknowledging cluster heads. The literature [15] proposed a model for building a malicious behavior knowledge base based on a five-element model. The model first extracts and constructs entities using machine learning methods to obtain network security knowledge. The paper inferred rules partly and used the NER method to complete the knowledge extraction method in the cyber security domain. However, the above literature does not fully consider the graph structure characteristics of malicious behavior traffic in the network when constructing the knowledge base.

Graph neural networks are gradually applied to DDoS detection due to their ability to learn the spatial structure properties of sample nodes. The literature [16] proposed a graph neural network framework called GLASS to detect and identify DDoS attacks in an SDN environment. Meanwhile, the authors analyzed the impact of DDoS attacks on throughput, transmission delay, and other network performance, but the authors did not complete a traceability attack on DDoS. So the literature [17] used a graph neural network model to effectively extract the temporal and spatial features of the network state and find the path of a DDoS attack. The literature [18] proposed a detection framework called FAPDD. This framework includes three stages in detecting DDoS attacks: building a network graph model, calculating network graph scatter, and dynamic threshold detection. However, none of these literature constructs a suitable topology for the characteristics of DDoS attacks, which leads to the inability of the model to make full use of the data.

Federated learning allows individual participants to protect their own private data premise and complete joint training of multiple participants, which effectively solves the problem of the insufficient amount and less variety of data. The literature [19] combined federated learning with blockchain technology, and their anomaly detection models are chained on a distributed ledger. This combined approach with blockchain allows the privacy and security of the data to be well maintained through federated learning. A DDoS attack detection model called FLDDoS was proposed in the literature [20]. The framework uses a hierarchical aggregation algorithm based on K-Means and a data resampling method based on SMOTEENN to address the problem of the extremely uneven distribution of detection datasets and the small percentage of attack samples. The literature [21] pointed out that since the dataset for federated learning is distributed among multiple clients, the data samples and labels belong to unknown cases, a situation that may significantly degrade the performance of federated learning. In addition to this, all the above-mentioned literature only detects a small number of kinds of DDoS attacks and cannot achieve the detection of multi-level and multi-species DDoS attacks.

3 Multi-Domain Malicious Behavior Knowledge Base Framework

We propose a multi-domain knowledge base framework for DDoS behavior detection, which is built based on graph neural networks and federated learning, while we divide the overall system into four modules. First, we introduce the overall framework of the system. Then, we detail the methods and steps for collecting malicious traffic of DDoS attacks in the malicious behavior acquisition module. To achieve local learning of malicious behavior knowledge, we introduce the construction algorithm of knowledge graph and the training process of graph neural network. To protect the privacy of participant data, we introduce the implementation principle and details of federated learning. Finally, we present the construction method of the malicious behavior knowledge base, which will unify the management of the knowledge generated in the system.

3.1 Framework Overview

To achieve more comprehensive detection of multiple types of DDoS attacks, we propose a multi-domain malicious behavior knowledge base construction and application scheme. As shown in Fig. 1, the overall framework is divided into a malicious behavior data collection module, a graph neural network module, a federated learning module, and a malicious behavior knowledge base module. Our system is distributed and deployed at the entrance gateways of four network domains. When malicious behaviors of DDoS attacks occur in this domain, the malicious behavior collection module collects malicious traffic of DDoS attacks occurring in this domain and provides source data for the graph neural network module and the federated learning module after data preprocessing and malicious behavior dataset design. After that, we construct the malicious behavior category graph and the malicious behavior structure graph respectively, and the two graphs provide the feature information and structure information of nodes for the graph neural model respectively. At the same time, we will deposit the graph into the malicious behavior repository. After the local learning is completed, we implement the knowledge-sharing work of malicious behaviors based on the federal learning module. We extract the fully connected layer of the graph network model and construct a forward neural network that can be federated for learning to implement multi-domain federation training. Eventually, we deposit the trained local graph neural network model and the global federation model into the detection model repository. The malicious behavior knowledge base contains the knowledge graph and the trained models and implements a management mechanism for malicious behavior knowledge. The malicious behavior knowledge base will model DDoS attacks occurring in the network domain based on the knowledge graph and detection models, and then complete the DDoS behavior detection task.

3.2 Malicious Behavior Collection Module

The malicious behavior collection module implements the malicious behavior knowledge collection mechanism. The main function of the malicious behavior collection module is to collect the malicious traffic of DDoS attacks occurring in the current network domain and to design a suitable malicious behavior dataset for the distributed architecture of the system. As shown in Fig. 2, it contains 3 steps malicious behavior acquisition, data pre-processing and malicious behavior dataset design.

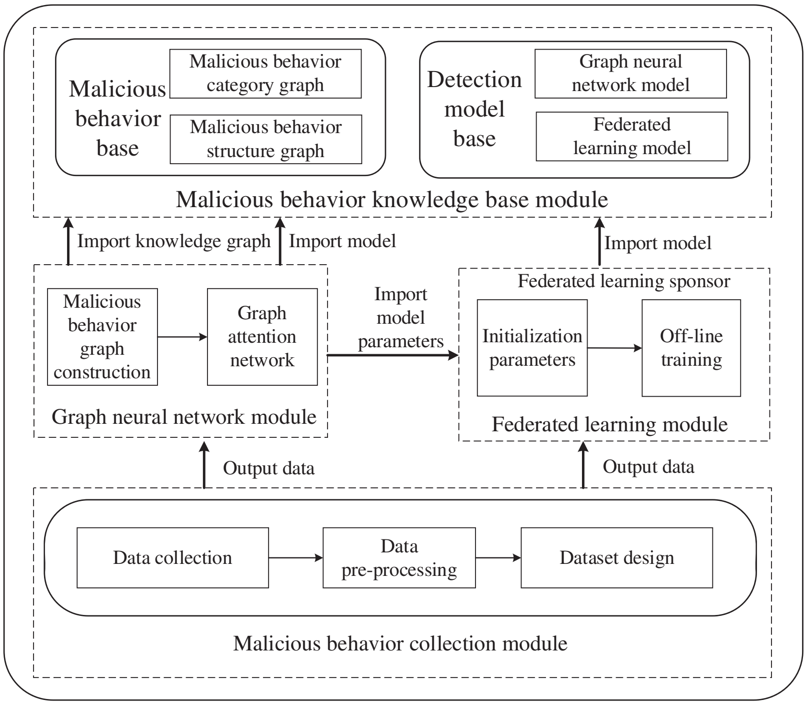
Malicious behavior base
Malicious behavior category graph
Malicious behavior structure graph
Detection model base
Graph neural network model
Federated learning model
Malicious behavior knowledge base module
Import knowledge graph
Import model
Import model
Malicious behavior graph construction
Graph attention network
Graph neural network module
Import model parameters
Federated learning sponsor
Initialization parameters
Off-line training
Federated learning module
Output data
Output data
Data collection
Data pre-processing
Dataset design
Malicious behavior collection module

Figure 1: Multi-domain malicious behavior knowledge base framework

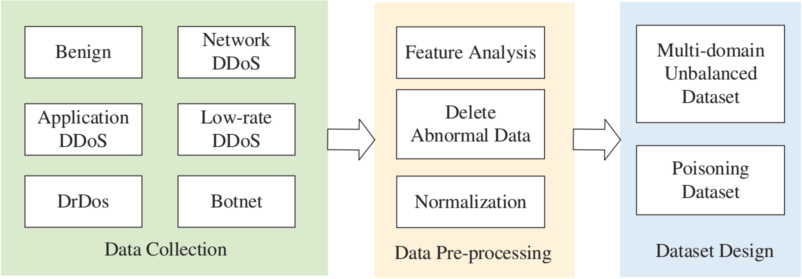

Figure 2: Malicious behavior collection process

3.2.1 Malicious Behavior Collection

Since all the current DDoS attack datasets exist only for some kinds of attacks, to achieve the detection of more kinds of DDoS attacks, we collected 5G normal traffic and 5 mainstream types of DDoS attacks. Among them, the 5 mainstream types of DDoS attacks contain 22 small types of DDoS attacks. We collected 22 types of DDoS attacks, including ACK, UDP, SYN, SlowBody, Shrew, SlowHeaders, SlowRead, Ares, BYOB, Miral, Zeus, IRC-Botnet, TFTP, Memcached, DRDoS_SSDP, DRDoS_NTP, Chargen, DRDoS_SNMP, CC, HTTP-Get, HTTP-Flood and HTTP-Post.

We use tools and scripts to simulate attacks and normal flow [22], and we use TCPCPUDUMP [23] collect experimental flow at the network entrance of each domain for a certain period of time to get the original data flow. Then, we use the CICFlowMeter [24] to extract the features of the traffic. CICFlowMeter can read the packet information in the pcap file, extract the relevant flow feature information, and finally output 84 dimensions of feature information The number of raw datasets is shown in Table 1.

Table 1: Statistics of the original dataset

DDoS label	Included subtypes of attacks	Quantity
Network layer DDoS	ACK, UDP, SYN	140051
Application layer DDoS	CC, HTTP-Get, HTTP-Flood, HTTP-Flood	425402
LDDoS	SlowBody, Shrew, SlowHeaders, SlowRead	324299
DRDoS	TFTP, Memcached, DRDoS_SSDP, DRDoS_NTP, Chargen, DRDoS_SNMP	151198
Botnet DDoS	Ares, BYOB, Miral, Zeus, IRC-Botnet	715240

As shown in Table 2, the original dataset contains 84 features. These 84 features include six aspects of features such as stream identification features, packet header Features, flag bit features, time features, Stream Attribute Features and payload features and payload features. Stream identification features represent the network attribute information of a stream. Packet header features represent the statistical features of the packet header. Flag bit features represents the statistical information of the flag field in the communication message. Time features represent the time interval information of packets in a stream. Stream attribute features represent the statistical features of packets in forward and reverse streams. Packet payload features information about the number of bytes occupied by the packet payload.

Table 2: Dataset features explanation

Feature type	Feature name
Stream identification features	Flow ID, Src IP, Dst IP, Src Port, Dst Port, Protocol, Timestamp, Label
Packet header features	FWD Init Win Bytes, FWD Win Bytes Mean\Min\Max\Std, Bwd Init Win Bytes, Bwd Win Bytes Mean\Min\Max\Std, Fwd Header Length Mean\Min\Max\Std\Sum, Bwd Header Length Mean\Min\Max\Std,
Flag bit features	FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, CWR Flag Count, ECE Flag Count, URG Flag Count, Fwd PSH Flag, Fwd URG Flag, Bwd PSH Flag, Bwd URG Flag
Time features	Flow duration, Flow IAT Min\Max\Mean\Std, Fwd IAT Min\Max\Mean\Std\Total, Bwd IAT Min\Max\Mean\Std\Total, Active Min\Max\Mean\Std, Idle Min\Max\Mean\Std
Stream attribute features	Total Fwd Packet, Total Bwd Packet, Flow Bytes/s, Flow Packets/s, FWD Packets/s, Bwd Packets/s, Down/Up Ratio, Fwd Act Data Pkts, Bwd Act Data Pkts, Act Packet Length Mean, Act Fwd Packet Length Mean, Act Bwd Packet Length Mean, Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes, Fwd Bytes/Bulk Avg, Fwd Bulk Rate Avg, Fwd Packet/Bulk Avg, Bwd Bytes/Bulk Avg, Bwd Bulk Rate Avg, Bwd Packet/Bulk Avg

(Continued)

Table 2 (continued)

Feature type	Feature name
Packet payload features	Packet Length Min\Max\Mean\Std, Packet Length Variance, Fwd Packet Length Min\Max\Mean\Std, Total Length of Fwd Packet, Bwd Packet Length Min\Max\Mean\Std, Total Length of Bwd Packet

3.2.2 Malicious Behavior Pre-Processing

The malicious behavior collection part will output data with data anomalies, invalid features, and other behaviors, so we need to process the collected malicious behavior.

Among the 84-dimensional flow features generated by the CICFlowMeter tool, 8 features, including flow ID, source IP address, destination IP address, source port number, destination port number, protocol, timestamp, and label, are not related to the popular form. This information is used to identify the attack traffic of a particular entry and cannot be used as a classification feature. If these features are added, it will reduce the generalization ability of the model, so we remove these features.

In addition to the invalid features, we also need to complete three tasks: anomaly data clarity, feature coding and data normalization. For the missing values in the collected data, if we set them to specific values or zero, it will affect the original likeness of the traffic features. Considering that we have collected sufficient normal traffic and malicious behavior traffic, we choose to remove the traffic with missing values. Then, we use min-max standardization to transform the values of the numerical features to be in the range of 0–1 [2]. The dataset that completes the above steps will be input to the dataset design module to provide a data source for the construction of the malicious behavior knowledge base.

3.2.3 Malicious Behavior Dataset Design

Graph neural networks are more efficient for graph-structured data learning than traditional models. And the size of the adjacency matrix for mapping node relationships grows with the amount of data in a $O\left(n^2\right)$ relationship. Considering the training cost issue and verifying the effectiveness of the local learning mechanism, we designed a small batch data set to verify the DDoS behavior detection capability of the model.

Since the system is distributed and deployed in multiple network domains, we need to design malicious behavior datasets suitable for each network domain. We divide the five malicious communication behaviors into four network domains, where each domain has data related to normal traffic and network layer DDoS attacks. At the same time, each domain has data for one malicious behavior alone. The multi-domain dataset we designed is shown in Table 3.

3.3 Graph Neural Network Module

The graph neural network module is used to implement a local learning mechanism for malicious behavior knowledge. As shown in Fig. 3, the graph neural network module includes two steps constructing a knowledge graph and graph neural network training. We construct the knowledge graph containing the relationship of malicious behavior nodes based on the quaternion <source IP address, destination IP address, source port number, destination port number> and timestamp information of DDoS attack flow. We also use the graph neural network model to realize the learning work of malicious behavior knowledge in this domain. At the same time, we will store the constructed knowledge graph in the malicious behavior base and store the trained local graph neural network model in the detection model base.

Table 3: Multi-domain malicious behavior data set

Participant	Category	Quantity	Participant	Category	Quantity
Party ID:10000 IP:192.168.98.79	Benign	3000	Party ID:9999 IP:192.168.98.80	Benign	3000
	Network DDoS	3000		Network DDoS	3000
	Application DDoS	3000		Application DDoS	333
	DRDoS	333		DRDoS	3000
	LDDoS	333		LDDoS	333
	Botnet	334		Botnet	334
Party ID:9998 IP:192.168.98.81	Benign	3000	Party ID:9997 IP:192.168.98.82	Benign	3000
	Network DDoS	3000		Network DDoS	3000
	Application DDoS	333		Application DDoS	333
	DRDoS	333		DRDoS	333
	LDDoS	3000		LDDoS	334
	Botnet	334		Botnet	3000

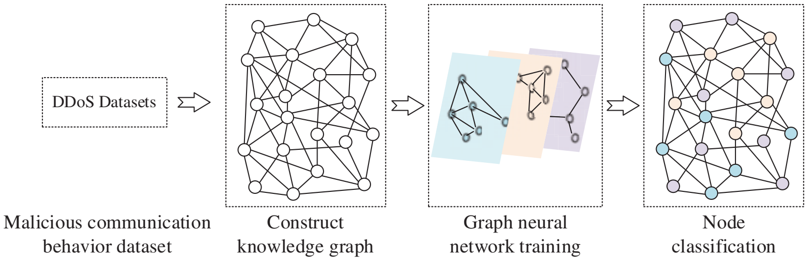
DDoS Datasets
Malicious communication behavior dataset
Construct knowledge graph
Graph neural network training
Node classification

Figure 3: Graph neural network module process

3.3.1 Malicious Behavior Graph Construction

There are many types of DDoS attacks. The attackers also have a variety of objectives, such as exhausting server resources or stealing resource information. However, regardless of the type of DDoS attack, the process always generates attack traffic, which is generating traffic characteristics. To describe the characteristic information and structural information of malicious communication behavior, we carve each malicious communication behavior as a malicious communication behavior node.

Considering that the graph neural network requires the input of the feature matrix and adjacency matrix of malicious behavior nodes, we construct the malicious behavior category graph and the malicious behavior structure graph respectively. The malicious behavior category graph contains specific malicious communication behavior nodes, which point to the traffic type node they belong to. Each entity in the malicious behavior category graph contains 84 dimensions of traffic features to provide feature information for DDoS behavior detection. The malicious behavior structure graph is a knowledge graph after the coarse-grained division of nodes, containing the connection relationship between normal traffic and five types of malicious communication behaviors, representing the adjacency matrix between malicious behavior nodes.

$$E_{i,j} = \begin{cases} 1, \text{if Source}_{ip,port} = \text{Dst}_{ip,port} \text{and timestamp } < 10 \\ 0, \text{else} \end{cases} \tag{1}$$

$E = \left\{E_{i,j}\right\}$ represents the adjacency matrix between malicious behavior nodes, we use it to represent the connection relationship between malicious behavior nodes, we first extract the <source IP address, destination IP address, source port number, destination port number> quaternions and timestamp information of multiple malicious communication behaviors, if two malicious behavior nodes have the same quaternion information and the time difference between two malicious communication behaviors occurring is less than 10 s, we consider that these two malicious behavior nodes have connection relationship, otherwise we consider that they are not connected.

3.3.2 Graph Attention Network Model

For effective learning of the knowledge base of malicious behaviors in this domain, we introduced the graph attention network model (GAT) [25]. The graph attention network adaptively assigns weights to each neighbor node through an attention mechanism and improves the expressiveness of the model by aggregating the features of the neighbors during the training process. In this paper, we use graph attention networks to implement local learning of malicious behavior knowledge. The graph attention network will learn the feature information and structural information of malicious behavior nodes stored in the malicious behavior knowledge base, and then acquire the ability to detect the collected malicious behaviors in this domain.

The feature matrix of malicious behavior nodes is denoted as $h = \{h_1, h_2, \ldots, h_n\}, h_i \in \mathbb{R}^F$, where N denotes the global number of malicious traffic and F denotes the feature dimension of each malicious traffic. If $E_{i,j} = 1$, then we calculate the value of $\alpha_{i,j}$. $\alpha_{i,j}$ is the attention interrelationships, which represents the magnitude of influence between malicious traffic node i and malicious traffic node j. The feature vector of malicious behavior nodes h may be negative, and the Leaky ReLU function will output non-zero values on negative inputs, which has better resistance to saturation, so GAT chooses the Leaky ReLU function to calculate the attention coefficient.

$$\alpha_{i,j} = softmax\left(e_{i,j}\right) = \frac{\exp\left(e_{i,j}\right)}{\sum_{K \in N_i}\left(e_{i,k}\right)} = \frac{\exp\left(\text{Leaky ReLU}\left(a\left(Wh_i || Wh_j\right)\right)\right)}{\sum_{K \in N_i}\left(\text{Leaky ReLU}\left(a\left(Wh_i || Wh_k\right)\right)\right)} \tag{2}$$

$W \in \mathbb{R}^{F' \times F}$ is a shared linear change matrix that can be learned. The matrix will be applied to each malicious traffic node, transforming the original feature space into a higher-level feature space to obtain better node representation.

After obtaining the number of inter-correlations of all nodes, we can calculate the representation of node features in the high-level dimensional space by node feature aggregation and forward propagation.

$$h_i^{l+1} = \sigma\left(\sum_{j \in N_i} \alpha_{i,j} W h_j^l\right) \tag{3}$$

In the process of graph neural network training, the malicious behavior category graph provides feature information about each node, and the malicious behavior structure graph provides structural information about the relationship between nodes. Compared with traditional deep learning methods, graph attention networks can learn both feature information and structural relationships of malicious behavior nodes, transforming the DDoS behavior detection task into a node classification task.

3.4 Federated Learning Module

The Federated Learning Module is used to implement a knowledge-sharing mechanism among multiple malicious behavior knowledge bases. Transferring data among multiple malicious behavior knowledge bases may cause cyber security issues such as privacy leakage. We transform the graph neural network model obtained from local learning. We extract the fully connected layers of the graph neural network and the corresponding parameters, and then construct a forward propagation neural network that can perform the classification task. Then, we let the model be federated between multiple domains for training, allowing the local model to gain the ability to detect malicious behavior in other network domains. For example, domain 1 contains datasets of low to medium-rate DDoS attacks, and domain 2 does not contain datasets of low-rate DDoS attacks, so its ability to detect low-rate DDoS attacks cannot be obtained in the local learning phase. But through federal learning, domain 2 can learn the low-rate DDoS attack data contained in domain 1, and then obtain the ability to detect this kind of DDoS attack.

We extract the forward-propagating fully connected layer of the graph neural network model, construct it into a deep network model. The specific algorithm is described as shown in Algorithm 1, which is implemented by 4 participants and 1 federated server. Our model uses 2 fully connected layers to complete the extraction of features and uses a softmax function in the output layer to complete the DDoS behavior detection task.

Algorithm 1: Off-line training algorithm

```
Input: K participants, Epoch E, learning rate $\eta$
/*Central server*/
for i=0; i < E; ++i
    Receive the gradient of each participant $g_k$
    Calculate the polymerization gradient $\overline{g} = \sum_{k=1}^{K} \frac{n_k}{n} g_k$
    Send aggregation gradient to each participant
endfor
/*Client update*/
Import the trained model parameters
for t=0; t < E; ++i
    for j=0; k < K; ++j
        Each participant calculates the local gradient $g_{k,t} = \nabla F_{k,t}(w_t)$
        Send $g_{k,j}$ to the central server
        Get the Polymerization gradient from the server $\overline{g}$
    update local NN model $w_{t+1} \leftarrow w_t - \eta\overline{g}$
    endfor
endfor
```

The local model of each participant encrypts the gradient using homomorphic encryption [26] after the gradient is computed, and the aggregation server uses secure aggregation techniques to achieve privacy protection of the model parameters after receiving the encrypted gradient [27]. After that, the aggregation server distributes the encrypted gradient obtained by the aggregation to each participant. Each participant updates the local model after the decryption step.

3.5 Malicious Behavior Knowledge Base Module

As shown in Fig. 4, The workflow of the malicious behavior knowledge base is divided into five steps. Step 1: The malicious behavior collection module collects simulated attack traffic within the network domain and Internet-initiated attack traffic. The simulated attacks within the network domain are labeled data. Step 2: The labeled data is exported to the malicious behavior base to build the knowledge graph. Step 3: The malicious behavior base provides feature information and structure information of malicious behavior nodes to the detection model base to train the model. Step 4: Outputs unlabeled data to the detection model base for detection. Step 5: Feeds the detection results to the malicious behavior repository to increase the knowledge capacity of the malicious behavior knowledge base.

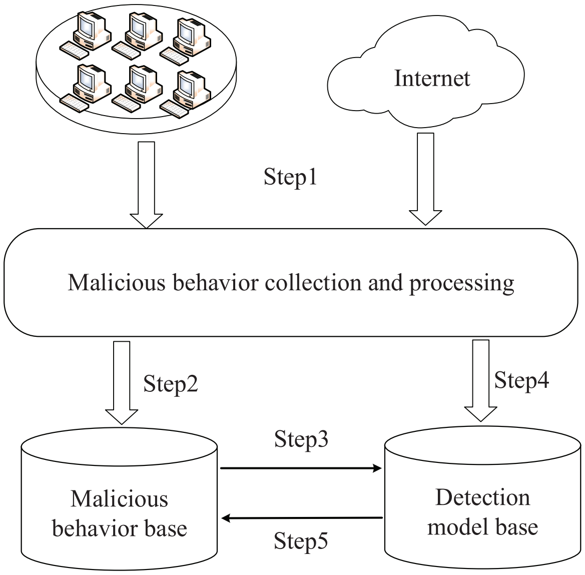

Figure 4: Working process of malicious behavior knowledge base

When a DDoS attack occurs in the network, the detection model identifies it, and at the same time adds the detection results to the malicious behavior base. This process continuously expands the capacity of the knowledge base, thus realizing an effective knowledge management mechanism.

3.5.1 Malicious Behavior Base

The malicious behavior repository organizes the data in the form of a knowledge graph, and the mathematical representation of the knowledge graph is shown in formula (3)

$$\text{Knowledge Graph} = \langle E, R, P \rangle \tag{4}$$

E denotes the set of entities in the knowledge graph, which is the mathematical representation of the objects stored in the knowledge graph. *R* denotes the connection between different entities, which is the mathematical representation of the objects associated with the knowledge graph. *P* Represents the set of attributes in the knowledge graph, which are the characteristics possessed by entities and relations.

The malicious behavior category graph includes six types of entity nodes, which are communication behavior, malicious communication behavior, normal communication, DDoS attack traffic, normal traffic, and entity behavior nodes. The entity behavior node corresponds to one specific communication behavior, and all the remaining nodes are virtual nodes to indicate the type of communication behavior. The entity behavior node points to the DDoS attack node or normal traffic node, and the DDoS attack type node points to the corresponding malicious communication behavior. The malicious behavior category graph shows the relationship before and after coarse and fine granularity division of DDoS attack types, while the graph manages the feature information of multi-category and multi-level DDoS attack traffic, which provides source data for analyzing the characteristics of DDoS attacks and training of detection models. The entities, attributes and relationships of the malicious behavior category graph constitute a triad as shown in Table 4. The relationship indicates which type of node this type is related to.

Table 4: Triads of malicious behavior category graph

Entity	Attribute	Relationships
Communication behavior	Virtual node	Malicious/normal communication
Malicious communication	Virtual node	Communication behavior
Normal communication	Virtual node	Communication behavior
DDoS attack traffic	Virtual node	Malicious communication
Normal traffic	Virtual node	Normal communication
Entity behavior node	Traffic feature	Related traffic type

The malicious behavior structure graph includes six types of entity nodes, which are normal traffic, network layer DDoS attack, application layer DDoS attack, LDDoS attack, DrDoS and botnet nodes. In the malicious behavior structure graph, the same type of malicious communication behavior nodes is connected after satisfying the spatial and temporal relationships. The graph can model and portray the relationship between different malicious communication behaviors, while providing structural information of malicious behavior nodes for the detection model. The entities, attributes and relationships of the malicious behavior structural graph diagram constitute a triad as shown in Table 5.

Table 5: Triads of malicious behavior structure graph

Entity	Attribute	Relationship
Normal traffic	Traffic feature	Related other normal traffic node
Network DDoS	Traffic feature	Related other network DDoS node
Application DDoS	Traffic feature	Related other application DDoS node
LDDoS	Traffic feature	Related other LDDoS node
DrDoS	Traffic feature	Related other DrDoS node
Botnet DDoS	Traffic feature	Related to other botnet node

3.5.2 *Detection Model Base*

The detection model base stores the network structure of the model and the parameters obtained after training, and it is represented by the formula as shown in formula (4).

$$\text{Detection Model} = \langle S, W \rangle \tag{5}$$

S denotes the network structure of the local model or the global model, and W denotes the final parameters obtained from model training and learning. We store the models as h5 files, which can preserve the complete network structure and parameter information of the models. The detection model repository manages the graph neural network model and the federation model. The detection model allows the malicious behavior knowledge base to gain DDoS behavior detection capability. We deploy it distributed at the ingress gateway of each network domain to identify all traffic uniformly and provide guidance for detecting and mitigating malicious communication behaviors occurring in the network.

4 Experimental Results

4.1 Experimental Environment

Our experimental topology environment is shown in Fig. 5. We build the experimental environment based on the virtual platform of VMware vSphere. There are 4 hosts for the federated learning participants, while Participant 1 will act as the aggregator. Each host has an OS version of Ubuntu 18.04, 8 virtual cores, and 16 GB of memory. our graph neural network model is based on TensorFlow [28], and our federation learning experiments are based on the FATE [29].

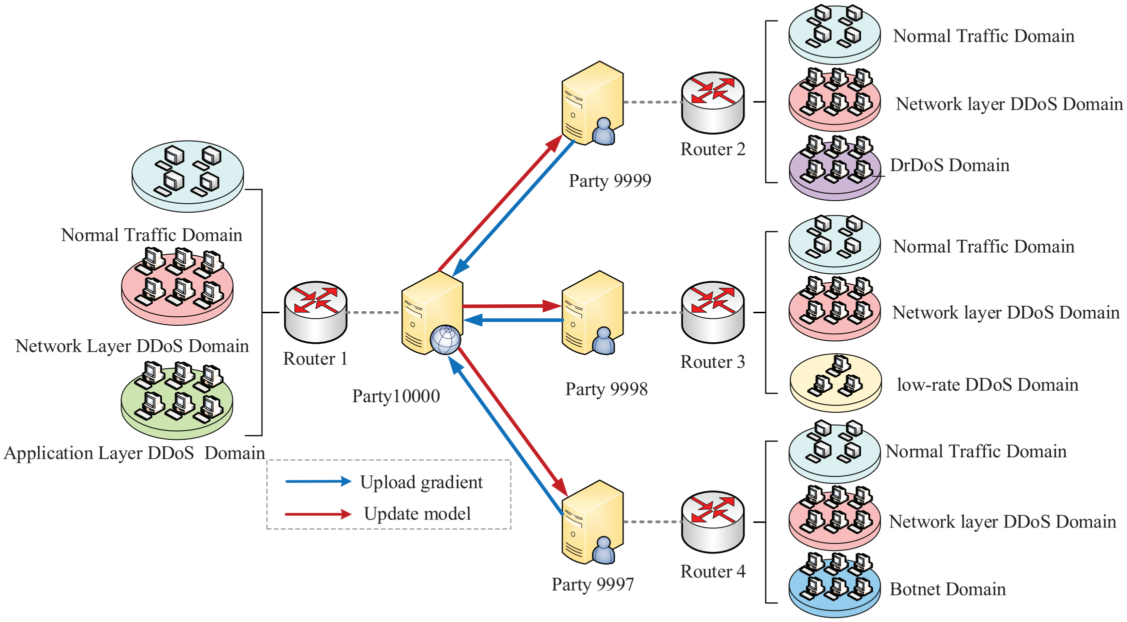

Figure 5: Network topology of the experimental environment

Each participant only collects the relevant data of its connected DDoS attack domain. For example, the data of Participant 1 is collected through router 1. It only has the relevant data of network layer DDoS attacks, application layer DDoS attacks and normal traffic. Therefore, he will not have the relevant data on other types of DDoS attacks, so the participant's ability to detect other types of DDoS in the local learning stage will be limited.

4.2 Knowledge Graph Construction Results

4.2.1 Initial State of Malicious Behavior Base

Our system captures both labeled traffic and unlabeled traffic. For labeled traffic, we can directly determine the connection relationship with other nodes based on the type of malicious behavior nodes. However, for unlabeled traffic, we can only discriminate it by the detection model and then incorporate it into the knowledge graph. Fig. 6 represents the initial state of the malicious behavior base. We import both labeled traffic and unlabeled traffic into the knowledge base. Normal communication behaviors and the five types of malicious communication behaviors are labeled into corresponding colors and clustered together, while the unlabeled traffic is scattered around the various known labeled traffic.

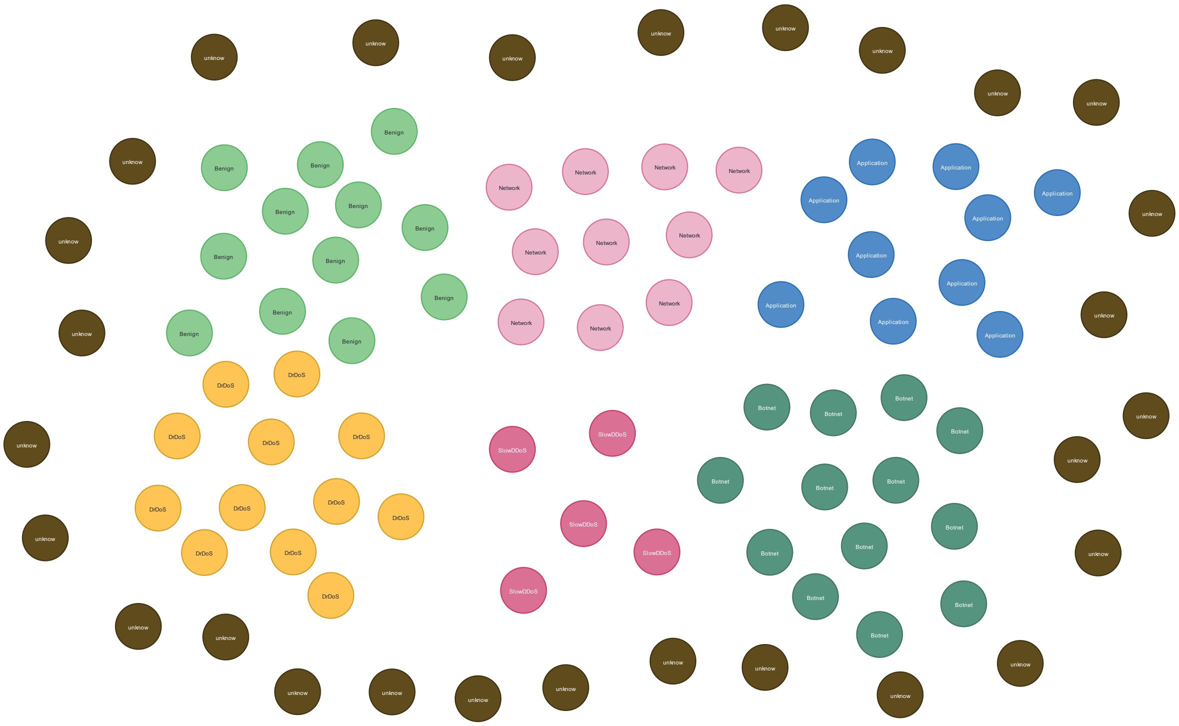

Figure 6: Initial state of the malicious behavior base

4.2.2 Build Malicious Behavior Knowledge Graph

For labeled traffic, we can construct a malicious behavior category graph and a malicious behavior structure graph based on their types. Among them, the malicious behavior category graph describes the category and characteristic information of malicious behavior nodes. To show the basis of our classification of 22 DDoS attacks into 5 malicious communication behaviors, we make each entity behavior node point to the DDoS attack type node it belongs to, and let the DDoS attack traffic type node point to the malicious communication behavior category node it is classified into. Fig. 7 represents our preliminary malicious behavior category graph, brown nodes are unknown label nodes, which need to be added to the knowledge graph after model detection.

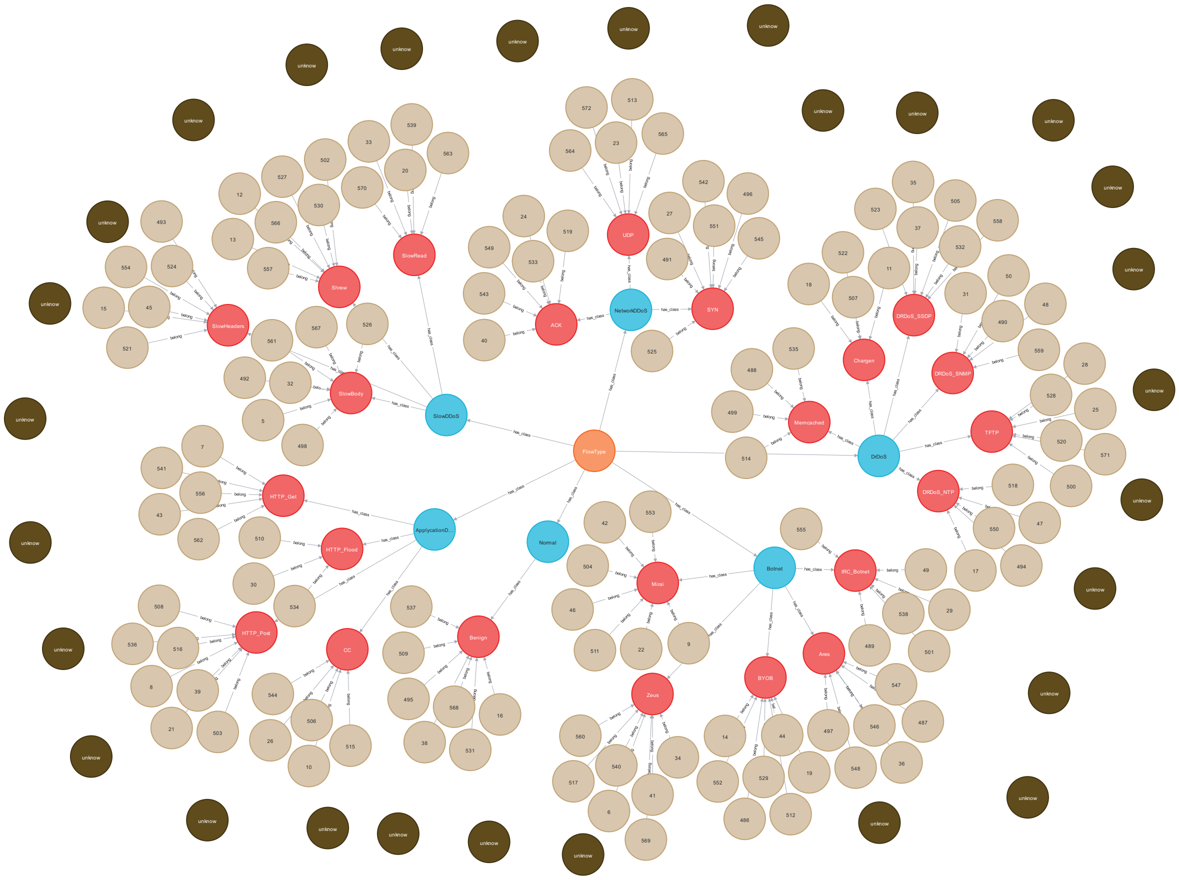

Figure 7: Initially constructed malicious behavior category graph

The malicious behavior structure graph depicts the structural information of the malicious behavior nodes. For the entity behavior nodes of known category, we divide them into different clusters according to the method described in Subsection 3.2.3, and connect the malicious behavior nodes with relationships within the clusters. Fig. 8 represents our initial construction of the malicious behavior structure graph, where the brown nodes denote the unknown labeled nodes. The unknown label nodes can only be connected to the corresponding clusters after being detected by the model.

4.2.3 Update of Knowledge Graph

The graph neural network performs semi-supervised learning for malicious behavior nodes that have constructed relationships, and it is unable to perform feature aggregation for nodes that do not construct the spatial structure. The federated learning model can identify nodes with unknown labels. We feed the results of the federation learning model detecting the obtained labels into the malicious behavior base and add the corresponding nodes to the malicious behavior category graph and malicious behavior structure graph. As shown in Figs. 9 and 10, after the model training and learning, the structure of the knowledge graph is established for all nodes in the malicious behavior base. The knowledge graph is updated with more feature information and structure information of malicious behavior nodes.

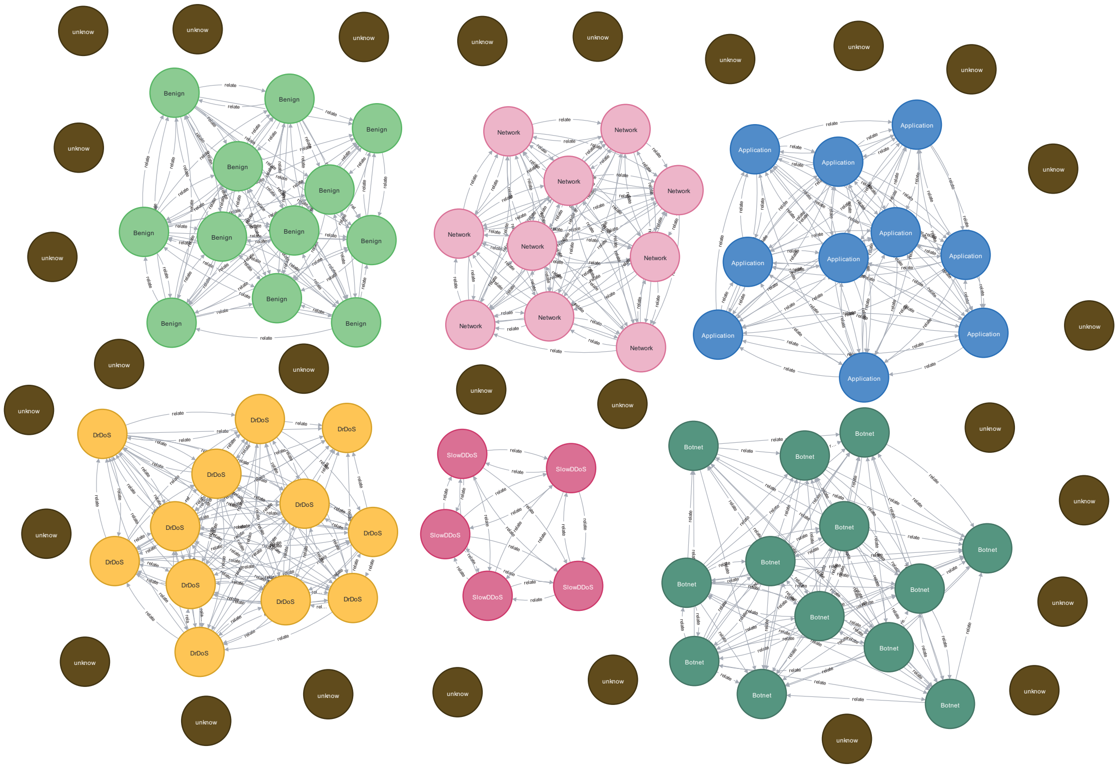

Figure 8: Initially constructed malicious behavior structure graph

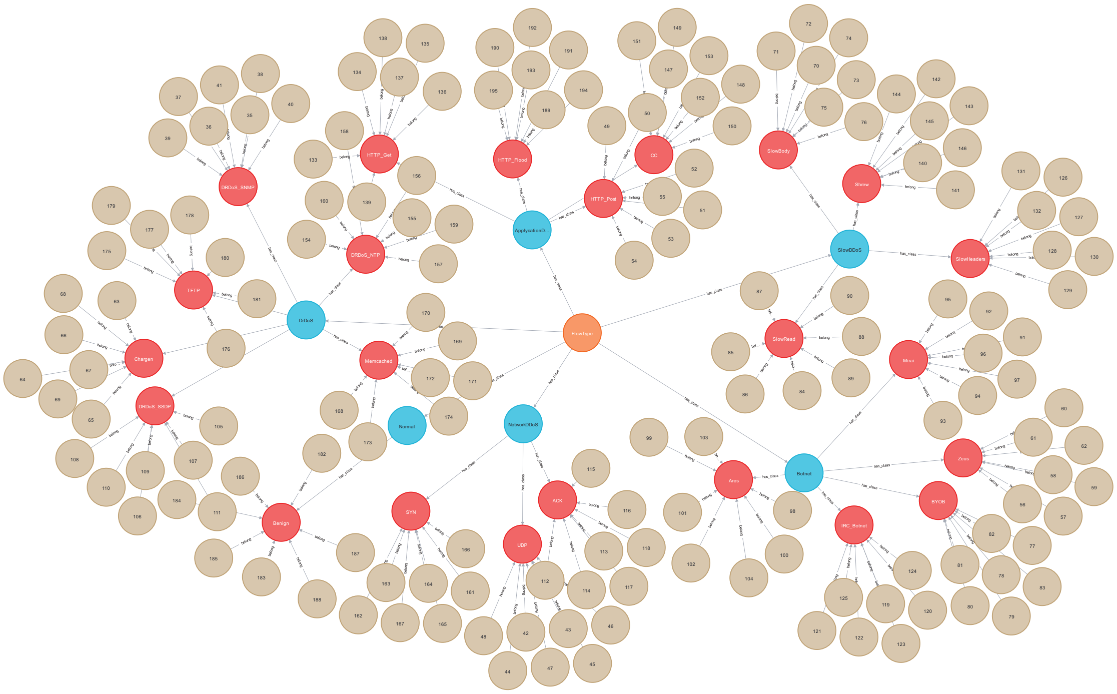

Figure 9: Malicious behavior category graph

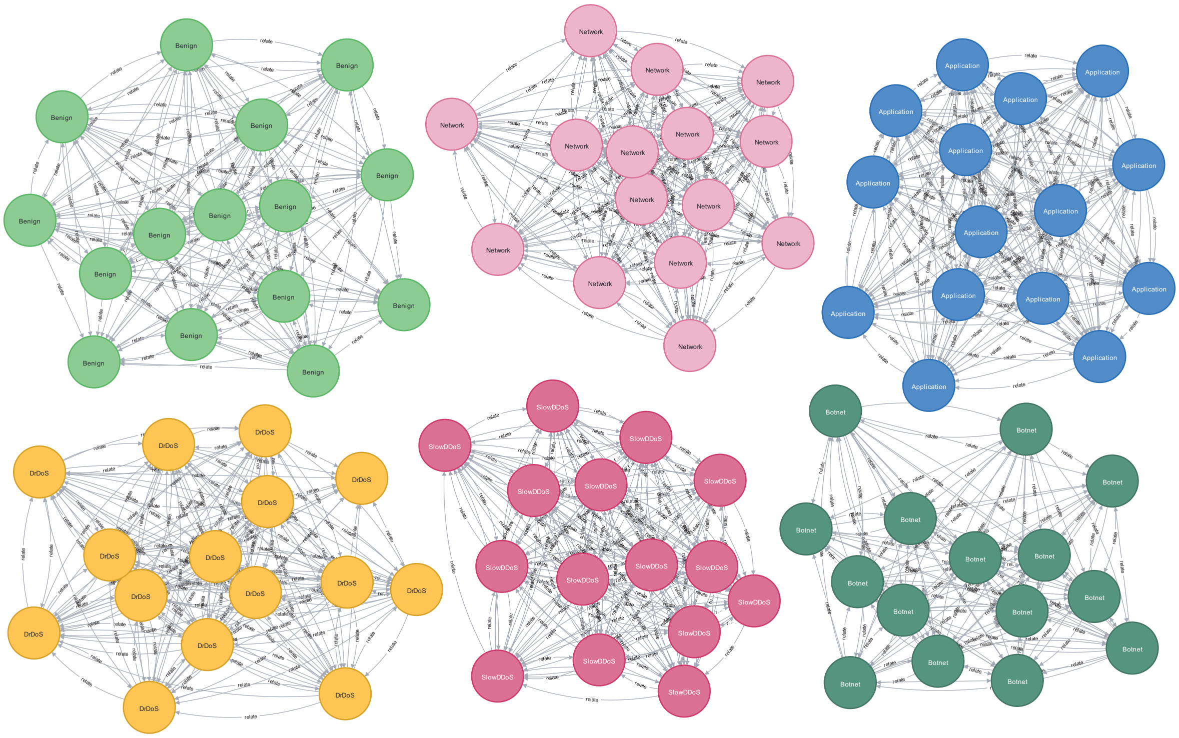

Figure 10: Malicious behavior structure graph

4.3 Model Detection Results

4.3.1 Analysis of Learning Structure Information

In the process of training, the graph attention network will aggregate the characteristics of malicious behavior nodes according to the adjacency matrix, so that the model can learn the structural information of malicious behavior nodes. We use the distance between classes to measure the learning of graph neural network for the structural characteristics of malicious behavior nodes.

$$dis = \frac{tr\left(B_k\right)}{k-1} = \frac{\sum_{q=1}^{k} n_q \left(c_q - c_E\right)\left(c_q - c_E\right)^T}{k-1} \tag{6}$$

In formula (6), k represents the total number of categories, n_q represents the number of nodes of category q, c_q represents the central coordinates of the category q, and c_E represents the central coordinates of all nodes. As shown in Fig. 11, during the training process, the inter-class distance of different clusters gradually increases, and the accuracy of model detection increases. This is because the greater the distance between classes, the easier the classifier can find the boundaries of different classes. Therefore, graph neural network can better complete the task of node classification and DDoS behavior detection according to the spatial structure of malicious behavior nodes.

4.3.2 Comparison of Local Model and Global Model

Each participant only collects the relevant data of its connected DDoS attack domain. For example, the data of Participant 1 is collected through router 1. It only has the relevant data of network layer DDoS attacks, application layer DDoS attacks and normal traffic. Therefore, he will not have the relevant data on other types of DDoS attacks, so the participant's ability to detect other types of DDoS in the local learning stage will be limited.

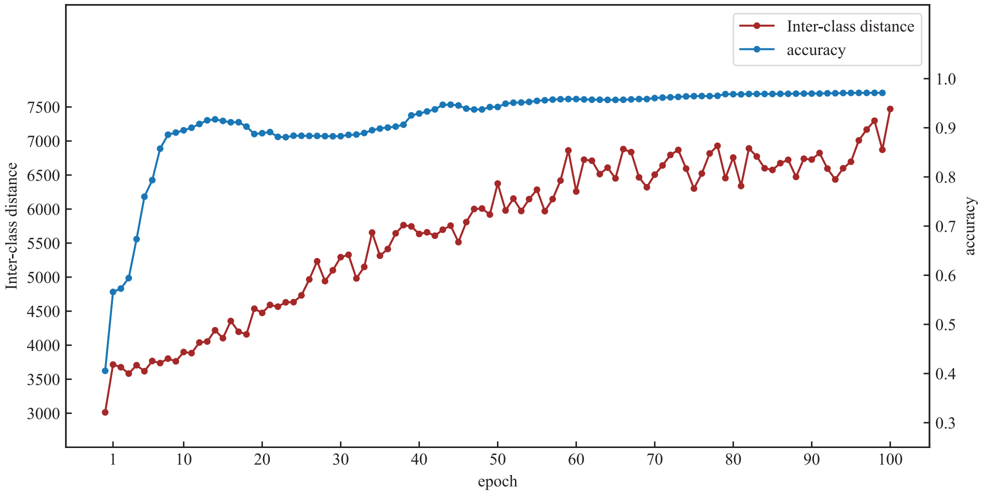

Figure 11: Change trend of detection accuracy and inter-class distance

Table 6 shows the results of the comparison between the local model and the federated model for the four participants. As can be seen, since each participant has data for both benign and network DDoS, each participant's local model is effective in detecting both types of traffic. In addition to this, each participant shows better results for the datasets they alone own. For example, Party 1000 for application layer DDoS attacks, Party 9999 for DrDoS, Party 9998 for LDDoS, and Party 9997. For Party 9999, the accuracy of detection is only 0.3212 due to the lack of data on application-layer DDoS attacks. after the joint training of multiple malicious behavior knowledge bases, the accuracy of application-layer DDoS attacks improves from 0.3212 to 0.985, and the recall improves from 0.7719 to 0.9240. This proves that the federated learning module effectively completes the knowledge-sharing mechanism among multiple malicious behavior knowledge bases.

4.3.3 Comparison of Multi-Domain Detection Algorithm

To verify the effectiveness of our framework in multi-domain DDoS behavior detection, we compare the scheme adopted in this paper with the HomoNN (Homogeneous Neural Network) [30] and HomoSecureBoost [31]. As shown in Fig. 12, we can see that the detection effect of the knowledge base framework in normal traffic, network layer DDoS attack, application layer DDoS attack and DrDoS attack is slightly higher than the other two schemes. Especially in the accuracy of Botnet and the recall rate of low-rate DDoS attacks, the method proposed in this paper has better performance. This is because the knowledge base framework has acquired the ability to detect malicious behavior in the local learning stage, thus helping the federated learning stage to obtain better results.

To demonstrate that local training accelerates the federated learning process, we analyzed the overall time consumption of the system. As shown in the red part of Fig. 13, the time consumption of our multiple hosts for federated learning is much lower than that of the HomeNN and HomeSecureBoost. The green and orange parts of the figure are the construction of the malicious behavior knowledge graph and the training time consumed by the graph neural network, respectively. These two parts are performed independently, and each participant could complete both parts at any point before the start of the training. As can be seen, our local training process accelerates the learning process of the federation training, which is because our scheme has already learned some initial weights in the local model, so the federation learning time has been greatly reduced.

Table 6: Comparison between the local model and global model

Model name	Category	Precision	Recall	Model name	Category	Precision	Recall
Local model party10000	Benign	1	0.9196	Local model party 9999	Benign	1	0.9257
	Network DDoS	0.9750	0.9613		Network DDoS	1	1
	Application DDoS	0.9360	0.9504		Application DDoS	0.3212	0.7719
	DrDoS	0.9310	0.8182		DrDoS	1	0.9967
	LDDoS	0.6061	0.9091		LDDoS	0.8	1
	Botnet	0.4545	0.7895		Botnet	0.9683	0.9242
Local model party 9998	Benign	0.9787	0.8747	Local model party 9997	Benign	1	0.9086
	Network DDoS	0.9857	0.9787		Network DDoS	0.9913	0.9879
	Application DDoS	0.6431	0.7120		Application DDoS	0.4156	0.3333
	DrDoS	0.8966	0.7879		DrDoS	0.9833	0.9219
	LDDoS	0.8990	0.9745		LDDoS	0.5231	0.9444
	Botnet	0.6667	0.9167		Botnet	0.9918	0.9618

Model name	Category	Precision	Recall
Federated learning model	Benign	0.9867	0.99
	Network DDoS	1	1
	Application DDoS	0.9850	0.9640
	LDDoS	0.9721	0.9838
	DrDoS	0.9881	1

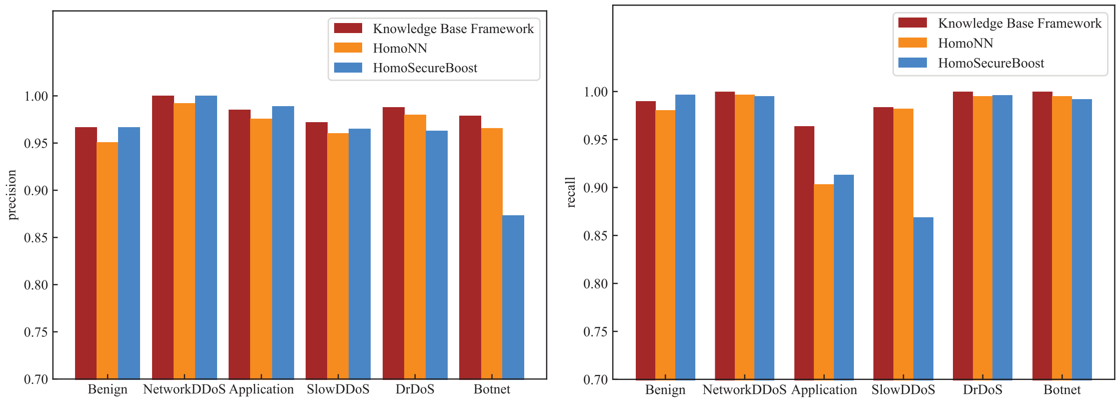

Figure 12: Comparison of different methods

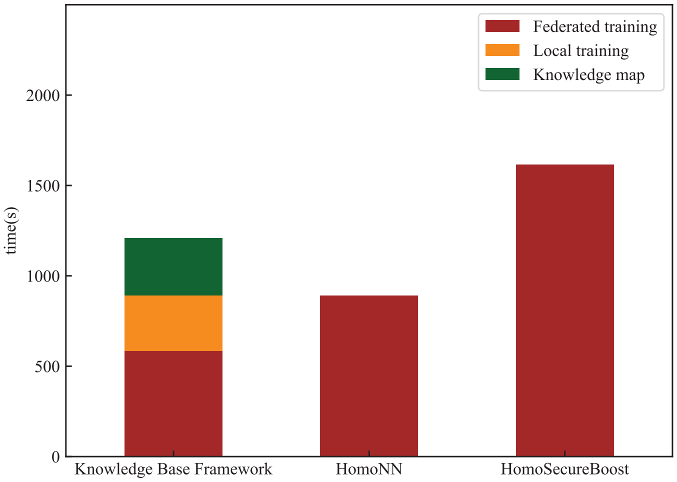

Figure 13: Time-consuming analysis

4.3.4 Poisoning Data Experiment

We designed the data poisoning dataset to verify the anti-interference capability of the system. We manually set the label of the original data as error label and increase it by 0.05. As the percentage of data contamination increases, the detection accuracy of the system keeps decreasing. As shown in Fig. 14, we show the variation of accuracy of the model for the four participants in our system, and we also show the best and worst performance of HomoNN and HomeSecureBoost among the four participants. We can see that the resistance of the Party9999 and Party9997 participant models is better than the HomoNN and HomoSecurebost algorithms, and the performance of the other two participants is higher than the poorer performance of the HomoNN and HomoSecurebost. In terms of average performance, our system performs better than the other two algorithms. In general, when the proportion of incorrect labels is less than 20%, the detection performance of each participant in our system can still reach above 0.8. When the proportion of incorrect labels is greater than 20%, the detection accuracy of the system for malicious communication behavior will be significantly reduced.

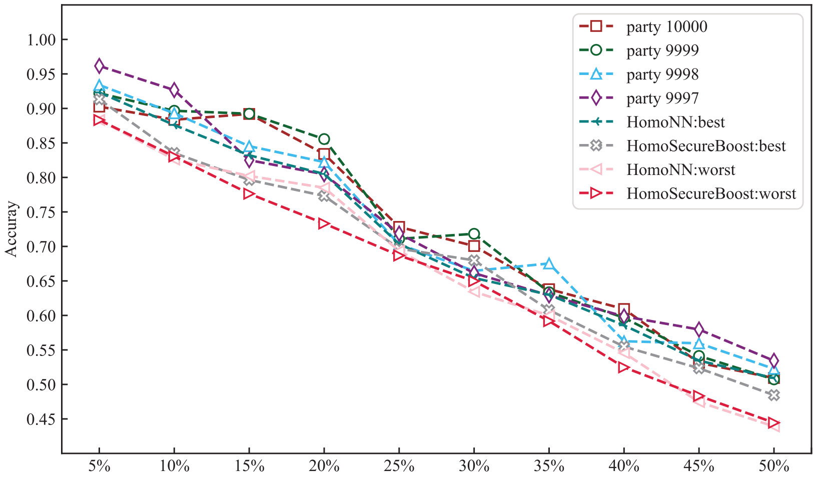

Figure 14: Influence of data poisoning on system detection accuracy

4.3.5 Online Detection Experiment

We deploy the federated model at the network entry point and perform real-time attacks to verify the detection performance of the model in real scenarios. It is worth noting that when performing online detection, we should ensure that the normalized scale of the model input data is uniform with the training data, otherwise it can have very bad consequences. As shown in Fig. 15, we demonstrate the detection performance for various attacks under different time windows. The time window refers to the time when the attack traffic is collected, and the detection performance can be best demonstrated when the time window is 120 s. Although the online detection performance is slightly worse than the offline detection, the detection performance of each attack can still reach above 0.95 in the 120 s time window.

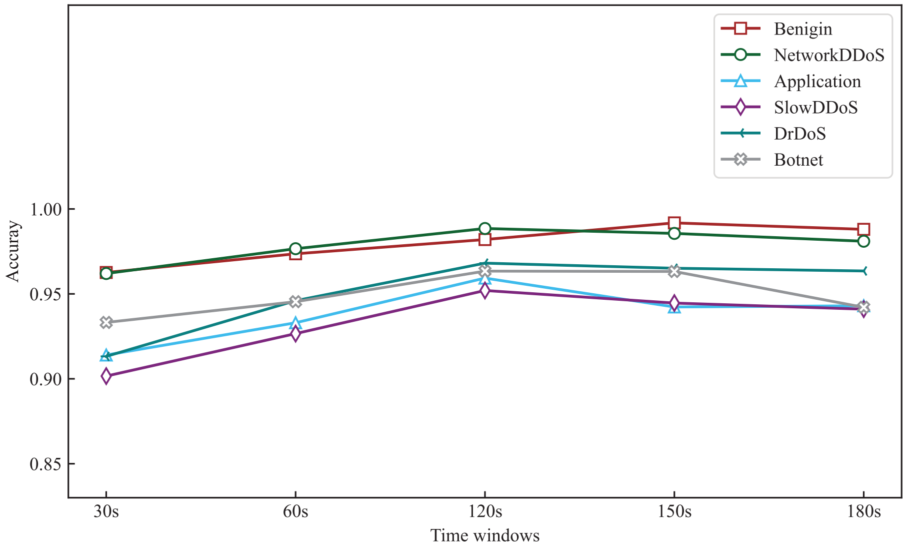

Figure 15: Online detection of experimental effects

5 Conclusion

In this paper, we implement a multi-domain malicious behavior knowledge framework for DDoS behavior detection. We collected 5G normal traffic and 5 mainstream types of DDoS attacks. To deeply mine the feature information and structural information of malicious communication behaviors, we construct two malicious behavior knowledge graphs. Meanwhile, we complete the local knowledge learning mechanism based on the graph attention network. Then, we implement a knowledge-sharing mechanism among multiple malicious behavior knowledge bases based on federation learning, and the process protects the data privacy of each participant. The experimental results show that we can achieve detection effectiveness above 0.95 for most types of attacks. Finally, we store the constructed knowledge graphs and the trained models in the malicious behavior knowledge base. The malicious behavior knowledge base guides the detection and mitigation of malicious communication behaviors occurring in the network in the future.

In the future, we will investigate how to mitigate the impact of data poisoning on the model. We will also consider optimizing the structure and parameters of the graph neural network model and the federated learning model to further improve the accuracy of detection.

Funding Statement: This paper is supported by the National Key R&D Program of China under Grant No. 2018YFA0701604.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Churcher, R. Ullah, J. Ahmad, S. U. Rehman, F. Masood *et al.,* "An experimental analysis of attack classification using machine learning in IoT networks," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 446, 2021.

[2] M. Li, Y. Qin and H. Zhou, "A DDoS detection method with feature set dimension reduction," in *Mobile Internet Security: 5th Int. Symp., MobiSec 2021*, Jeju Island, South Korea, pp. 365–378, 2022.

[3] A. Praseed and S. P. Thilagam, "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 661–685, 2022.

[4] A. J. Perez-Diaz, A. I. Valdovinos, R. K. K. Choo and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.

[5] R. R. Nuiaa, S. Manickam, H. A. Alsaeedi and S. E. Alomari, "A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks," *Int. J. Electr. Comput. Eng*, vol. 12, no. 2, pp. 1869–1880, 2022.

[6] A. T. Tuan, V. H. Long, H. L. Son, R. Kumar, I. Priyadarshini *et al.,* "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 283–294, 2020.

[7] K. Li, H. Zhou, Z. Tu and H. Zhang, "Blockchain empowered federated learning for distributed network security behaviour knowledge base in 6G," *Security and Communication Networks*, vol. 2022, pp. 1–11, 2022.

[8] K. Liu, F. Wang, Z. Ding, Z. Yu and Y. Zhou, "Recent progress of using knowledge graph for cybersecurity," *Electronics*, vol. 11, no. 15, pp. 2287, 2022.

[9] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommunication Systems*, vol. 73, no. 1, pp. 3–25, 2020.

[10] M. Mittal, K. Kumar and S. Behal, "Deep learning approaches for detecting DDoS attacks: A systematic review," *Soft Computing*, vol. 2022, pp. 1–37, 2022.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson and A. B. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2012.

[12] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li *et al.,* "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, no. 1, pp. 106775, 2021.

[13] X. Chen, S. Jia and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 216, no. 6, pp. 112984, 2020.

[14] A. Mehmood, A. Khanan, M. M. Umar, S. Abdullah, Z. A. K. Ariffin *et al.,* "Secure knowledge and cluster-based intrusion detection mechanism for smart wireless sensor networks," *IEEE Access*, vol. 6, pp. 5688–5694, 2017.

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

[16] K. Nagaraj, A. Starke and J. McNair, "Glass: A graph learning approach for software defined network based smart grid ddos security," in *ICC 2021-IEEE Int. Conf. on Communications*, Montreal, QC, Canada, pp. 1–6, 2021.

[17] Y. Cao, H. Jiang, Y. Deng, J. Wu, P. Zhou *et al.,* "Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 3855–3872, 2021.

[18] X. Liu, J. Rem, H. He, B. Zhang, C. Song *et al.,* "A fast all-packets-based DDoS attack detection approach based on network graph and graph kernel," *Journal of Network and Computer Applications*, vol. 185, no. 2, pp. 103079, 2021.

[19] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang *et al.,* "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 4177, no. 6, pp. 4177–4186, 2019.

[20] J. Zhang, P. Yu, L. Qi, S. Liu, H. Zhang *et al.,* "Flddos: DDos attack detection model based on federated learning," in *2021 IEEE 20th Int. Conf. on Trust, Security and Privacy in Computing and Communications*, Shenyang, China, pp. 635–642, 2021.

[21] Y. Zhao, M. Li, L. Lao and N. Suda, "Federated learning with non-iid data," *arXiv preprint*, arXiv:1806.00582, 2018.

[22] M. Li, H. Zhou and Y. Qin, "Two-stage intelligent model for detecting malicious DDoS behavior," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 2532, 2022.

[23] Tcpdump, 2019. [Online]. Available: https://www.Tcpdump.org/

[24] CICFlowMeter, 2022. [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter/

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio *et al.,* "Graph attention networks," *arXiv preprint*, arXiv:1710.10903, 2017.

[26] A. Acar, H. Aksu, S. A. Uluagac and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[27] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, B. H. McMahan *et al.,* "Practical secure aggregation for federated learning on user-held data," *arXiv preprint*, arXiv:1611.04482, 2016.

[28] TensorFlow, 2022. [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter/

[29] FATE, 2023. [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter/

[30] FATE Component Description, 2023. [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter/

[31] K. Cheng, "SecureBoost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 35, no. 6, pp. 97–98, 2021.