FIR-GNN: A Graph Neural Network using Flow Interaction Relationships for Intrusion Detection of Consumer Electronics in Smart Home Network

Mengyi Fu, Pan Wang, *Member, IEEE*, Shidong Liu, Xuejiao Chen, Xiaokang Zhou, *Member, IEEE*

***Abstract*—In the smart home scenario, the Consumer Internet of Things (CIoT) deeply integrates into daily life with various Consumer Electronics (CEs) like home cameras, smart speakers, smoke/fire detectors, VR/AR game boxes/handles, and future home medical terminals. However, CEs face multiple risks due to attack concealment and protocol differences. Against this backdrop, embedding Network Intrusion Detection System (NIDS) in the smart home gateway is proposed. Despite Machine Learning (ML) and Deep Learning (DL) enhancing network intrusion detection, challenges remain in sample collection, traffic feature expression, and gateway resource constraints. To address these, we propose FIR-GNN. It constructs a FIRG graph for traffic pattern capture, uses edge-wise graph attention in FIR-GNN for semi-supervised learning, and selects features by SHAP to cut resource consumption. Experiments show FIR-GNN improves classification performance by 3-5% on BoT-IoT and CICIDS2017 data, safeguarding smart home CEs.**

***Index Terms*—Network intrusion detection, graph neural networks, smart home gateways, graph attention networks, consumer electronics.**

I. Introduction

In the smart home scenario, the Consumer Internet of Things (CIoT) [1] has been deeply integrated into people's daily lives, covering numerous fields. Consumer Electronics (CEs) play a crucial role in it. For example, home cameras are used for home security monitoring, smart speakers enable convenient voice interaction, smoke/fire detectors ensure residential safety, VR/AR game boxes/handles bring immersive entertainment experiences and even future home medical terminal devices safeguard the health of family members [2], bringing people an unprecedentedly convenient experience. However, the complexity of the CIoT network in the smart home has exacerbated security risks and has become a key bottleneck restricting its further development.

There are numerous hidden risks among CEs. For CEs like home cameras, if they are subject to a man-in-the-middle attack, attackers can intercept, modify, and steal information during the data transmission process between them and servers or other devices, which may lead to the leakage of the family's private images. If smart speakers are hacked, attackers may remotely control them to play audio without reason, thus interfering with normal life. Once smoke/fire detectors are maliciously interfered with, they will fail to give timely alarms in case of a fire, which will bring great danger to families. In addition, a large number of heterogeneous CEs in smart homes, such as smart TVs, smart refrigerators, and smart lighting systems, are interconnected and communicate with each other, forming a huge and complex network topology. These CEs from different manufacturers adopt different communication protocols and technical standards, making it difficult for traditional security protection means to respond effectively.

In recent years, Machine Learning (ML) and Deep Learning (DL) techniques have been applied to the field of network security. Machine learning can extract latent patterns from traffic features for traffic classification. However, obtaining high-quality traffic features is both time-consuming and labor-intensive. Deep learning techniques have also been widely used due to their capabilities of automatic feature extraction and uncovering deep nonlinear features. Nevertheless, they face problems such as weak generalization and robustness, especially in the highly dynamic and heterogeneous smart home IoT environment when dealing with traffic data with limited representativeness. For example, most ML/DL-based methods have difficulties in handling traffic data in non-Euclidean space, which is common in smart home networks. Although some studies convert network traffic into graph-based representations to make better use of Graph Neural Network (GNN) methods, existing deep learning models still require a large number of labeled samples. In smart home networks, the scarcity of labeled data can lead to a decline in classification performance. Additionally, Graph Neural Networks consume a large amount of resources during the training and inference processes, making it difficult to meet the requirements of resource-constrained smart home gateways.

Against this background, embedding and deploying a Network Intrusion Detection System (NIDS) [3] on the smart home gateway has opened up new ideas for solving the CIoT security problems of the CEs in smart homes [4]. As the core hub of the home network, the smart home gateway is capable of monitoring and managing the data traffic of all connected CEs. By embedding and deploying the network intrusion

This work is supported by the Suzhou Science and Technology Planning Project under Grant No.SYG202311 and Suzhou Science and Technology Planning Project under Grant No.LHT202326.

M.Fu and P.Wang are with the School of Modern Posts, Nanjing University of Post&Telecommunications, Nanjing, China, (e-mail: 2023070802@njupt.edu.cn, wangpan@njupt.edu.cn), S.Liu is with the China Electric Power Research Institute, Nanjing, China, (e-mail: liushidong1@epri.sgcc.com.cn), X.Chen is with the School of Communications, Nanjing Vocational College of Information Technology, Nanjing, China (e-mail: chenxj@njcit.cn), X.Zhou is with the Faculty of Business Data Science, Kansai University, Osaka 565-0823, Japan, and also with the RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo 103-0027, Japan (e-mail: zhou@kansai-u.ac.jp).

Manuscript received April 19, 2021; revised August 16, 2021.

detection system, abnormal behaviors in the network can be monitored in real-time, potential attacks can be detected and blocked promptly, and the safe operation of various types of CEs in smart homes can be effectively ensured, safeguarding the privacy and safety of family members.

A. Motivation

Although Machine Learning and Deep Learning techniques have significantly improved the performance of NIDS, challenges remain, especially in the effective representation of network traffic features. These problems become even more prominent when they are closely associated with CEs.

Firstly, it is extremely difficult to obtain attack samples [5]. Due to the concealment and diversity of the attack behaviors faced by CEs, it is even more challenging to collect comprehensive and representative attack samples. As a result, some methods that rely on a large number of labeled attack samples for supervised learning are unable to fully learn the characteristic patterns of attacks targeting CEs when there is a lack of sufficient samples.

Secondly, the insufficient expression ability of traffic features is a key issue. Current technologies have obvious limitations in describing the traffic interaction patterns of abnormal attacks targeting CEs, and they are unable to fully capture the complex dynamic relationships therein. Taking the group of smart home appliances as an example, existing methods can often only analyze some superficial features of the traffic among devices such as smart refrigerators and smart TVs, and it is difficult to dig deeper into the attack interaction logic and potential security threats hidden behind the traffic. For instance, attackers may take advantage of loopholes in the communication protocols of smart home appliances, and manipulate smart TVs to send malicious instructions to other devices, while existing technologies can hardly detect such deep-seated hidden dangers.

Finally, the deployment requirements for smart home gateways with limited resources [6] are extremely challenging. Smart home gateways usually have limitations in terms of computing resources, storage capacity, and energy supply [7], which are closely related to a large number of connected CEs. On the one hand, the massive real-time traffic generated by numerous CEs, such as home VR/AR game boxes, needs to be processed quickly by the gateway, imposing huge pressure on its computing resources. On the other hand, the limited storage space of the gateway can hardly accommodate the complex model parameters corresponding to the attack characteristics of different CEs.

To overcome the limitations of existing technologies and improve the detection ability of the home smart gateway against the intrusion behaviors of CEs, we are in urgent need of a brand-new method and model. This method and model can not only effectively solve the problem of obtaining attack samples and enhance the expression ability of traffic features, but also adapt to the environment where the smart home gateway has limited resources and is closely connected to numerous CEs.

B. Contributions

The embedded intrusion detection framework based on FIR-GNN proposed by us aims to fully utilize the flow interaction behavior and time-related payload interaction information to provide support for the security protection of the home smart gateway.

Firstly, we propose a traffic graph named FIRG, which is a directed graph that describes the traffic interaction process between the IPs of various devices at the topological level. With the help of FIRG, the traffic behavior is transformed into a directed graph of network flows, so as to better capture the traffic interaction patterns based on source and destination IPs under the constraint of time correlation. Secondly, we propose a network intrusion detection model named FIR-GNN (Flow Interaction Relationship Graph Neural Network) based on Graph Neural Network. The edge-wise graph attention mechanism is introduced to characterize the microscopic interaction characteristics of network flows through packet direction and length information. At the same time, the labeled nodes can aggregate the features of the surrounding unlabeled nodes to enhance their feature expression ability, and semi-supervised learning can be achieved by only calculating the loss of the labeled nodes during training, thereby reducing the dependence of the model on labeled samples. In addition, we use a method of selecting a subset of traffic features based on SHAP. The importance of the features is measured by calculating their marginal contribution (Shapley value) to the prediction result, and then the best subset of features is selected based on the importance of the feature. Thus, the number of input features of the model can be significantly reduced to improve training time and computational resource requirements.

In summary, the main contributions of FIR-GNN include:

- A novel topological level traffic graph representation of FIRG, which can capture the traffic interaction patterns based on IPs under the constraint of time correlation.
- Based on the construction method of FIRG, we proposed FIR-GNN NIDS method, achieving semi-supervised learning on a small number of labeled samples.
- A feature selection method based on SHAP, which selects a subset of features based on feature importance, thereby reducing the resource consumption of the model.
- The experimental results show that on the BoT-IoT [8] and CICIDS2017 [9] data, FIR-GNN can achieve a 3-5% improvement in classification performance compared to other methods.

The rest of this paper is structured as follows. Section II reviews related work on GNN-based NID. Section III illustrates the CIoT security framework in a smart home scenario. Section IV details the FIR-GNN NIDS method, including FIRG construction and the structure of the FIR-GNN model. Section V evaluates our method using the BoT-IoT and CICIDS2017 datasets. Finally, Section VI summarizes our findings and suggests future research directions.

II. Related Works

A. Netwrok Traffic Graph Construction

The construction of network traffic graphs [10]–[17] is the process of converting network communication behaviors into graphical structures, aiming to capture the characteristics and patterns of network traffic through graphical representations. Some studies, such as FRG proposed by Jiang et al. [10], integrate packet-level details and flow-level relationship data. The work of Zheng et al. [11] has successfully combined the statistical properties of network flows with their structural context to capture the homogeneity of links and the characteristics of specific flows, but this method may not be able to fully express the dynamic interactions among flows. Methods such as MAppGraph [12] and TIG [13] focus on static graph structures and ignore the time dimension and dynamic changes between traffic flows, which is a significant drawback when facing complex scenarios such as multi-step attacks. CGNN [14] captures the complex associations between packets using PMI as edge features, and TFE-GNN [15] designs a dual embedding layer to handle header and payload byte information. Although these methods enhance the understanding of the internal information of traffic, their capture of the interactions among flows is still limited. The work of Huoh et al. [16] and Xu et al. [17] attempts to introduce the time dimension into traffic graphs, but they often rely on a single application flow or local subgraphs.

The existing traffic graph construction methods have limitations in capturing the interactions between flows, time series characteristics, and spatial relationships, which restricts their performance in practical applications.

B. GNN-based Network Intrusion Detection for IoT

In the field of the Internet of Things (IoT) [18], Graph Neural Networks, as part of Intrusion Detection Systems [19]–[29], have already demonstrated the advantages in dealing with complex network traffic patterns and capturing the relationships among nodes. The method proposed by Ning et al. [19] improves the cross-domain generalization ability of the model through semi-supervised learning, transfer learning [20], and domain adaptation. However, it usually requires a large amount of labeled data. However, it usually requires a large amount of labeled data. The Intelligent Intrusion Detection System (IIDS) proposed by Anbalagan et al. [21] focuses on 5G vehicle networking [22], improves the detection performance by utilizing enhanced convolutional neural networks and hyperparameter optimization techniques. Chang et al. [23] introduce residual connections by extending the GraphSAGE and GAT algorithms to deal with the class imbalance problem, but this method may fail to fully utilize the interaction relationships between flows. Nie et al. [24] detect abnormal traffic fluctuations in the Internet of Vehicles (IoV) [25] by analyzing the link load behaviors of Road Side Units (RSUs). Faced with limited labeled samples [26], the FT-GCN method proposed by Deng et al. Faced with limited labeled samples [26], the FT-GCN method proposed by Deng et al. [27] improves statistical features by constructing traffic graphs with interval constraints and introducing a spatial attention mechanism at the node level.

Existing methods have some limitations, especially in dealing with resource-constrained environments and handling imbalanced classes.

C. Feature Selection for Network Intrusion Detection

In network intrusion detection systems, feature selection [30]–[34] is a crucial step for improving detection performance, reducing the consumption of computational resources. IGRF-RFE [30] filters by combining information gain and random forest and optimizes feature selection through the recursive feature elimination of MLP. The FS-DL method of Zhang et al. [31] removes redundant features through standard deviation and association rule mining, and is applicable to online abnormal traffic detection of SDN controllers. Turukmane et al. [32] use the modified singular value decomposition (M-SvD) for feature extraction and optimize features through the Oppositional Northern Goshawk Optimization algorithm (ONgO). Thakkar et al. [33] select features with high discrimination and deviation through a statistical significance method that fuses standard deviation and mean-median difference, but this method may ignore the interaction among features. Wu et al. [34] have developed a feature selection method that combines FRS and FKD to optimize memory usage and efficiency, and it is combined with GAT for real-time network data update.

The existing research work has put forward a variety of feature selection methods, but they have some limitations in measuring the importance of features and have poor interpretability.

III. The CIoT Security Framework for CEs in Smart Home

As shown in Fig. 1, from the data collection and interaction at the consumer electronic layer to the security detection and traffic management at the network layer, and then to the remote monitoring and management at the cloud service layer, all aspects of this architecture work closely together to jointly ensure the security, stability, and convenience of the smart home environment.

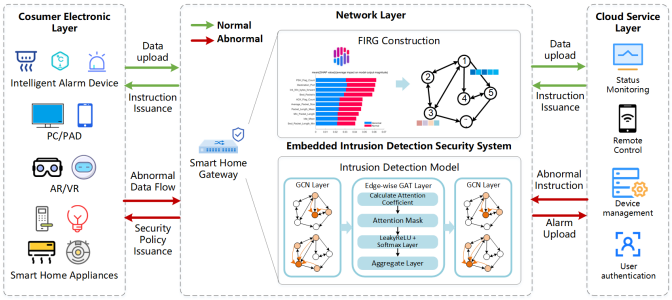


Fig. 1. The CIoT security framework for CEs in smart home.

The CEs layer contains a variety of smart home devices, such as intelligent alarm devices, PC/PAD, AR/VR, smart home appliances, etc. They will upload their own operating

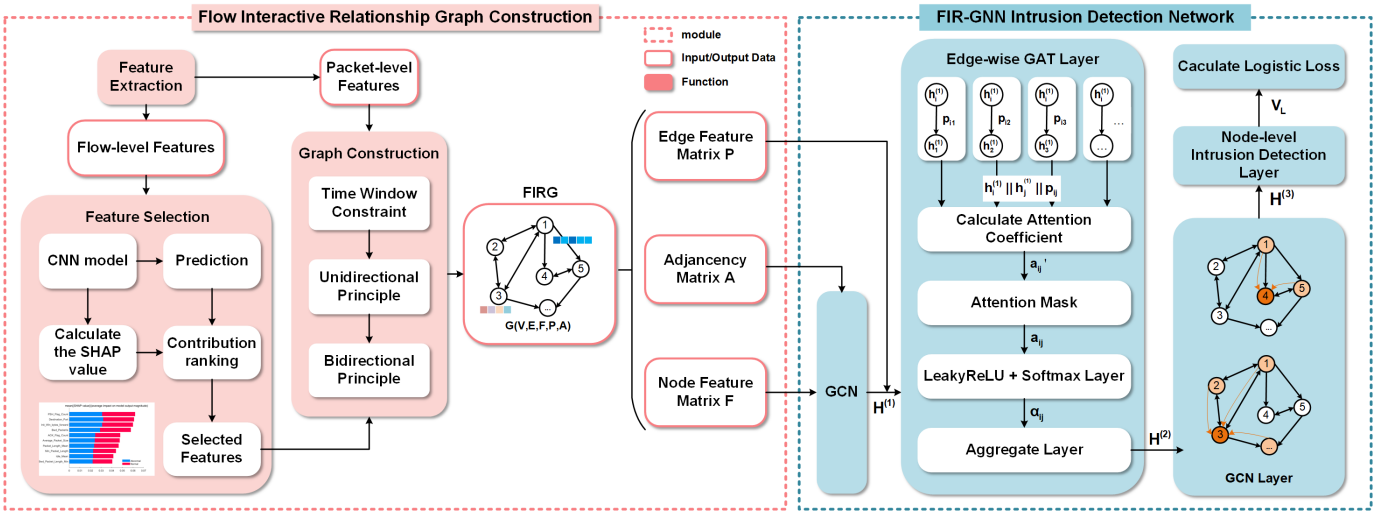


Fig. 2. The intrusion detetction security system for CEs embedded in smart home gateway.

status data to the network layer and can also receive security policies from the network layer. The network layer shoulders the important task of data transmission and ensures the smooth and secure information exchange between the CEs device layer and the cloud service layer. The smart home gateway is the core hub and contains an embedded intrusion detection security system. This system constructs traffic data into FIRG traffic graphs and inputs them into the FIR-GNN intrusion detection model to identify potential cybersecurity threats. The cloud service layer integrates a rich variety of functional services such as status monitoring, remote control, device management, and user authentication.

1) Normal situation: The operating data and status information generated by the CEs layer are uploaded through the smart home gateway at the network layer for functions such as status monitoring at the cloud service layer. When users perform remote operations or trigger instructions according to preset rules, the cloud service layer sends instructions to the smart home gateway. After receiving the instructions from the cloud service layer, the smart home gateway transmits them to the corresponding CEs, prompting the CEs to perform actions such as turning on/off or adjusting, thus completing the remote control loop.

2) Abnormal situation: Once the embedded intrusion detection system at the network layer detects abnormal traffic or signs of attacks, it will upload abnormal instructions or alarm information to the cloud service layer. Meanwhile, according to the preset security policies, it will automatically send instructions to the CEs Layer, taking emergency measures such as traffic blocking, device isolation, password resetting, etc., to promptly contain the spread of security incidents and minimize losses to the greatest extent.

IV. The Proposed Method FIR-GNN

Through the above elaboration on the CIoT security framework in smart homes, we have gained a clear understanding of the security architecture and data interaction processes in the smart home environment. To effectively address these challenges and enhance the intrusion detection capabilities for consumer electronics in smart home networks, we propose a novel method FIR-GNN. This section describes the overall framework and workflow of our proposed method first as shown in Fig. 2, then demonstrates each part of the modeling process, including the construction of the Flow Interaction Relationship Graph (FIRG), and the structure of the FIR-GNN model, respectively.

A. Preliminary

1) Network Traffic Flow Definitions:

A traffic flow is identified by a five-tuple consisting of the source address, destination address, source port, destination port, and the TCP/UDP protocol. A flow is made up of multiple packets that move in opposite directions.

2) Graph Notations:

Graph $G = (V, E, \mathbf{F}, \mathbf{P}, \mathbf{A})$, where $V = V_L \cup V_U = \{v_1, v_2, ..., v_N\}$ denotes the set of nodes, with V_L representing labeled nodes and V_U representing unlabeled nodes. $E = \{e_{ij} = (v_i, v_j) | v_i, v_j \in V\}$ describes the edges in the FIRG. The nodes and edges are characterized by feature matrices, $\mathbf{F}$ for the nodes, and $\mathbf{P}$ for the edges, which represent their properties and attributes. We use $\mathbf{A}$ to denote an adjacency matrix and $N_i = \{v_j | e_{ij} \in E\}$ to denote the neighbor nodes of v_i.

B. FIRG Construction

In this section, we need to convert the original PCAP files into a graph structure, where the nodes represent the flows identified by the five-tuple, and the edges denote interactions between the flows, as shown in Fig. 3. We create a flow interaction relationship graph to describe the traffic interaction process between IP hosts, extracting similarity relationships based on link homogeneity among network flows, thus transforming network intrusion detection into a node classification task. We aim to convert the Traffic Trace into a directed graph by mining the interaction characteristics between flows. Specific details are provided in Algorithm 1.

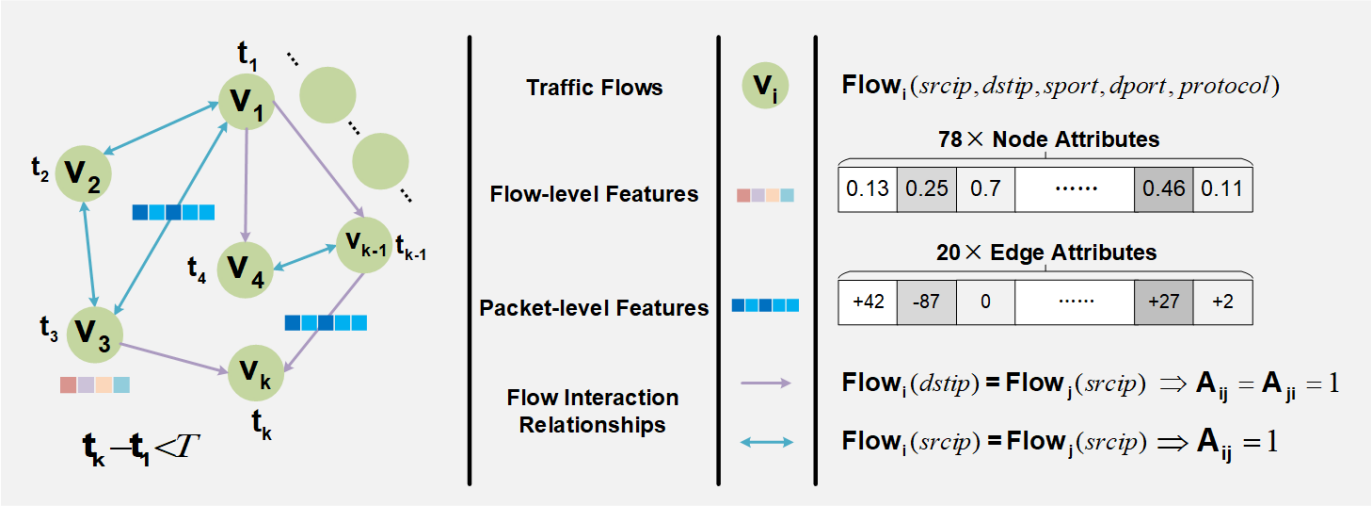


Fig. 3. A graph-structured representation of a FIRG's overall structure, as well as detailed information on the nodes and edges.

*1) **Feature Extraction**:*

The traffic characteristics here consist mainly of two types: one type is the flow level characteristics extracted using the CICFlowMeter tool, including flow length, flow duration, the number of packets in the flow, and their mean, variance, maximum, and minimum values, denoted as $[f_1, f_2, \ldots, f_m]$. The other type is the packet length feature sequence that is extracted by using the flowcontainer tool. In this case, the packet direction determines the sign of the packet length, and it is denoted as $[p_1, p_2, \ldots, p_n]$, where p_i represents the length and direction feature of the i-th packet within the flow. When extracting the packet length sequence, irrelevant packet information, such as the three-way handshake in TCP, is not considered.

*2) **SHAP-based interpretable feature selection**:*

After obtaining the raw traffic features by processing PCAP files, additional operations such as feature cleaning, Min-Max normalization, and feature selection are required. This paper uses a SHAP-based interpretable feature selection method to select flow-level features, aiming to reduce the model complexity and computational load.

Firstly, a simple-structured CNN model is pre-trained on the pre-processed dataset. This model consists of convolutional layers, pooling layers, and fully-connected layers. During the training process, the Adam optimizer and cross-entropy loss function are used. Subsequently, the trained CNN model is used to predict the test dataset. Based on the prediction results, Shapley values are calculated to determine the contribution of each feature to the final outcome, and then a descending ranking of feature importance is generated.

Considering the performance and storage limitations of the smart home gateway, the appropriate value of K is determined through multiple experiments. The top K-ranked features are selected. When constructing the FIRG for the FIR-GNN model, these features are used to populate the node feature matrix F. During the training and inference processes of the FIR-GNN model, these features are used to calculate the interactions between nodes, conduct semi-supervised learning, and classify network traffic. This not only reduces the computational and storage requirements but also improves the intrusion detection performance.

*3) **Graph Construction**:*

In this process, an adjacency matrix $\mathbf{A}$ is generated to describe the structural information of the graph, along with a node feature matrix $\mathbf{F}$ and an edge feature matrix $\mathbf{P}$ to represent the features of the graph. The specific construction methods are as follows:

Node Feature Matrix F. In the FIRG, a node represents a flow, and the best flow-level features $[f'_1, f'_2, \ldots, f'_K]$ selected by SHAP-based interpretable feature selection are used as node features. Based on the experimental results, we choose the optimal characteristic dimension $K = 20$.

Adjacency Matrix A. In the FIRG, an edge represents the similarity between any two flows. Here, we use $Flow_i$ to denote the i th flow. The elements $\mathbf{A_{ij}}$ of the adjacency matrix $\mathbf{A}$ are set according to the following three principles to indicate whether an edge exists between nodes v_i and v_j, and whether it is bidirectional:

(1) Time Window Constraint. We introduce a hyperparameter T to limit the time window for flows. An edge will only be considered between two flows if they appear simultaneously within a time window of length T, that is, $t_j - t_i < T$. This is because as the time interval increases, the correlation between flows decreases. Furthermore, if the time window T is too large, it can lead to an excessively dense FIRG, resulting in significant memory consumption for edge storage.

(2) *If* $Flow_i.dstip = Flow_j.srcip$, *then* $\mathbf{A_{ij}} = |\mathbf{t_i} - \mathbf{t_j}|, \mathbf{A_{ji}} = \mathbf{0}$. If an attack targets a destination node, intermediate nodes, such as in a DoS attack, might also be impacted.

(3) *If* $Flow_i.srcip = Flow_j.srcip$, *then* $\mathbf{A_{ij}} = \mathbf{A_{ji}} = |\mathbf{t_i} - \mathbf{t_j}|$. When $Flow_i$ and $Flow_j$ share the same source IP, it is likely that they exhibit similar activities or applications.

Edge Feature Matrix P. In constructing edges, we consider two crucial aspects. Firstly, we take into account the time interval between the start times of flows and the correlation of starting IPs. This helps us simulate interactions between flows, as the temporal and IP relationships can provide valuable

insights into the possible connections and dependencies among different flows.

Secondly, the packet length feature plays a vital role in characterizing network traffic. Different types of traffic, whether it is normal business traffic or malicious attack traffic, display distinct distribution patterns in terms of packet length. By extracting and combining the packet length features of the first N packets, we can zoom in on the interaction characteristics of the traffic at the micro level during the initial stage. At the onset of network communication, the changes in packet length can disclose some essential attributes of the traffic. For instance, normal communication typically exhibits a relatively stable packet length pattern, whereas attack traffic might present abnormal combinations like the frequent alternation between extremely large packets and small packets.

Algorithm 1 The construction of FIRG.

Input: PCAP file;
Output: The $FIRG = (V, E, \mathbf{F}, \mathbf{P}, \mathbf{A})$;
▷ **Step 1: Feature Extraction**
Extract flow-level features using CICFlowMeter:
Flow length, Flow duration etc. $\rightarrow [f_1, f_2, \ldots, f_m]$
Extract packet-level features using Flowcontainer:
Packet length sequence $\rightarrow [p_1, p_2, \ldots, p_n]$
▷ **Step 2: SHAP-based Interpretable Feature Selection**
Remove duplicates, missing items, and anomalies.
Perform Min-Max normalization.
Compute Shapley values for each flow-level feature:

$$\text{Score}(f_i) = \sum_{j=1}^{m} \text{Shapley}(f_{ij})$$

Select the top K most important flow-level features.
▷ **Step 3: Graph Construction**
Initialize node feature matrix $\mathbf{F}$ using selected features $[f'_1, f'_2, \ldots, f'_K]$.
Initialize adjacency matrix $\mathbf{A}$.
Initialize edge feature matrix $\mathbf{P}$.
for each pair of flows $(Flow_i, Flow_j)$ **do**
  if $t_j - t_i < T$ **then**
    if srcIP of $Flow_i$ = srcIP of $Flow_j$ **then**
      $1 \rightarrow \mathbf{A_{ij}}, \mathbf{A_{ji}}$
      $[p_1^i, p_2^i, \ldots, p_n^i, p_1^j, p_2^j, \ldots, p_n^j] \rightarrow \mathbf{P_{ij}}, \mathbf{P_{ji}}$
    end if
    if dstIP of $Flow_i$ = srcIP of $Flow_j$ **then**
      $1 \rightarrow \mathbf{A_{ij}}$
      $[p_1^i, p_2^i, \ldots, p_n^i, p_1^j, p_2^j, \ldots, p_n^j] \rightarrow \mathbf{P_{ij}}$
    end if
  end if
end for

By combining these packet length features as edge features, the model is empowered to grasp the interaction details between traffic more precisely, thereby enhancing its capacity to distinguish between various traffic relationships. Consequently, for edge feature construction, we concatenate the packet length feature sequences of the first N packets from both flows to effectively capture their packet-level interaction features.

The method in this paper conducts a series of experiments, compares the performance and memory usage of the model under different values, and then selects a compromise value $T = 20$ after weighing various factors, so as to balance the performance and memory consumption of the model. We set $N = 5$, padding with zeros if there are fewer than 5 packets, and truncating if there are more than 5. Selecting the first 5 packets can ensure that enough packet-level interaction features are captured to describe the relationships between traffic flows, while at the same time not affecting the overall performance of the model due to being overly large and complex.

C. The Structure of FIR-GNN Model

In this section, we introduce the Edge-wise Intrusion Detection Network FIR-GNN for feature learning, which operates directly on graph-structured data. Two layers of GCN and one layer of Edge-wise GAT are employed as core algorithms for classifying each node in the graph. During this process, an edge-wise attention mechanism is applied to calculate an attention matrix for edges which aggregates neighbor node information through pooling layers. Then GCN is utilized to extract deeper-level features for final node-level classification purposes.

1) Graph Convolutional Network Layer:

The GCN layer comprises both node features and adjacency information. After performing Laplacian eigenvalue decomposition, ReLU activation is applied to obtain the first layer feature embedding $\mathbf{H}^l$. The formula is as follows:

$$\mathbf{H}^l = \text{ReLU}(\hat{\mathbf{A}}\mathbf{F}\mathbf{W}^l) \tag{1}$$

We transform the input adjacency matrix $\mathbf{A}$ into a normalized Laplacian matrix $\hat{\mathbf{A}}$ using the following formula:

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\mathbf{1/2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\mathbf{1/2}} \tag{2}$$

In this context, $\tilde{\mathbf{A}}$ is computed as $\mathbf{A} + \mathbf{I_N}$, where $\mathbf{I_N} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ represents the identity matrix. $\tilde{\mathbf{D}}$ is given by $\tilde{\mathbf{D}}\mathbf{ii} = \sum \mathbf{v_j}\tilde{\mathbf{A}}_{\mathbf{ij}}$, and $\tilde{\mathbf{D}}^{-\frac{\mathbf{1}}{\mathbf{2}}}$ is the inverse of the square root of the diagonal matrix.

2) Edge-wise Graph Attention Network layer:

We first concatenate the node feature matrices with the edge feature matrix and then apply a linear transformation. Subsequently, a MASK operation is used to select the neighboring nodes of v_i. The formula is as follows:

$$a_{ij}^{(l)} = \vec{a}^T[\mathbf{W}^{(l)}\vec{h}_i^{(l-1)} || \mathbf{W}^{(l)}\vec{h}_j^{(l-1)} || \mathbf{W}^{(l)}\vec{p}_{ij}] \tag{3}$$

$$\text{MASK}(a_{ij}) = \begin{cases} a_{ij} & \text{if} \mathbf{A_{ij}} \neq 0 \\ 0 & \text{if} \mathbf{A_{ij}} = 0 \end{cases} \tag{4}$$

where $\mathbf{W}^{(2)}$ represents the current weight matrix; $\vec{a}$ is a feedforward neural network; $\vec{h}_i^{(l-1)}$ and $\vec{h}_j^{(l-1)}$ respectively represent the first layer embeddings of the central node v_i and one of its adjacent nodes v_j; $\vec{p}_{ij}$ refers to the initial feature embedding of the edge e_{ij}.

Next, we apply the LeakyReLU activation function as σ to perform a non-linear transformation on it and normalize using the softmax function. The purpose of doing this is to allow the model to better focus on important neighboring nodes.

$$\begin{aligned} \alpha_{ij}^{(l)} &= softmax(\sigma(\text{MASK}(a_{ij}^{(l)}))) \\ &= \frac{\exp(\sigma(\text{MASK}(a_{ij}^{(l)})))}{\sum_{v_j \in N_i} \exp(\sigma(\text{MASK}(a_{ij}^{(l)})))} \end{aligned} \tag{5}$$

Finally, we multiply the feature vectors of all neighboring nodes of node v_i by their corresponding attention coefficients and sum them up to obtain a new node feature vector $h_i^{(2)}$. This process can be seen as a weighted fusion of the features of node v_i, where the weights are determined by the attention coefficients.

$$h_i^{(l)} = \mathrm{ReLU}(\sum_{v_j \in N_i} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \vec{h}_j) \tag{6}$$

V. Experimental Evaluation

A. Experimental Settings

1) Experimental Environment: We used a computer equipped with a 13th Gen Intel® Core™ i7-13700KF × 24 processor, 16GB of memory, and an NVIDIA RTX 4080 graphics card. The software environment includes CUDA 12.2 and CUDNN 9.1.0. Python 3.9.12 was chosen as the programming language, and the PyG framework was used to implement the GNN.

2) Datasets: We compare different network traffic classification algorithms on CICIDS2017 [8] and BoT-IoT [9] datasets. The BoT-IoT dataset originally contained 8 categories. However, due to the extremely small number of samples in the UDP Denial of Service (UDP DoS) and Keylogging categories, we excluded them from our analysis. The CICIDS2017 dataset contains 13 types of attack samples. For the specific categories of these two datasets, please refer to Table III and Table IV. On both datasets, we did not take any class balancing measures to augment the samples. We selected 3000 samples from each attack category whenever possible; if the number of samples was less than 3000, we used all available samples. In addition, 80% of the datasets were used for training and 20% for testing.

3) Implementation Details and Baselines: This paper comprehensively evaluates the proposed FIR-GNN [1] intrusion detection method from the perspectives of classification performance and resource consumption, conducting the following three sets of experiments: comparison of classification performance of different methods, comparison of different feature selection methods, and comparison of different parameter settings. Table I presents the hyperparameter configurations. We set the maximum time interval T to 30 seconds, and the packet length sequence for a single flow is set to 10, resulting in edge feature dimensions of 20. The ratio of labeled samples is default set to 0.8, and in the semi-supervised learning experiments with limited labeled samples, it is varied from 0.1 to 0.9 in increments of 0.1. All models are implemented using PyTorch, and each experiment is independently run 10 times. Our approach FIR-GNN is compared with three GNN-based models (GCN, GAT, GCN+GAT) and four DL-based models (CNN, AE, VAE, ET-BERT [35]). GCN and GAT both utilize the same FIR graph construction method as ours. The GCN node classification model consists of two layers of GCN, and the GAT model comprises two layers of single-head GAT, with the latent feature dimensions set to 32 for both.

[1]Source code available at https://github.com/MengyiFu/FIR-GNN.

TABLE I
THE HYPERPARAMETERS OF EXPERIMENTS.

Parameters	Values
Number of packets	5
Time window/s	20
Label ratio	0.1-0.9
Hidden channels	32
GAT heads	4
Optimizer	Adam
Learning rate	0.01
Loss	CrossEntropyLoss
Train epoch	300

B. Experimental Results and Discussion

1) Comparison of classification performance of different methods: Evaluate the performance of different neural network methods in various network attack detection scenarios. Conduct a comparative analysis of the accuracy, recall, and F1 value of methods such as CNN, AE, VAE, ET-BERT, GAT, GCN, GCN+GAT, and FIR-GNN when detecting different types of network attacks, so as to verify the effectiveness of the FIR-GNN method. As shown in Table II, FIR-GNN's overall performance is excellent, second only to ET-BERT and better than others. CNN, AE, and VAE have relatively low performances with accuracy and F1 score below 0.97.

TABLE II
THE COMPARISON OF DIFFERENT IDS METHODS ON BOT-IOT AND CICIDS2017 DATASETS.

Datasets	BoT-IoT				CICIDS2017			
Methods	AC	PR	RC	F1	AC	PR	RC	F1
CNN	0.9690	0.9693	0.9690	0.9690	0.9337	0.9414	0.9337	0.9305
AE	0.9440	0.9444	0.9440	0.9440	0.9395	0.9422	0.9395	0.9359
VAE	0.9612	0.9619	0.9612	0.9611	0.9724	0.9688	0.9724	0.9701
ET-BERT [35]	0.9998	0.9998	0.9998	0.9998	0.9986	0.9986	0.9986	0.9986
GAT	0.9465	0.9443	0.9465	0.9449	0.9670	0.9568	0.9670	0.9617
GCN	0.9491	0.9479	0.9491	0.9475	0.9741	0.9658	0.9741	0.9693
GCN+GAT	0.9523	0.9494	0.9523	0.9503	0.9641	0.9559	0.9641	0.9594
FIR-GNN	0.9926	0.9903	0.9926	0.9913	0.9859	0.9841	0.9859	0.9849

In Table III for the BoT-IoT dataset's attack detection, FIR-GNN reaches high levels (close to 1) in metrics like PR, RC and F1. For attacks like UDP DDoS, TCP DDoS and HTTP DDoS, its precision, recall, and F1 value are all 1, showing few misjudgments or missed detections. FIR-GNN performs better in almost all attack types than others. Although GCN's accuracy (0.9491) is relatively high, FIR-GNN (0.9926) is superior. In the ServiceScan detection, compared to GCN + GAT's metrics, FIR-GNN's better metrics also indicate its detection performance superiority. In the CICIDS2017 dataset (Table IV), FIR-GNN's precision is excellent in most attack detections.

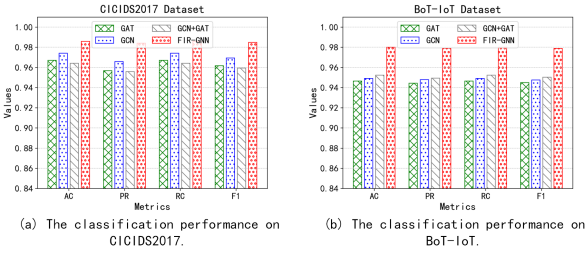
Fig. 4 presents the performance evaluation results of four GNN-based IDS methods across two datasets. It shows that the performance of FIR-GNN is superior to other methods across all metrics, with GCN performing next best, while GAT and GCN+GAT exhibit relatively weaker performance. In the BoT-IoT dataset, FIR-GNN achieved an accuracy improvement of approximately 3.55% compared to GAT.

TABLE III
THE CLASSIFICATION REPORT OF FIR-GNN ON THE BOT-IOT DATASET.

Methods	AC	Metrics	HTTP DDoS	HTTP DoS	OSFingerprint	ServiceScan	TCP DDoS	TCP DoS	Data Theft	UDP DDoS
GAT	0.9465	PR	0.9321	0.8776	0.9167	0.4583	0.9983	0.9804	0.7183	0.9934
		RC	0.87	0.9317	0.6875	0.5238	1	1	0.68	1
		F1	0.9	0.9038	0.7857	0.4889	0.9992	0.9901	0.6986	0.9967
GCN	0.9491	PR	0.933	0.8505	0.9231	0.7917	0.9983	0.9967	0.8933	0.9885
		RC	0.835	0.9383	0.75	0.9048	1	1	0.8933	1
		F1	0.8813	0.8922	0.8276	0.8444	0.9992	0.9983	0.8933	0.9942
GCN+GAT	0.9523	PR	**0.9397**	0.8877	0.7333	0.4286	0.995	0.995	0.6914	1
		RC	0.8833	**0.9483**	0.6875	0.2857	1	1	0.7467	1
		F1	0.9107	0.917	0.7097	0.3429	0.9975	0.9975	0.7179	1
FIR-GNN	0.9801	PR	0.936	**0.9964**	**1**	**0.9444**	**1**	**1**	0.8022	**1**
		RC	**0.9983**	0.925	**0.9375**	**0.8095**	**1**	**1**	0.9733	**1**
		F1	**0.9661**	**0.9594**	**0.9677**	**0.8718**	**1**	**1**	0.8795	**1**

TABLE IV
THE CLASSIFICATION REPORT OF FIR-GNN ON THE CICIDS DATASET.

Methods	AC	Metrics	Benign	Bot	Brute Force	DDoS	FTP Patator	GoldenEye	Heartbleed	Hulk	Infiltration	PortScan	SSH Patator	Slowhttptest	Slowloris
GAT	0.967	**PR**	0.9272	0.9031	0	0.9967	0.8543	0.9772	0	0.9983	0	0	0.9453	0.9919	0.9914
		RC	0.9333	0.9223	0	1	0.9923	1	0	0.9917	0	0	0.9167	0.9919	1
		F1	0.9302	0.9126	0	0.9983	0.9181	0.9885	0	0.995	0	0	0.9308	0.9919	0.9957
GCN	0.9741	**PR**	**0.9626**	0.9204	0	0.9983	0.8667	0.9772	**1**	0.995	**1**	0	0.9621	0.9736	0.9978
		RC	0.945	**0.9399**	0	1	1	1	**0.3333**	0.995	0.3333	0	0.9621	1	1
		F1	0.9537	0.93	0	0.9992	0.9286	0.9885	**0.5**	0.995	0.5	0	0.9621	0.9866	0.9989
GCN+GAT	0.9641	**PR**	0.9133	0.8566	0.5	0.9983	0.8591	0.9983	0	0.9967	0	0	0.9549	0.9762	0.9987
		RC	0.9133	0.8869	0.0769	1	0.9846	1	0	0.995	0	0	0.9621	1	1
		F1	0.9133	0.8715	0.1333	0.9992	0.9176	0.9992	0	0.9958	0	0	0.9585	0.9879	0.9989
FIR-GNN	0.9859	**PR**	0.9586	**0.9331**	**0.6875**	**1**	**1**	**0.9984**	0	**0.9983**	0.5	**1**	**1**	**1**	**1**
		RC	**0.965**	0.9364	**0.8462**	**1**	**1**	**1**	0	**1**	**0.5**	**0.9412**	**0.9924**	**1**	**1**
		F1	**0.9618**	**0.9347**	**0.7586**	**1**	**1**	**0.9992**	0	**0.9992**	**0.5**	**0.9697**	**0.9962**	**1**	**1**



(a) The classification performance on CICIDS2017. (b) The classification performance on BoT-IoT.

Fig. 4. The comparison of classification performance.

2) Comparison of different feature selection methods: The purpose of the experiment is to compare the model performance of different feature selection methods, including SHAP-based methods, Univariate Feature Selection (UFS), Recursive Feature Elimination (RFE), Random Forest Importance (RFI), and Information Gain (IG), when the number of feature selections is 5, 10, 15, and 20 respectively on different datasets (BoT-IoT and CICIDS2017), so as to prove the effectiveness of the SHAP-based feature selection method.

Under the BoT-IoT dataset, as the node feature count rises from 5 to 20, the accuracy stays relatively high, shown in Table V. At 5 features, it's 0.9648, and at 20, it's 0.9901 with little fluctuation, showing the method can screen features well for stable model performance. The CICIDS2017 dataset shows a similar pattern, with an accuracy of 0.9505 at 5 features and 0.9806 at 20, reflecting its adaptability and stability across different datasets and feature scales.

Fig. 5 (a) and (e) present the FIR-GNN model's accuracy

TABLE V
THE FIR-GNN IDS ACCURACY OF DIFFERENT FEATURE SELECTION METHODS.

Datasets	Node Features	SHAP	UFS	RFE	RFI	IG
BoT-IoT	5	0.9648	0.8686	0.9686	0.893	0.9292
	10	0.9814	0.9811	0.9872	0.9362	0.9279
	15	0.9872	0.992	0.9846	0.9849	0.9641
	20	0.9901	0.9923	0.9897	0.9862	0.9885
CICIDS2017	5	0.9505	0.9181	0.9641	0.973	0.9455
	10	0.9757	0.9767	0.9741	0.9733	0.9764
	15	0.9775	0.9741	0.9806	0.972	0.9738
	20	0.9806	0.9801	0.9801	0.9806	0.9775

with five feature selection algorithms for different feature counts. The SHAP-based method often has higher accuracy than UFS in both datasets. For example, at 5 node features. It also shows good performance compared to RFE, RFI, and IG, having unique advantages in considering feature interactions and contributions, and can mine better feature subsets, while other methods have limitations.

3) Comparison of different parameter settings: Analyze the sensitivity of the FIR-GNN intrusion detection method to three parameters (time window T, number of data packets N, and number of node features K) so that appropriate parameter combinations can be selected according to specific circumstances in practical applications. Table VI and Table VII presents the impact of varying parameters, Time Window T, Number of Packets N and Number of Node Features K, on the performance of the system in terms of Max GPU Used

TABLE VI
THE COMPARISON OF DIFFERENT PARAMETER SETTINGS ON THE BOT-IOT DATASET.

Parameters		Max GPU Used (MB)	FIRG Data Size (MB)	Training Time (s)	Accuracy
Time Window T	10	6793.72	1372.9339	64.8093	0.9567
	20	7377.01	1490.5101	72.8175	0.9936
	30	7772.66	1571.4549	84.6054	0.9933
	40	8277.63	1673.3657	85.9352	0.9923
	50	8590.09	1736.6289	89.5583	0.9939
Number of Packets N	3	6276.96	574.4096	60.3539	0.9936
	5	6703.81	859.2796	67.1918	0.9936
	10	7772.66	1571.4549	84.6054	0.9933
	15	8842.37	2283.6301	89.3253	0.9933
	20	9911.23	2995.8053	101.3873	0.9936
Number of Node Features K	76	7772.66	1571.4549	78.4180	0.9923
	20	7769.25	1568.0644	76.1922	0.9872
	15	7769.01	1567.8264	83.0824	0.9599
	10	7768.72	1567.5290	79.4386	0.9814
	5	7768.42	1567.2316	75.8253	0.9362

TABLE VII
THE COMPARISON OF DIFFERENT PARAMETER SETTINGS ON THE CICIDS2017 DATASET.

Parameters		Max GPU Used (MB)	FIRG Data Size (MB)	Training Time (s)	Accuracy
Time Window T	10	1555.04	310.0489	8.8784	0.9825
	20	2334.43	468.2792	14.1643	0.9851
	30	2927.18	588.4906	18.1867	0.9840
	40	3619.12	728.7949	19.9524	0.9843
	50	4227.79	851.7776	23.6469	0.9846
Number of Packets N	3	2369.68	217.6327	15.5072	0.9846
	5	2528.91	323.5921	16.1638	0.9848
	10	2927.18	588.4906	18.1867	0.9840
	15	3325.06	853.3892	17.3197	0.9843
	20	3723.54	1118.2877	18.1324	0.9851
Number of Node Features K	76	2927.18	588.4906	17.2599	0.9840
	20	2923.01	584.3417	17.2409	0.9775
	15	2922.72	584.0506	18.3156	0.9785
	10	2927.18	583.6866	17.4464	0.9728
	5	2923.01	583.3227	18.8376	0.9471

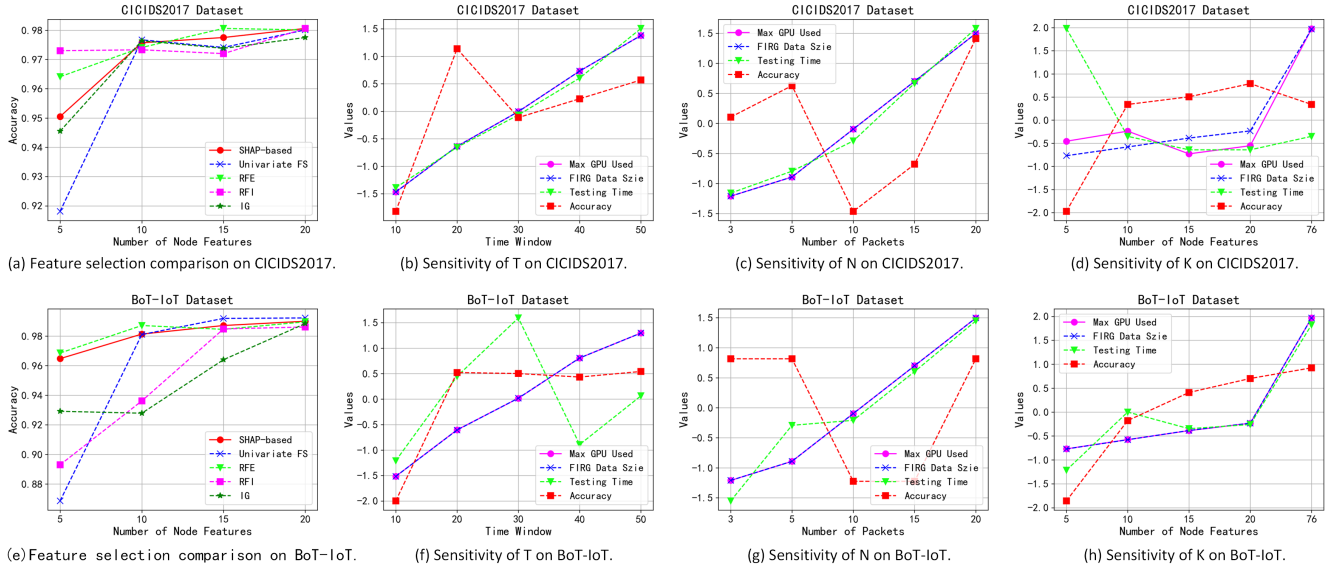
(MB), FIRG Data Size (MB), Testing Time (s), and Accuracy.

Fig. 5 (b), (c), (d), (f), (g), and (h) show the model performance and resource consumption under different parameter settings. Due to the large difference in data magnitudes, the data was standardized using z-score normalization. It is obvious that the polylines of Max GPU used and FIRG Data Size have a relatively high degree of overlap and a similar trend. In the FIR-GNN intrusion detection method, the processing of data and the operation of the model are closely related. When the amount of processed data (FIRG Data Size) increases, more computing resources are usually required to handle this data. As the GPU is the main computing resource, its usage (Max GPU used) will naturally increase as the amount of data increases.

Based on these findings, the optimal parameter settings would need to balance classification performance against resource consumption. When a high level of accuracy is required and resources are sufficient, a larger time window T (such as T = 20) and a greater number of node features K (such as K = 20) can be chosen, because these two parameters can improve the accuracy within an appropriate range. The number of data packets N can be selected according to the actual data situation. Since it has a relatively small impact on the accuracy, it can be appropriately increased when resources permit. If GPU resources and data processing time is limited, a smaller time window T (such as T = 10), a smaller number of data packets N (such as N = 3), and a moderate number of node features K (such as K = 10) can be selected. In this way, the occupation of resources can be controlled to a certain extent while maintaining a relatively high accuracy.

VI. CONCLUSION

This study proposed a novel FIR-GNN framework to address the network security challenges in cyber-physical sys-



(a) Feature selection comparison on CICIDS2017. (b) Sensitivity of T on CICIDS2017. (c) Sensitivity of N on CICIDS2017. (d) Sensitivity of K on CICIDS2017.

(e) Feature selection comparison on BoT-IoT. (f) Sensitivity of T on BoT-IoT. (g) Sensitivity of N on BoT-IoT. (h) Sensitivity of K on BoT-IoT.

Fig. 5. The comparison of different feature selection methods and parameter settings.

tems (CPS), especially in smart home settings. The framework outperforms existing methods in accuracy, precision, recall, and F1 value, and shows high stability. It constructs a directed graph and uses an edge feature self-attention mechanism, combined with semi-supervised learning, to enhance generalization. The SHAP-based feature subset selection reduces the computational cost.

In addition to developing adaptive continuous learning methods for NIDS, there are several potential future development directions for FIR-GNN. As more and more home devices are connected to the network, real-time intrusion detection has become crucial. FIR-GNN can be optimized to further reduce its response time, ensuring that it can quickly identify and respond to threats in high-speed traffic scenarios. Moreover, FIR-GNN can be applied in enterprise IoT environments, where the security requirements are equally strict. By adapting to the larger-scale and more complex network topologies in enterprises, FIR-GNN helps protect enterprise-owned smart devices and sensitive data.

Overall, the FIR-GNN framework has broad prospects for enhancing the security of a wide range of CPS. Continuous research and development in this field are likely to yield more effective security solutions.

ACKNOWLEDGMENTS

The paper is supported by the Suzhou Science and Technology Planning Project under Grant No.SYG202311 and Suzhou Science and Technology Planning Project under Grant No.LHT202326.

REFERENCES

[1] J. Huang, T. Yu, C. Chakraborty, F. Yang, X. Lai, A. Alharbi, and K. Yu, "An energy harvesting algorithm for uav-assisted tinyml consumer electronic in low-power iot networks," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, pp. 7346–7356, 2024.

[2] T. Bhavani, P. VamseeKrishna, C. Chakraborty, and P. Dwivedi, "Stress classification and vital signs forecasting for iot-health monitoring," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 21, no. 4, pp. 652–659, 2024.

[3] P. Wang, F. Ye, and X. Chen, "A smart home gateway platform for data collection and awareness," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 87–93, 2018.

[4] X. Chen, P. Wang, Y. Yang, and M. Liu, "Resource-constraint deep forest-based intrusion detection method in internet of things for consumer electronic," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 2, pp. 4976–4987, 2024.

[5] D. Srivastava, R. Singh, C. Chakraborty, S. K. Maakar, A. Makkar, and D. Sinwar, "A framework for detection of cyber attacks by the classification of intrusion detection datasets," *Microprocessors and Microsystems*, vol. 105, p. 104964, 2024.

[6] X. Zhou, J. Wu, W. Liang, K. I.-K. Wang, Z. Yan, L. T. Yang, and Q. Jin, "Reconstructed graph neural network with knowledge distillation for lightweight anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11 817–11 828, 2024.

[7] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway," *IEEE Access*, vol. 6, pp. 55 380–55 391, 2018.

[8] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[9] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for internet of things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, 2020.

[10] M. Jiang, Z. Li, P. Fu, W. Cai, M. Cui, G. Xiong, and G. Gou, "Accurate mobile-app fingerprinting using flow-level relationship with graph neural networks," *Computer Networks*, vol. 217, p. 109309, 2022.

[11] J. Zheng, Z. Zeng, and T. Feng, "Gcn-eta: High-efficiency encrypted malicious traffic detection," *Security and Communication Networks*, vol. 2022, no. 1, p. 4274139, 2022.

[12] T.-D. Pham, T.-L. Ho, T. Truong-Huu, T.-D. Cao, and H.-L. Truong, "Mappgraph: Mobile-app classification on encrypted network traffic using deep graph convolution neural networks," in *Proceedings of the 37th Annual Computer Security Applications Conference*, 2021, pp. 1025–1038.

[13] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Du, "Accurate decentralized application identification via encrypted traffic analysis using graph neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2367–2380, 2021.

[14] G. Hu, X. Xiao, M. Shen, B. Zhang, X. Yan, and Y. Liu, "Tcgnn: Packet-grained network traffic classification via graph neural networks," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106531, 2023.

[15] H. Zhang, L. Yu, X. Xiao, Q. Li, F. Mercaldo, X. Luo, and Q. Liu, "Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2066–2075.

[16] T.-L. Huoh, Y. Luo, P. Li, and T. Zhang, "Flow-based encrypted network traffic classification with graph neural networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1224–1237, 2022.

[17] R. Xu, G. Wu, W. Wang, X. Gao, A. He, and Z. Zhang, "Applying self-supervised learning to network intrusion detection for network flows with graph neural network," *Computer Networks*, vol. 248, p. 110495, 2024.

[18] J. K. Samriya, C. Chakraborty, A. Sharma, M. Kumar, and S. K. Ramakuri, "Adversarial ml-based secured cloud architecture for consumer internet of things of smart healthcare," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2058–2065, 2023.

[19] J. Ning, G. Gui, Y. Wang, J. Yang, B. Adebisi, S. Ci, H. Gacanin, and F. Adachi, "Malware traffic classification using domain adaptation and ladder network for secure industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 058–17 069, 2021.

[20] X. Zhou, Q. Yang, Q. Liu, W. Liang, K. Wang, Z. Liu, J. Ma, and Q. Jin, "Spatial–temporal federated transfer learning with multi-sensor data fusion for cooperative positioning," *Information Fusion*, vol. 105, p. 102182, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523004980

[21] S. Anbalagan, G. Raja, S. Gurumoorthy, R. D. Suresh, and K. Dev, "Iids: Intelligent intrusion detection system for sustainable development in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 15 866–15 875, 2023.

[22] X. Zhou, W. Liang, A. Kawai, K. Fueda, J. She, and K. I.-K. Wang, "Adaptive segmentation enhanced asynchronous federated learning for sustainable intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6658–6666, 2024.

[23] L. Chang and P. Branco, "Embedding residuals in graph-based solutions: the e-ressage and e-resgat algorithms. a case study in intrusion detection," *Applied Intelligence*, vol. 54, no. 8, pp. 6025–6040, 2024.

[24] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng, and Y. Li, "Data-driven intrusion detection for intelligent internet of vehicles: A deep convolutional neural network-based method," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2219–2230, 2020.

[25] M. Fu, P. Wang, M. Liu, Z. Zhang, and X. Zhou, "Iov-bert-ids: Hybrid network intrusion detection system in iov using large language models," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 1909–1921, 2025.

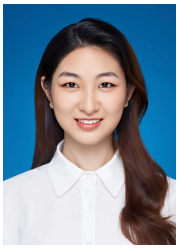
[26] X. Zhou, X. Zheng, X. Cui, J. Shi, W. Liang, Z. Yan, L. T. Yang, S. Shimizu, and K. I.-K. Wang, "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3191–3211, 2023.

[27] X. Deng, J. Zhu, X. Pei, L. Zhang, Z. Ling, and K. Xue, "Flow topology-based graph convolutional network for intrusion detection in label-limited iot networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 684–696, 2022.

[28] Z. Li, P. Wang, and Z. Wang, "Flowgananomaly: Flow-based anomaly network intrusion detection with adversarial learning," *Chinese Journal of Electronics*, vol. 33, no. 1, pp. 58–71, 2024.

[29] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, "Hierarchical adversarial attacks against graph-neural-network-based iot

network intrusion detection system," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9310–9319, 2022.
[30] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, and J. Kwak, "Igrf-rfe: a hybrid feature selection method for mlp-based network intrusion detection on unsw-nb15 dataset," *Journal of Big Data*, vol. 10, no. 1, p. 15, 2023.
[31] L. Zhang, K. Liu, X. Xie, W. Bai, B. Wu, and P. Dong, "A data-driven network intrusion detection system using feature selection and deep learning," *Journal of Information Security and Applications*, vol. 78, p. 103606, 2023.
[32] A. V. Turukmane and R. Devendiran, "M-multisvm: An efficient feature selection assisted network intrusion detection system using machine learning," *Computers & Security*, vol. 137, p. 103587, 2024.
[33] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system," *Information Fusion*, vol. 90, pp. 353–363, 2023.
[34] Y. Wu, L. Nie, X. Xiong, B. Sadoun, L. Yang, and Z. Ning, "Incremental update intrusion detection for industry 5.0 security: A graph attention network-enabled approach," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2004–2017, 2024.
[35] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *Proceedings of the ACM Web Conference 2022*. New York, NY, USA: Association for Computing Machinery, 2022, p. 633–642.

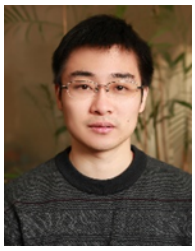


Mengyi Fu (S'23)is currently pursuing a Ph.D degree at Nanjing University of Posts and Telecommunications, Nanjing, China. She received a B.Sc from Nanjing University of Posts and Telecommunications in 2022 and got an MD-PhD qualification with an examination-free recommendation. She has published some research works on IEEE TVT/TCE/NETWORK, etc. Her research includes encrypted traffic identification, deep learning, and traffic prediction.

Pan Wang (M'18) received the BS/MS/Ph.D. degree in Electrical & Computer Engineering from Nanjing University of Posts&Telecommunications, Nanjing, China, in 2001, 2004, and 2013, respectively. He is currently a Full Professor at Nanjing University of Posts & Telecommunications, Nanjing, China. His research interests include AI-powered networking and security in B5G/6G/IoT/Smart Grid/CFN and AI-enabled big data analysis. From 2017 to 2018, he was a visiting scholar at the University of Dayton (UD) in the Department of Electrical and Computer Engineering, OH, USA. He served as a TPC member of the IEEE CyberSciTech Congress. He is also a reviewer for several journals, including IEEE Transaction on Network and Service Management, IEEE Transaction on EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE Internet of Things Journal, IEEE Journal on Selected Areas in Communications, IEEE ACCESS, Computer Networks, Computer&Security, Computer Communications, Engineering Applications of Artificial Intelligence, Big Data Research, etc.

Shidong Liu received the BS/MS/Ph.D. degree in Computer Engineering & Telecommunication Engineering from Dalian University of Technology, Dalian, Nanjing University of Posts & Telecommunications, Nanjing, China, in 1992, 2000, and 2008, respectively. He is currently employed at Institute of Information and Communication Technology, China Electric Power Research Institute. His research interests include AI-powered data network, SDN, IMS and Network operation and maintenance of Electric communication network.

Xuejiao Chen received the B.E. and M.E. degrees from Nanjing University of Posts and Telecommunications (NUPT), majoring in communication and information systems, in 2001 and 2006, respectively. Now she is an associate professor at the Nanjing Institute of Information Vocational Technology. She has been a visiting scholar of the University of Dayton (OH, USA) from 2017 to 2018. Her research areas are B5G/6G network security and artificial intelligence.



Xiaokang Zhou (M'12) is currently an associate professor with the Faculty of Business Data Science, Kansai University, Japan. He received the Ph.D. degree in human sciences from Waseda University, Japan, in 2014. From 2012 to 2015, he was a research associate with the Faculty of Human Sciences, Waseda University, Japan. He was a lecturer/associate professor with the Faculty of Data Science, Shiga University, Japan, from 2016 to 2024. He also works as a visiting researcher with the RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Japan, since 2017. Dr. Zhou has been engaged in interdisciplinary research works in the fields of computer science and engineering, information systems, and social and human informatics. His recent research interests include ubiquitous computing, big data, machine learning, behavior and cognitive informatics, cyber-physical-social systems, and cyber intelligence and security. Dr. Zhou is a member of the IEEE CS, and ACM, USA, IPSJ, and JSAI, Japan, and CCF, China.