Spatial-Temporal Graph Model Based on Attention Mechanism for Anomalous IoT Intrusion Detection

Xinlei Wang, Xiaojuan Wang, Mingshu He, *Member, IEEE*, Min Zhang, and Zikui Lu

***Abstract*—We propose an attention-weighted model for parallel extraction of spatial-temporal features to enhance the detection capabilities in the message queuing telemetry transport protocol, widely used in the Internet of Things. Our approach involves constructing perception node collection graphs based on packet header information, which capture transmission-dependent and context-sequence-dependent relationships in the data streams. We leverage a message-passing mechanism to aggregate adjacent nodes and update the weight matrix accordingly. Additionally, we employ a bidirectional long short-term memory model to capture long-distance dependencies in the sequence. The updated graph and the output of the time-series model are fused and processed by a self-attention mechanism, generating weights for classification. The classification results are obtained using a fully connected network. We evaluate our approach on four datasets (ToN-IoT, BoT-IoT, UNSW-NB15, and DoHBrw2020) and compare it against nine different algorithms. Experimental results demonstrate the effectiveness of our method, achieving high accuracy levels, such as 0.8874 on ToN-IoT, 0.9386 on BoT-IoT, 0.9390 on DoHBrw2020, and the best accuracy of 0.8659 on the unbalanced UNSW-NB15 dataset.**

***Index Terms*—Anomaly traffic detection, deep learning, Internet of Things (IoT), self-attention mechanism, spatial-temporal features.**

I. Introduction

THE Internet of Things (IoT) has extensively promoted the digital transformation of industry. However, industrial IoT (IIoT) intelligent devices connected to internal services with important information have become a viral target for hackers. Tampering with the transmitting commands and feedback can deviate the decision of intelligent devices leading to false alarms. Protecting data generated by IoT devices from the risk of tampering is critical. To this end, a payload encryption scheme based on message queuing telemetry transport (MQTT) protocol is proposed to liberate application messages from malicious listeners or untrusted MQTT clients. The application of this scenario is straightforwardly observed in Fig. 1, with the built-in core components including an MQTT broker and IoT devices covering subscribers and publishers.

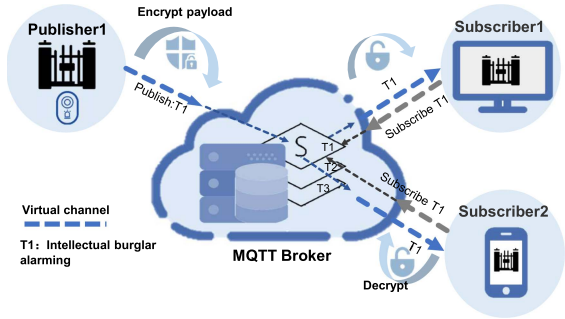

Fig. 1. MQTT broker in the cloud follows a publish–subscribe schema, collecting publishable data from IoT devices and enabling public users to access the data through subscription topics. After authentication, device publishers transmit encrypted payloads to the broker via the MQTT API. The broker cannot read the encrypted payload, and only trusted subscribers possess the decryption key.

By employing data encryption in IoT transmissions, the visibility of IoT devices to malicious stream attacks is effectively mitigated. Extracting essential features from encrypted data packet headers offers additional benefits, enabling potential real-time classification. However, the use of a few low-grade features poses challenges for accurate detection. To address this, we propose a graph-based representation called perception node collection graph (PNCG), which integrates neighbor nodes to obtain higher-level features and rich semantic information, incorporating features such as *packet size (PS)* and *interpacket time (IPT)*, and the direction of interaction between MQTT brokers and clients. This graph-based approach converts network flow classification into graph classification, allowing for enhanced detection capabilities with limited features.

Current deep learning (DL) models have been proposed to automatically learn essential features through multilayer hidden structures [1], [2]. However, directly applying these models to graph classification requires modifications for feasibility. Existing intrusion detection methods often rely on convolution or recursive models to extract single time/space features from flattened flow structures. In contrast, the graph structure can fully

Manuscript received 12 December 2022; revised 8 June 2023; accepted 9 August 2023. Date of publication 11 September 2023; date of current version 23 February 2024. This work was supported by the National Natural Science Foundation of China under Grant 62227805, Grant 62071056, and the project under Grant 30603020403. Paper no. TII-22-5057. *(Corresponding author: Xiaojuan Wang.)*

The authors are with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: ann96125@126.com; wj2718@bupt.edu.cn; hemingshu@bupt.edu.cn; mz2016bupt@163.com; ludeer97@outlook.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3308784.

Digital Object Identifier 10.1109/TII.2023.3308784

represent spatial features through node fusion and updating. Furthermore, contemporary methods tend to overlook imbalanced data and prioritize normal behavior profiles in classification, resulting in a high false positive rate for attacks with fewer instances [3].

More recently, the field of natural language processing began to transform words or sentences into graphs for classification. Gui et al.[15] transform sentences into directed graphs, treating each character as a node and connecting the first and last characters of the lexicon word. Huang et al. [5] utilize graph convolutional networks (GCN) to perform text classification by the local structure of the text and the node dependencies defined on the graph. Therefore, we analyze the correlation between network flow and text to find more graph-building and classification solutions. Network flows consist of packets of different sizes, and packets consist of the number of bytes. This structure is similar to the "letter-word-sentence" hierarchy that divides language, which inspired us to construct a PNCG representation of a single stream.

In PNCG, inspired by the concept of treating words as nodes in [5], we employ PS and IPT for different nodes. It is necessary to leverage time series analysis to identify anomalies in the IoT, as the data generated by smart devices commonly exhibit distinct sequential and temporal characteristics. On this basis, an attention-weighted model for parallel extraction of spatial-temporal features (APS-T) is proposed. In particular, the original input features are assimilated into the bidirectional long short-term memory (Bi-LSTM) network to mine the time-based context sequential flow features. After incorporating space dimension information into the output at the time-reconstruction level of Bi-LSTM, the model automatically assigns attention weights to the fused features. PNCG is constructed using only time-dependent package features to accelerate feature extraction, which sacrifices precision. To address the need for improving classification accuracy, it is necessary to reserve an expandable entry in PNCG. Furthermore, adding more features does not increase the complexity of building the PNCG. The main contributions of this thesis are as follows.

1) To compensate for the deficiency of fewer features and weak representation of encrypted data in security systems, we propose a new attention-based mechanism named APS-T. The model can extract the spatial-temporal stream features in parallel, integrating graph, and time series benefits to address the changeable threats in IoT.
2) The PNCG was developed for capturing rich traffic packet transmission information, prioritizing the construction of the model based on time and spatial features. It also provide distinctive construction strategies for time overhead and performance guarantee. In pursuit of test accuracy, the PNCG extensible entry is developed without additional construction complexity.
3) The proposed model applies to at least four IoT-related scenarios. Compared with the compelling method, the APS-T model makes up for the problem of achieving a high detection rate with a small number of features and provides an attempt for trending real-time malicious network flow detection.

TABLE I
COMPARISON OF THE ADVANTAGES (PROS.) AND DISADVANTAGES (CONS.) OF EACH ANOMALY DETECTION METHOD (MTD.)

Mtd.	Pros.	Cons.
[28]	High performance achieved with limited training data.	Further improvement required in practical applications.
[29]	Applying data stream Processing to packets analysis.	Lack of integration with standard controllers.
[38]	LSTM-enhanced packet sequence state.	Time series-reduced attacks can impair detection.
[30]	Solely based on statistical and sequential features.	Time-consuming feature selection process.
[36]	Extracting time series features from encrypted flows.	Time-consuming traffic fingerprint extraction.

The rest of this article is organized as follows. Section II outlines the distinctions between the proposed model and other related works. Sections III and IV introduce the PNCG and APS-T models, respectively. Section V presents the analysis and results of the experiments conducted. Finally, Section VI concludes this article.

II. RELATED WORK

A. Encrypted Traffic Analysis

The encryption-based lightweight MQTT protocol is highly suitable for securing IoT systems targeting low-power and low-computation embedded devices in our research. Encrypted data helps prevent risks such as tampering and eavesdropping, but it does not provide absolute immunity against malicious network flow intrusion. Therefore, this subsection examines malicious detection methods for encrypted flows, including traffic decryption and traffic analysis. After stream decryption, general intrusion detection methods can be applied. However, the network configuration range of stream decryption is limited, and the destruction of end-to-end transmission provides a hotbed of attack for malicious opponents [27]. The advantages and limitations of methods for traffic anomaly detection are summarized in Table I.

The traffic analysis approach, in contrast, bypasses decryption and instead centers on feature extraction and the application of detection methods. Effective feature extraction granularity includes packet and flow level. The study [28] focuses on extracting network flow-level features and conducting feature selection to eliminate those with weaker classification capabilities. Although these methods perform well, background flow fluctuations may affect the detection accuracy and have an exhausting feature selection process. In order to mitigate the consequences of network intrusions, such as business disruptions and prolonged outages, it is crucial to maintain a continuous monitoring system and swiftly respond to security changes by extracting real-time insights from analytical data. Therefore, it is crucial to continuously monitor network flow fluctuations by analyzing a minimal set of packet-level features [29].

Zhao et al. [38] used data packet sequences and a time series model to simulate common network phenomena, such as packet loss, by representing four different data packet actions. Bayat et al. [30] employed features such as packet size, interarrival

TABLE II
COMPARISON OF THE ADVANTAGES (PROS.) AND DISADVANTAGES (CONS.) OF EACH GRAPH-BASED CLASSIFICATION METHOD (MTD.)

Mtd.	Pros.	Cons.
[23]	Representing individual nodes with multiple features enhances the richness of node information.	Excludes encrypted data from packet content analysis.
[33]	Capable of classifying traffic of "heavy-traffic flows with long message sequences" type.	① Single temporal feature
[21]	Sharing node features and adaptability to imbalanced datasets.	② Lacking Dynamic Graph
[22]	Constructing a graph network by extracting readily available features such as duration and packet size.	③Local perspective ④Insufficient node-level information

①: Focusing solely on time-based message sequence features may overlook other important traffic characteristics, such as spatial features.
②: After the construction phase, the graph cannot accommodate the addition of new node information.
③: Node classification tasks often focus on the local neighborhood and subgraph information of nodes, which may limit the understanding and utilization of global graph properties and structure.
④: One potential drawback of using nodes as IP address representations is that it may result in a loss of information or limited representation power.

intervals, and payload size extracted from the hypertext transfer protocol secure (HTTPS) for classification. However, selecting features for different protocols can be a time-consuming process. We realize that encrypted flows possess significant time series features [36], enabling rapid response through the utilization of relative arrival times combined with message sizes. CNN, typically employed for extracting spatial features [34], often struggle to achieve high-precision classification due to their reliance on samples with limited features.

Graph structures have the ability to extract richer features, with each subgraph allowing for the allocation of globally shared attributes among nodes [35]. It becomes easier to incorporate other general information, such as average packet pair lengths, into the graph. These intuitions encourage us to extend network traffic detection to DL models that accept graph feature extraction.

B. Spatial Graph Construction

Graph representation enables the construction of dynamic interactive topologies and the extraction of multimodal features for unstructured real-world data [21]. Significant progress has been made in the construction of network topology [23], [32]. For example, in traffic prediction, [23] considers the correlation between traffic byte states and utilizes the contextual semantics of time series nodes in graph neural network (GNN). Chen et al. [33] focus on the negotiation process between sender and receiver, extracting message size as node features. Zhang et al. [21] address graph node classification using multiweighted cosine similarity and network reconstruction technology combined with GNN joint learning. Chang et al. [22] construct the graph network by extracting duration, transaction bytes, and transmitted data packet size. For further detailed summary of these methods, please refer to Table II.

Graph-based approaches exploit diverse features beyond encrypted streams, surpassing the scope of our research. Additionally, the above method overlooks nonadjacent nodes' potential interactions within the given graph structure space. Thus, relying solely on spatial feature extraction methods that consider local relationships may hinder effective long-term dependence learning.

C. Differences

The future trend lies in the realization of real-time and fast-response network flow anomaly detection. The aim is to avoid the cumbersome process of feature selection and extract more valid potential representations from non-Euclidean spatial data using as few features as possible. Effective features at the packet level enable real-time or near real-time tracking of performance degradation or security threats. Moreover, it is crucial to propose a breakthrough from the conventional spatial relationship extraction methods and develop an effective approach to capture comprehensive spatial-temporal correlation features. This approach allows for the extraction of long-term dependencies within network flows.

III. CONSTRUCTION OF PNCG

This section focuses on representing IoT streams using PNCG. It covers the selected features for PNCG construction, the detailed process of constructing PNCG based on the MQTT protocol, and the node update strategy that enhances the classification model's input.

A. Graph Features Selection

The data generator shown in Fig. 2 was customized to generalize the complete traffic data transformation chain. We parse packet capture (PCAP) files containing abundant and disordered packet sequences composed of different traffic flows, which are split into separate streams based on the five-tuple. Taking encrypted streams into consideration, we collect only the packet header information irrelevant to the actual content. Han et al. [4] visualized the packet time interval of various attack measures, proving that the timestamp feature helps distinguish different attacks. In addition, the length of each packet is a commonly-used and compelling feature of traffic classification. Therefore, we decided to extract only two features, PS and IPT, to generate the nodes' representation.

In addition to the explicit features mentioned above, we added some implicit features to build connections between nodes in PNCG. Fig. 3 shows the packet transmission process in the publish-subscribe channel based on the MQTT protocol. In the case of a publish message, a set of packets transmitted in the same direction (publisher to broker) is defined as an implicit feature called a *Segment*. Packet messages published to the broker are called *message segments (MS)*, whereas packets received by the publisher back from the broker are called *Ack Segments (AS)*. If there are packets in transit, taking the MS transmission direction as the initial direction, then the direction of AS is defined as negative, and IPT is always positive. On

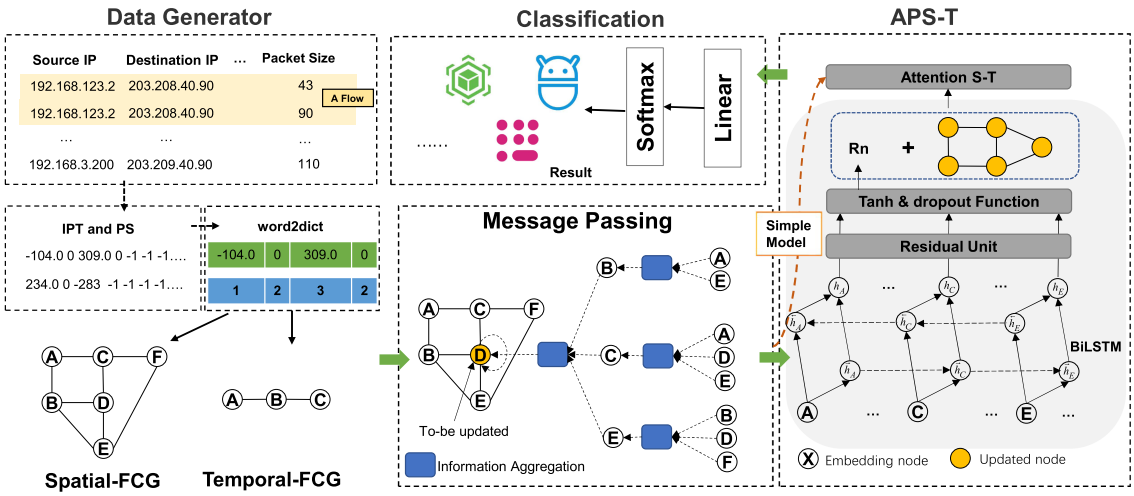

Fig. 2. APS-T model building process and PNCG preconstruction work.

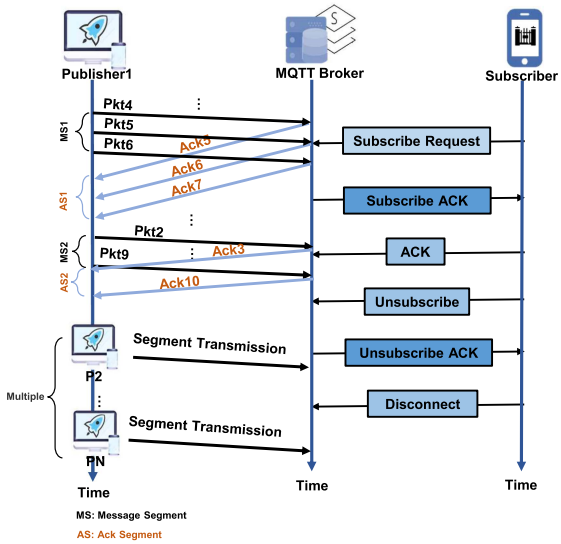

Fig. 3. Example of a publish–subscribe rule based on MQTT. The implicit features are defined in packet transfer between the publisher and the MQTT broker.

the contrary, if no packet is transmitted, the PS and IPT are both -1. After processing, the flow samples are input into word embedding and fed into the model afterward with graph construction.

B. PNCG

PNCG includes two graph structures: communication-dependent *spatial-PNCG* and context-sequence-dependent *temporal-PNCG*. We first give the formal definition of PNCG before introducing the design process of temporal-PNCG and spatial-PNCG with concrete examples, respectively.

Here, we take a labelled graph with some nodes and edges, which can be formally defined as $PNCG = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ represents a set of vertices consisting of n nodes $\{v_1, ..., v_n\}$, where $\mathbf{A} \in \mathbf{R}^{n \times m}$ is a matrix that contains a_{ij} denotes the jth adjacent node of the ith node. m is the total number of neighbors of node v_i. We also defined *hop distance (h)*, which means that the number of neighbors of a node in a specific direction cannot exceed the maximum value h. Each node v_i and adjacent node-set a_i of v_i in the PNCG is encoded as the corresponding embedding vector. The adjacent node set of each node in the PNCG is expressed as *neighbor routing table (NRT)*, and its length is $2h$ in temporal-PNCG or $3h$ in spatial-PNCG. Generally, a specific PNCG structure corresponds to a complete session connection consisting of several (PS, IPT) tuples. Each tuple element is called an *S-T Feature*.

1) Temporal-PNCG: In the existing flow representation method [6], flow F is expressed as $F = (p_1, ..., p_N)$, where p_i represents the size of packets loaded by streams that are exchanged from server and client. Inspired by this, we proposed adopting temporal-PNCG, a graph construction method based on packet sequence, as shown in Fig. 4(a). The difference is that the nodes of temporal-PNCG contain both IPT and ST. The neighbors of each node are defined as an adjacent node with a fixed hop distance. If there are not enough adjacent nodes within the hop distance, it is padding with 0. In Fig. 4(a), node 2 is used as an example, and $h = 3$. We specify that the NRT of node 2 is distributed in the left and right directions. Since $h = 3$, the NRT is represented as [0, 0, node 1, node 3, node 4, node 5]. From this example, each node has an NRT length of six. It is worth noting that only explicit features are used in temporal-PNCG, and natural adjacency relations replace the connection between nodes. Algorithm 1 lists the neighbors of each node, where h can be any value that does not exceed the length of the input packet sequence.

2) Spatial-PNCG: To completely represent the interaction traffic behavior of publisher and broker, we use a graph structure with sufficient information to represent network flows, which is spatial-PNCG. As shown in Fig. 4(b), similar to temporal-PNCG, each node represents IPT or PS in AS or MS. We set $h = 1$ for simplicity, and in this way, the NRT of each node only includes the neighbors directly connected to it. Since the NRT length of each node is controlled to three, the length

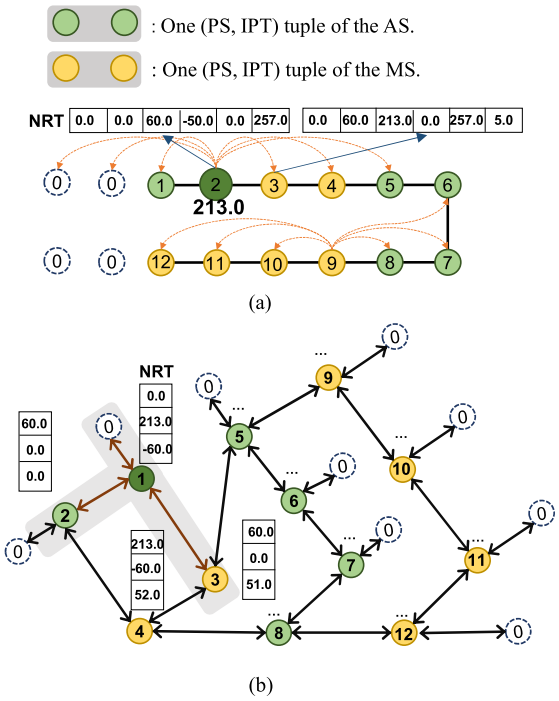

Fig. 4. Two forms of PNCG are shown. Each node in the PNCG has a specific IPT or PS value. For example, node 2 in (a) corresponds to IPT = 213.0 in AS1. (a) An example of the composition of Temporal-PNCG. (b) An example of the composition of Spatial-PNCG.

Algorithm 1: NRT Construction for Each Node in Temporal-PNCG.

Input: $\mathcal{V} = \{v_1, ..., v_n\}$: a vertex sequence of the S-P features in a flow; Placeholder $\mathcal{P}$: an all-zero array of the shape v_i.
Output: A matrix of neighbors of each node in $\mathcal{V}$
$E_{nbs} \leftarrow$ Empty array that will contain all node's neighbors
$e_i \leftarrow$ Empty array contains neighbor nodes of length $2h$ for v_i
$E_{nbs} = []$;
$n = \text{len}(\mathcal{V})$;
For $s_{idx} \leftarrow 1$ **to** n **do**
  $e_i = []$; #An all-zero array of the shape $(1, 2h)$;
  For $h_{idx} \leftarrow 1$ **to** h **do**
    $bf_{idx} = s_{idx}$ - 1 - h_{idx};
    e_i.append ($\mathcal{V}[bf_{idx}]$ if bf_{idx} exist else $\mathcal{P}$);
  For $h_{idx} \leftarrow 1$ **to** h **do**
    $af_{idx} = s_{idx}$ + 1 + h_{idx};
    e_i.append ($\mathcal{V}[af_{idx}]$ if af_{idx} exist else $\mathcal{P}$);
  E_{nbs}.append (e_i)
Return E_{nbs}

that is not satisfied is supplemented by a zero or zero serial (specifically, each node is represented by a feature vector). In nodes of different segments, only the first and the last nodes need to be connected to other segments, which follow the principle of "first to first and last to last."

To be more specific, in Fig. 4(b), (node 1, node 2) and (node 3, node 4) represent the (PS, IPT) tuples of MS1 and AS1, respectively. Since AS1 is opposite to MS1, which defaulted to the initial direction, PS in AS1 is defined as negative, but IPT is always positive. Then according to the internal order of MS1 and AS1: node 1 is connected to node 2; node 3 is connected to node 4. The connection relationship between different segments (AS1 and MS1) is established as follows: the first node (node 1) and the last node (node 2) of MS1 are connected to the first node (node 3) and the last node (node 4) of AS1, respectively. Finally, the NRT of the node is confirmed. Taking node 1 as an example, it is composed of its neighbor nodes whose $h = 1$, which is (0, node 2, node 3). Algorithm 2 lists the construction process of NRT for each node, in which h is fixed at 1.

Algorithm 2: NRT Construction for Each Node in Spatial-PNCG.

Input: $\mathcal{V} = \{v_1, \ldots, v_n\}$: a vertex sequence of the S-P features in a flow; Divide $\mathcal{V}$ into groups $\mathcal{S} = \{s_1, ..., s_m\}$ according to segments; Placeholder $\mathcal{P}$: an all-zero array of the shape v_i.
Output: A matrix of neighbors of each node in $\mathcal{V}$
$E_{nbs} \leftarrow$ Empty array that will contain all node's neighbors
$e_i \leftarrow$ Empty array contains neighbor nodes of length $3h$ (h=1) for v_i
$E_{nbs} = \emptyset$;
For s_i in $\mathcal{S}$ **do**
  $e_i = \emptyset$; #An all-zero array of (e_{i1}, e_{i2}, e_{i3})
  For v_i in s_i **do**
    If v_i.index == 0 **then**
      $e_{i1}, e_{i2}, e_{i3} \leftarrow s_{i-1}[0], s_i[1], s_{i+1}[0]$ if exist else $\mathcal{P}$;
    Elif v_i.index == len$(s_i) - 1$ **then**
      $e_{i1}, e_{i2}, e_{i3} \leftarrow s_{i-1}[-1], s_i[-2], s_{i+1}[-1]$ if exist else $\mathcal{P}$;
    else
      $e_{i1}, e_{i2}, e_{i3} \leftarrow s_i[v_i.\text{index}-1], \mathcal{P}, s_i[v_i.\text{index}+1]$;
    E_{nbs}.append (e_{i1}, e_{i2}, e_{i3});
Return E_{nbs}

3) Extensible PNCG: To enhance detection performance and generalize PNCG to a more general form, we propose an extensible PNCG that can accommodate additional features. The PNCG proposed in this article is essentially a *Tiled* form, and each node represents a feature. In contrast, the extensible PNCG is represented as a *Grouping* form in which a feature vector can represent each node. The advantage of grouping is that a single node can be expanded with more features to improve the classification performance. Furthermore, regardless of the capacity of the feature vector, the node count overhead of PNCG remains unaffected.

The segmentation of nodes based on message transmission direction is the sole factor determining node connectivity in extensible PNCGs. On the premise that all features within a packet share the same orientation, placing feature groups on the

same or different nodes does not interfere with graph connectivity. Experimental evidence supports this claim. In the APS-T model, aggregating neighboring nodes is crucial for feature extraction in the PNCG. The tiling form aggregates features within the same packet, while the grouping form directly aggregates features from other packets. We conducted experiments on the CICDDOS2019 dataset[1] using both tiled and grouping methods, extracting only PS and IPT features. The classification accuracy remained the same for most classes. This analysis and experimentation demonstrate that PNCG can be expanded without increasing node count cost.

C. Message Passing Neural Network (MPNN)

The construction of PNCG aims to extract hidden features and perform graph classification. To achieve this, we map the graph's topological information to a unified vector space. The Weisfeiler–Leman (WL) model [16] is recognized as the most advanced and standard measurement for graph comparison. It effectively maps feature and structural information to unified embeddings, allowing for the identification of isomorphism and categorization of nonisomorphic graphs. We implement the WL theoretical model by iteratively aggregating the immediate neighborhood expressions and updating the current node through message passing. The aggregation method is constrained by a permutation invariant function such as summing, averaging, or maximizing [17].

The MPNN method proposed by [5] is used to complete the process, whose forward propagation includes neighbor node aggregation and node message update. 1) To get the representation m_v^{t+1} of the neighbor node, allowing to define two variables, including the weight $e_{v\omega}$ of the edge between node v and its neighbor ω and the hidden states h_v^t of node v. Aggregate function $M_t(h_v^t, e_{v\omega})$ is used to obtain m_v^{t+1}. 2) $U_t(h_v^t, m_v^{t+1}, \eta_v)$ is used to aggregate the information of h_v^t and m_v^{t+1} to obtain the updated information of node v. η_v^t represents the weight of the node v, which can be constantly updated in the training process. The specific calculation method of M_t and U_t is as follows:

$$M_t(h_v^t, e_{v\omega}) = \max_{\omega \in N_v^m} h_v^t e_{v\omega},$$
$$U_t(h_v^t, m_v^{t+1}, \eta_v) = (1-\eta_v)m_v^{t+1} + \eta_v h_v^t \quad (1)$$

Due to the aggregation of node information, higher order features are obtained by each node in PNCG. The representation of nodes and the weight of edges are globally shared in information transmission, allowing each node update to have access to global weight information during training. In PNCG, edges also carry special meanings, including the delivery direction of packets and continuous transmission (CT). Edge features ($e_{v\omega}$) are considered in the node aggregation process of MPNN. While MPNN represents the spatial method in PNCG, the implementation of WL also includes spectral graph convolutions, which are effective for node classification tasks [21]. However, models based on spectral graph convolutions, such as GCN [18] and GraphSAGE [19], do not consider feature information on edges between nodes. Additionally, single-layer GCN with immediate neighborhood aggregation ignores long-range dependencies in the graph. Increasing the number of layers and accelerating the expansion of the message passing algorithm can lead to the homogenization of inference [20].

At this point, we have completed the updating of PNCG nodes. In Section IV, advanced features will be further extracted based on the updated node information to achieve better-inferred representation.

IV. Proposed APS-T

A. APS-T Overview

The APS-T training process, as shown in Fig. 2, involves several steps. First, a standard data generator is used to parse the packet sequence into network flows with uniform formats. These flows are then transformed into PNCG representations, as described in Section III. APS-T utilizes temporal-PNCG or spatial-PNCG as input and learns two classifiers: the simple model and the APS-T model. The APS-T model employs Bi-LSTM with residual units and attention mechanisms. In contrast, the input data of the simple model directly enters the attention S-T module without passing through other layers. During training, a validation dataset is provided every 100 batches to prevent overfitting. In the final test phase, the well-trained model assigns unknown flows (as PNCG) to predefined classes.

The simple model, a component of APS-T, is primarily responsible for parameter finalization in the early stage of PNCG. The overall working mechanism of APS-T is as follows: spatial features are extracted from the input PNCG using MPNN, while time-series features are automatically extracted by Bi-LSTM with residual units in parallel. The outputs from these modules are fused into a matrix. Self-attention mechanisms are then applied to assign weights to the fused feature matrices. Finally, the softmax layer utilizes the weighted feature representations to identify malicious traffic labels. In this section, we provide a brief introduction to the main modules of the APS-T model and analyze the effect of each module on feature information mining.

B. Bi-LSTM Residual

Bi-LSTM extends LSTM by incorporating a hidden layer connection in reverse time order, allowing it to capture more comprehensive context information from the packet sequence. This is achieved by concatenating the forward and reverse knowledge at each step.

To facilitate effective and efficient training, a residual connection is introduced in the Bi-LSTM unit. This enables the gradient to propagate directly through the short-range residual connection, leading to improved generalization on unbalanced datasets. In this study, we utilized an enhanced residual unit [7] that satisfies the following relationships:

$$X_L = X_l + \sum_{i=1}^{L-1} \mathcal{F}(X_i, \mathcal{W}_i) \quad (2)$$

[1]The reference to this dataset is only for a test of the extensible PNCG, which is detailed in https://www.unb.ca/cic/datasets/ddos-2019.html.

where X_i represents the input feature of the ith residual unit, $\mathcal{F}$ is the residual function. It is contingent on the combination of Bi-LSTM and the residual unit, making the parameter gradient can be passed unimpeded to any preceding layers via shortcut connections.

C. Attention S-T

The attention mechanism allows the network to focus on the crucial parts of the information relevant to the current task, while disregarding noncritical parts. To capture the weighted features in the fused S-T series, we incorporated a self-attention mechanism. Input the original $[(PS, IPT)_1, ..., (PS, IPT)_H]$ sequences of all streams into Bi-LSTM with the residual unit to get the output matrix B, which is composed of $[b_1, b_2, ..., b_H]$, and H is the number of features in the network flow. The representation R_n of the fusion matrix fed to the attention mechanism consists of these output vectors and the embedding of the update node $\hat{R}_V$

$$\begin{aligned} O &= tanh(B) \\ O &= dropout(O) \\ R_n &= O + \hat{R}_V \end{aligned} \tag{3}$$

where $B \in \mathbb{R}^{H \times d}$, d is the dimension of the embedding of each network flow feature vector. R_n, O, $\hat{R}_V$ all have dimensions of $[batch_{size}, H, d]$. We obtain the fusion matrix R_n and adopt a multihead self-attention mechanism to calculate the correlation value between two S-T vectors.

D. Analysis of APS-T

The Bi-LSTM residuals module has several notable qualities. It is constructed by stacking two hybrid layers, which include a batch normalization (BN) layer, an activation layer, and a 1-D convolution (Conv-1D). The integration of residual units in the Bi-LSTM module provides the following advantages. 1) The BN layer before rectified linear unit (ReLU) acts as a regularization mechanism to reduce overfitting. 2) The residual-based network is easier to optimize and avoids the gradient vanishing problem, even in mini-batches. 3) The total output of the residual network combines the outputs from the Bi-LSTM unit and the Conv-1D layer. The residual unit learns the difference between the input and output, enhancing the network's capability. 4) Adding Conv-1D to each residual unit allows for capturing local features in a refined manner, complementing the ability of Bi-LSTM to capture long-term dependencies. 5) Experimental results, including those in [22], demonstrate that the residual module improves classification accuracy, especially for unbalanced datasets.

The motivation for introducing attention mechanisms in the APS-T model is rooted in node aggregation. Inspired by the concept presented in [5], where each node in the graph aggregates adjacent features, the APS-T model employs multihead self-attention mechanisms. These mechanisms automatically select discriminative information from the sequence features generated by Bi-LSTM, Conv-1D, and the graph structure after neighbor aggregation, tailored to specific downstream classification tasks. In our approach, we follow the method described in [37] to incorporate temporal and spatial features by dimension. The attention mechanism captures global dependencies between the added features, optimizing the weight coefficients for node and time features in the target network flow. The aggregate information, denoted as R_n in (3), is input into the attention mechanism. To prevent sequential information loss, we incorporate a positional code consisting of two fully connected layers, which calculates the correlation of the embedded representation. The positional code injects relative position information into the weighted S-T vector as a residual, compensating for the loss of relative position information during message transmission.

V. Performance Evaluation

A. Preparatory Work

1) Competitors of Evaluation: To effectively assess the performance of APS-T, we supplemented nine typical models for comparison. Five of these models, GID [21], E-ResGAT [22], CAN-GAT [23], EnsembleRF [30], and ERNN [38], have been introduced in related work, and only the remaining four are briefly introduced here.

1) TextGCN, which is a typical model in the field of text classification. Each sample builds a graph separately, and the weights of nodes and edges are globally shared [5].
2) Transformer, which includes encoder and decoder are composed of scaled dot-product attention and multihead attention [8].
3) PointNet, which enriches the representation capability of point cloud structure by capturing spatial semantic features across long distances on original embedding [9].
4) Deep pyramid convolutional neural networks (DPCNN), which extracts long-distance text dependencies by deepening the network [10].

2) Scenarios of Evaluation: In order to verify the adaptability of the APS-T to different scenarios, we use four datasets (BoT-IoT [11], UNSW-NB15 [12], ToN-IoT [13], and DoHBrw2020 [14]) to simulate botnet attack, vulnerability attack, real-world IoT applications, and domain name system (DNS) attack, respectively. The following describes the four attack scenarios associated with the datasets.

1) *Botnet attack:* Botnets in the IoT have the ability to impact the physical environment and execute a range of attacks through Command and Control infrastructure. These attacks include distributed denial of service (DDoS), Keylogging, and the propagation of other botnet viruses. The BoT-IoT dataset serves as a simulation of diverse botnet attacks, offering representative data from IoT sensors.
2) *Vulnerability attack:* The exposure of IoT and cloud devices directly to the Internet increases the risk of data breaches through device vulnerabilities. The UNSW-NB15 dataset includes nine attack categories: Fuzzers, Analysis, Backdoors, and others. Among these categories, there are 6845 subtypes, with 3068 attacks related to vulnerabilities listed in the Common Vulnerabilities and Exposures library.

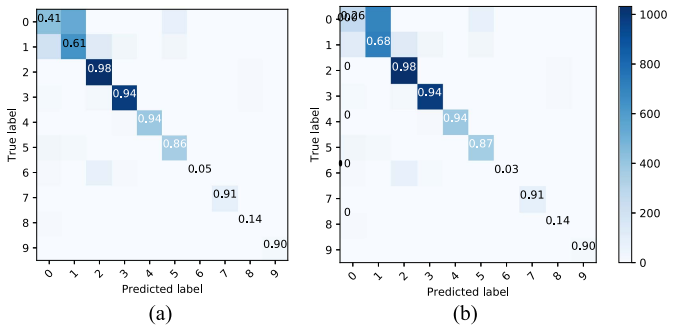

Fig. 5. Influence of different representation methods of nodes on classification accuracy of each class. (a) Each node is represented by a feature called the tiled form. (b) Each node is represented by a vector of features called the clustering form.

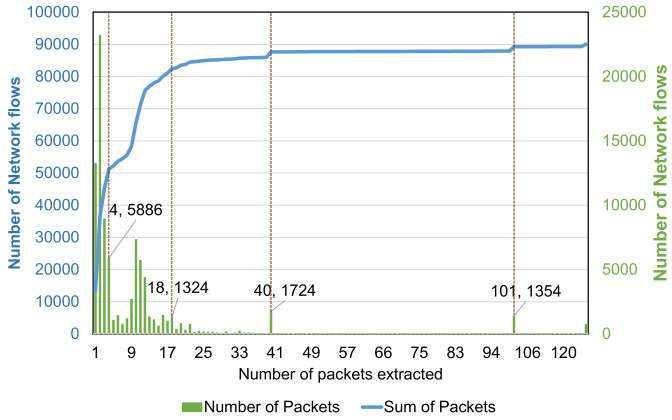

Fig. 6. Count the number of packets in each stream.

3) *Real-world IoT applications:* The above datasets are collected from simulated IoT environments, which may not accurately reflect the characteristics of malicious behavior in the IoT. Therefore, we utilized the ToN-IoT dataset, which generated nine real network security attacks, including cross site scripting (XSS), password cracking attacks, and man-in-the-middleattack (MITM), through the operation of IoT/IIoT sensors.
4) *DNS attack:* To broaden the application of APS-T to different scenarios, we applied it to detect traffic in common Internet applications. We assessed its performance using the DoHBrw2020 dataset, which encompasses a variety of benign and malicious DNS activity from a variety of websites, covering both DoH and non-DOH traffic captured from popular web browsers.

Before conducting the experiments, two important points need to be clarified. The real-world ToN-IoT dataset is used for evaluating all trials, except for the second part of subsection D. Furthermore, prior to subsection C, the performance evaluation solely relies on the accuracy (AC) metric, which measures the ratio of correctly predicted samples to the total sample size.

B. Parameter Tuning

1) Packet Number: The number of packets extracted from each flow will directly affect the performance and training time of the classifier, where the training duration involves two major stages: the construction time T_{cg} of PNCG and the actual training time T_{at} of the classifier. According to the analysis of [24], the maximum number of packets in each flow does not exceed 128. Therefore, we limit the extraction of packets per stream to be within 128. As illustrated in Fig. 6, the number of packets in the flow is mainly concentrated between 1–40. In addition, four inflection points appear at the first 4, 18, 40, and 101 packets across the total number of packets curve, which serve as additional monitoring points for further investigation.

The selected monitoring points for the number of packets are presented in Table III. We roughly used a simple classifier to obtain results, which offers the advantage of efficient execution and reflects the optimal number of packets to be extracted. By extracting only the first 40 packets from each stream, the test accuracy reached 0.8753. However, as more packets were included in constructing PNCG, both T_{cg} and T_{at} showed varying trends of increase while the accuracy fluctuated. This can be attributed to the classifier requiring additional time to learn the increasingly complex PNCG, and the presence of excessive redundant empty data in each flow when the number of packets exceeded 40, which interfered with the classifier. Ultimately, we opted to extract 40 packets from each flow to strike a balance between training accuracy and time cost.

TABLE III
TIME CONSUMPTION AND ACCURACY EVALUATION FOR DIFFERENT NUMBERS OF PACKETS

# Packets	4	18	40	60	101	128
AC	0.8612	0.8735	0.8753	0.8692	0.8725	0.8743
T_{cg}(s)	9	22	44	61	99	118
Tat(s)	243	285	328	452	663	1102

TABLE IV
TRAINING TIME CONSUMPTION AND ACCURACY EVALUATION FOR DIFFERENT NUMBERS OF EPOCHS

Epochs	1	2	5	8	10	14
AC	0.8704	0.8736	0.8758	0.8783	0.8786	0.8784
Tat(s)	74	148	362	564	702	972

2) Time-Consuming of Training Epochs: Table IV shows the training time of the simple model and the test accuracy corresponding to different epochs. After the first epoch, the accuracy of the model reaches about 0.87. The accuracy of the converged model fluctuates within a small range as the number of epochs increases. Although there is only a slight improvement in accuracy after the eighth epoch, training time nearly doubles when it reaches 14 epochs. Therefore, training is concluded at eight epochs to ensure prompt completion.

C. PNCG and APS-T Model Construction

1) PNCG: In Section III, we designed two graphs to extract S-T information. The experiments aimed to assess the sensitivity

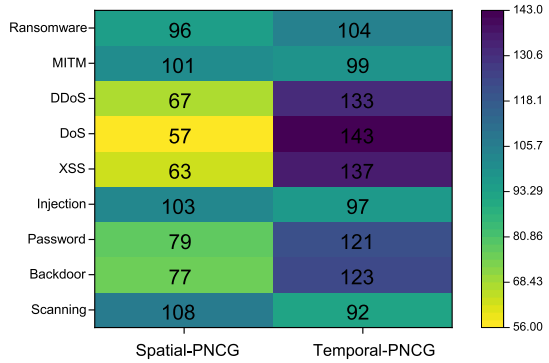

Fig. 7. Influence of two kinds of graph construction methods on the classification accuracy of different abnormal behaviors.

TABLE V
TIME CONSUMPTION OF THE TWO GRAPH CONSTRUCTION METHODS

	Spatial-PNCG	Temporal-PNCG
T_{cg}(s)	40	34
Tat(s)	463	609
Total(s)	503	643

of network flow abnormal behavior to these features. We trained the simple model with two input graphs for 200 cycles to improve efficiency. Additionally, we compared the accuracy of the same class in both graphs and added 1 to the corresponding class with higher scores. The thermal distribution diagram in Fig. 7 shows the cumulative scores of each type of malicious traffic under the S-T graphs.

DDoS, DoS, XSS, Password, Backdoor, and Ransomware exhibit varying degrees of association with temporal-PNCG features. DDoS and DoS attacks exploit vulnerabilities to cause network congestion or system resource exhaustion, leading to performance degradation. These attacks typically require a time-biased approach due to the accumulation of access over a period. Backdoor attacks involve the implantation of a malicious program that requires human intervention, making them less related to the natural transmission process of network flow. This suggests that certain attacks rely more on contextual information, and continuous sequence features aid in their expression. Conversely, injection and scanning attacks exhibit a slight inclination towards spatial-PNCG construction. Scanning attacks leverage collected information to launch complex attacks by scanning devices within the network system, indicating that the threat characteristics are less sensitive to time series.

Finally, we compare the construction and training time of the two graph structures, as shown in Table V. Despite the simple construction of temporal-PNCG, classifiers require more time to learn their representation during training.

2) APS-T Model: To assess the impact of different modules on anomaly detection performance, we conducted a series of ablation experiments to evaluate the overall improvement brought by each module. Leveraging the temporal-PNCG's remarkable ability to represent context information, we directly input the data into a Bi-LSTM to extract more expressive temporal features. The results are presented in Table VI.

TABLE VI
ABLATION EXPERIMENT OF S-T MODEL

Single Head	Multiple head	Bi-LSTM	Residual	AC
✓				0.8717
	✓			0.8785
	✓	✓		0.8821
	✓	✓	✓	0.8857

In our experiments, we incorporated a single-head attention mechanism, multihead attention mechanism, Bi-LSTM, and residual unit. The findings indicate that the addition of Bi-LSTM leads to a test accuracy of 0.8821, providing evidence for the significance of temporal dependencies within the input sequence. Furthermore, with the inclusion of residual units, the test accuracy is further enhanced to 0.8857, augmenting the model's expressive capability.

3) Comparative Experiment: We conducted four comparative experiments using a simple model and the APS-T model to demonstrate the effectiveness of integrating spatial-PNCG in improving the classification accuracy of abnormal behaviors inclined to the temporal dimension. The summarized results can be found in Table VII.

The temporal tendency of abnormal behaviors is marked with "*" in the table, and the classification accuracy is mainly compared. We use the proposed model to process spatial-PNCG and temporal-PNCG, respectively. Compared with the simple model, spatial-PNCG is represented by the APS-T model, which makes Password, DoS, DDoS, and Ransomware enjoy better performance in terms of accuracy. Also, applying the APS-T model on temporal-PNCG can significantly boost the predicted outcome of DDoS and Ransomware by 1.37% and 3.64%. Naturally, the final classification accuracy of the APS-T model is higher than that of the simple model. Therefore, for temporal-inclined malicious behaviors, the added modules can effectively capture the features in the original sequence and simultaneously fuse the spatial features of the network flow to improve the classification performance.

D. External Evaluations

This subsection fell into two different emphases. One is a detailed comparison between the APS-T model and other competitors in a single scenario, including training time (TT) among the metrics measured. Another is scenario assessment, a comprehensive comparative analysis of three different definitive datasets.

We will assess the efficiency and effectiveness of all classification models, considering the introduction of new metrics: precision (PR), recall (RC), and F1 score (F1). PR represents the ratio of correctly extracted labels to the total number of extracted labels. RC is the ratio of correctly predicted samples to the total sample size. RC and PR are complementary indicators, and the

TABLE VII
HYPERPARAMETER OPTIMIZATION AND RESULTS

		Scanning	Backdoor*	Password*	Injection	XSS*	DoS*	DDoS*	MITM	Ransomware*	AC
Simple	Spatial-PNCG	0.9478	0.8050	0.9192	0.7149	0.9429	0.9516	0.9179	0.9347	0.7827	0.8785
model	Temporal-PNCG	0.9389	0.8060	0.9281	0.7103	0.9396	0.9583	0.9178	0.9409	0.7712	0.8783
APS-T	Spatial-PNCG	0.9389	0.8037	0.9281	0.7227	0.9396	0.9535	0.9297	0.9428	0.7915	0.8831
model	Temporal-PNCG	0.9426	0.7952	0.9293	0.7172	0.9455	0.9528	0.9315	0.9284	0.8076	0.8828

TABLE VIII
COMPARISON BETWEEN NINE COMPETING MODELS AND APS-T MODEL WITH SPATIAL-PNCG IN AC, PR, RC, WF1 PERFORMANCE EVALUATION METIRCS, AND TRAINING TIME

Method	AC	PR	RC	WF1	TT(s)
GID [21]	0.1603	0.0589	0.1611	0.0731	–
E-ResGAT [22]	0.2170	0.2692	0.2170	0.1537	654
CAN-GAT [23]	0.1644	0.1594	0.1644	0.1362	375
EnsembleRF [30]	0.6202	0.7300	0.6200	0.6400	1646
ERNN [38]	0.2133	0.5985	0.2133	0.3145	654
TextGCN [5]	0.5614	0.4554	0.5608	0.4876	442
PointNet [9]	0.8680	0.8720	0.8680	0.8691	586
Transformer [8]	0.8711	0.8784	0.8711	0.8736	780
DPCNN [10]	0.8790	0.8817	0.8789	0.8789	25993
APS-T Model	**0.8874**	**0.8914**	**0.8871**	**0.8876**	**963**

The bold entities indicate the classification performance and training time that can be achieved by the proposed APS-T model.

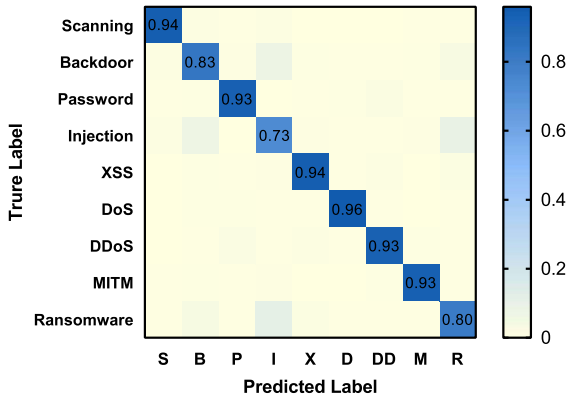

Fig. 8. WF1 confusion matrix for nine kinds of malicious behavior of network flows based on the ToN-IoT dataset.

weighted F1 (WF1) serves as a composite metric combining their scores, calculated as $F1 = (2 \times PR \times RC)/(PR + RC)$. In this study, WF1 is employed to evaluate the balanced dataset using the following formula: $WF1 = 2 \times (\mathcal{W} - PR) \times (\mathcal{W} - RC)/(\mathcal{W} - PR + \mathcal{W} - RC)$, where $\mathcal{W}$ represents the assigned weighted value based on the category scale.

In this experiment, we utilize the unbalanced dataset UNSW-NB15, which necessitates the use of metrics beyond accuracy to avoid misleading conclusions. Therefore, we employ Gmean [25], the area under the ROC curve (AUC) [26], and macro-F1 (MF1) as appropriate indicators for handling unbalanced datasets. Gmean measures the bias towards positive or negative accuracy ratios. AUC is a robust evaluation metric for binary tasks, commonly used to assess models in the presence of class skews by converting multiple classes into binary. MF1 assigns equal weight to each category and calculates the average PR and RC for all categories, which are then used in the F1 formula to obtain the result.

1) Competitors Evaluation: After conducting comparative experiments among the four proposed models, we selected the top-performing APS-T model with spatial-PNCG for comparison with other competitors. The specific comparison results are presented in Table VIII. Based on the detailed results, we draw the following conclusions.

1) The proposed method outperforms competitors with the highest AC (0.8874), PR (0.8914), RC (0.8871), and WF1(0.8876). Fig. 8 shows the confusion matrix of the APS-T model calculated by WF1, in which WF1 for six attacks can reach more than 0.9, and WF1 for all attacks can exceed 0.7.
2) DPCNN is a text-related model with a classification accuracy of 0.8790, second only to the APS-T model. It extracts long-distance text dependencies by deepening the network, which also applies to time-dependent network flows. However, the global representation and deep CNN network in DPCNN result in a significant time cost in gradient backpropagation.
3) The APS-T model achieves optimal performance with fewer epochs despite its longer training time. The construction of the graph requires more data, leading to longer loading times and increased time overhead. However, compared to DPCNN, the time cost of the proposed model is still acceptable.

The APS-T model outperforms the other nine competitors in terms of classification performance within an appropriate training time. Next, the four proposed models will be applied to other scenarios for a more comprehensive comparison.

2) Migration Scenarios: In the current experiment, we will select datasets under three different scenarios to compare the performance of nine competitors and four proposed models. The results are summarized in Table IX and the following conclusions are gained.

1) The proposed four models exhibit optimal performance across the datasets. The BoT-IoT dataset consists of seven

TABLE IX
COMPARATIVE ANALYSIS OF FOUR PROPOSED MODELS AND NINE COMPETING PRODUCTS ON THREE OPEN DATASETS WAS PERFORMED

Dataset	BoT-IoT				UNSW-NB15				DoHBrw2020			
Method	AC	PR	RC	WF1	AC	Gmean	AUC	MF1	AC	PR	RC	WF1
GID [21]	0.4000	0.1700	0.4000	0.2300	0.2500	0.4782	0.5787	0.1900	0.6100	0.6600	0.6100	0.6100
E-ResGAT [22]	0.4305	0.3565	0.3237	0.2924	0.2580	0.4838	0.5826	0.0986	0.5635	0.5102	0.5636	0.4638
CAN-GAT [23]	0.3188	0.3204	0.3188	0.2659	0.4877	0.6756	0.7118	0.1757	0.5167	0.5202	0.5167	0.5133
EnsembleRF [30]	0.8857	0.900	0.8900	0.8800	0.5673	0.7447	0.7665	0.2921	0.8857	0.9000	0.8900	0.8800
ERNN [38]	0.8424	0.8382	0.8424	0.8340	0.5622	0.7290	0.7538	0.2715	0.9282	0.8971	0.8683	0.9279
DPCNN [10]	0.9065	0.9069	0.9014	0.8973	0.8322	0.9026	0.9056	0.4963	0.9263	0.9272	0.9264	0.9267
Transformer [8]	0.8959	0.9027	0.8890	0.8892	0.8350	0.9043	0.9071	0.5759	0.9277	0.9356	0.9286	0.9283
PointNet [9]	0.9083	0.9113	0.9025	0.8991	0.8179	0.8940	0.8975	0.4775	0.9251	0.9321	0.9261	0.9256
TextGCN [5]	0.9187	0.9298	0.8410	0.8503	0.7887	0.8763	0.8811	0.4230	0.9315	0.9328	0.9315	0.9320
APS-T model+Spatial-PNCG	**0.9288**	**0.9386**	0.9244	**0.9237**	**0.8659**	**0.9227**	**0.9246**	**0.6726**	**0.9343**	0.9384	**0.9350**	**0.9346**
APS-T model+Temporal-PNCG	0.9278	0.9337	**0.9267**	0.9224	0.8343	0.9039	0.9068	0.5277	0.9297	0.9303	0.9298	0.9300
Simple model+Spatial-PNCG	0.9229	0.9335	0.9157	0.9163	0.7379	0.8464	0.8544	0.6425	0.9310	**0.9390**	0.9320	0.9314
Simple model+Temporal-PNCG	0.9274	0.9370	0.9182	0.9204	0.8261	0.8990	0.9021	0.4518	0.9330	0.9378	0.9338	0.9335

The bold entities represent the optimal classification performance achievable by the proposed model, which is consistently superior to the classification performance of existing models.

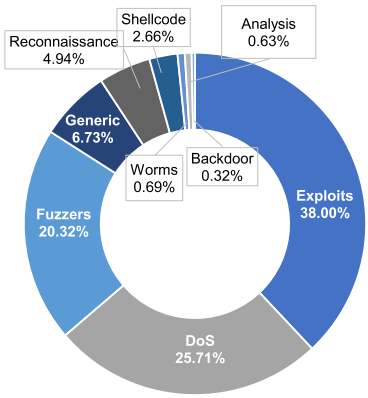

Fig. 9. Sample counts of all attack behavior categories in the unbalanced UNSW-NB15 dataset.

types of attack behaviors with a large number of attack samples (15 000 each). The UNSW-NB15 dataset used in this study has a smaller sample size of 57 517 with an unbalanced distribution of nine attack behaviors. The more comprehensive data distribution will be displayed in Fig. 9. The DoHBrw2020 dataset includes three types of attack behaviors with a large, balanced data volume of 35 000 for each behavior. Experimental results demonstrate that the proposed models enhance classification performance on imbalanced datasets and can adapt to datasets of varying scales and attack behavior counts.

2) The APS-T model consistently demonstrates superior classification performance across all datasets, indicating its strong stability. The TextGCN model demonstrates pronounced instability, with below-average accuracy scores on both the ToN-IoT and UNSW-NB15 datasets. Conversely, it achieves high accuracy rates of 0.9187 and 0.9315 on the BoT-IoT and DoHBrw2020 datasets, respectively, securing the top position among all competing models. The instability of TextGCN can be attributed to its use of MPNN to update node information without considering the importance of each node. Additionally, relying solely on a single time-series graph structure fails to accurately capture the features of network flows, leading to inherent instability. In contrast, the APS-T model adopts S-T feature extraction and incorporates the attention mechanism to measure the importance of nodes, resulting in improved performance.
3) On the small-scale and unbalanced UNSW-NB15 dataset, the classification performance of the models is generally subpar. Among the proposed models, the one that constructs network flows using spatial-PNCG and incorporates spatial graph representation achieves the highest MF1 score. However, the original model performs the worst on temporal-PNCG. These results indicate that the APS-T model based on spatial-PNCG exhibits strong robustness. Its inherent advantage lies in the feature extraction strategy combining graph structure and time series. The graph structure effectively captures features from different network flows, while the time series enhances the understanding of contextual features in the traffic data.
4) The collapse of the GID, E-ResGAT, and CAN-GAT graph models can be attributed to the lack of sufficient features for classification. These models face challenges due to the limited number of features available. For instance, the GID model treats each sample as a graph node and uses the GCN model to extract node information for classification. However, with only two features per node, the accuracy significantly decreases. Additionally, the PNCG model simulates encrypted streams, making it

unavailable to access the destination IP address and port number used in the E-ResGAT model. The CAN-GAT model primarily relies on timestamps and data content, but the lack of actual data content for encrypted streams leads to model failure when replaced with other features like PS and IPT. These three approaches may not be well-suited for detecting encrypted streams. Moreover, achieving high classification accuracy with only two features remains a challenge in the field of graph-based classification research. In contrast, our proposed model enhances classification accuracy by considering confidentiality and a limited number of features in near real-time.

VI. Conclusion

In IIoT, interconnected devices and sensors were widely used in production supply management and global resource coordination, making process monitoring crucial for safety and risk mitigation. We proposed an intrusion detection model specifically designed for the encrypted MQTT protocol in IoT, which demonstrated excellent performance. By leveraging temporal-PNCG and spatial-PNCG, constructed based on packet-level features, we captured communication connections and time-series relationships. Our model combined attention mechanism, Bi-LSTM, and residual unit to effectively handle long/short-term dependencies and assigned appropriate weights to features. Experimental results validated the effectiveness of our approach in various attack scenarios, including botnet attacks (BoT-IoT dataset), vulnerability attacks (UNSW-NB15 dataset), real-world IoT applications (ToN-IoT dataset), and DNS attacks (DoHBrw2020 dataset). Notably, our approach achieved an MF1 score of 0.6726 on the imbalanced UNSW-NB15 dataset, outperforming competing models with the highest MF1 score of only 0.5759.

Although PNCG was constructed for the encrypted MQTT protocol, it was still applicable to other security protocols that could be encrypted and support bidirectional transmission, such as the advanced message queuing protocol (AMQP). Encryption was required because we consider node characteristics in the PNCG based on encrypted transmission data extraction while providing inherent data privacy protection.

Further improvements are required in the efficiency of graph construction and model training time. Future research will focus on exploring efficient network flow graph construction and enhancing model structure and parameters to reduce computing overhead. These enhancements aim to cater to IoT edge devices, enabling effective utilization of limited computing resources.

References

[1] H. Lu, T. Wang, X. Xing, and T. Wang, "Cognitive memory-guided AutoEncoder for effective intrusion detection in Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 18, no. 5, pp. 3358–3366, May 2022.

[2] X. Zhang, J. Chen, Y. Zhou, L. Han, and J. Lin, "A multiple-layer representation learning model for network-based attack detection," *IEEE Access*, vol. 7, pp. 91992–92008, 2019.

[3] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward effective intrusion detection using log-cosh conditional variational AutoEncoder," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6187–6196, Apr. 2021.

[4] X. Han et al., "STIDM: A spatial and temporal aware intrusion detection model," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, 2020, pp. 370–377.

[5] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 3442–3448, 2019.

[6] M. Shen, Y. Liu, S. Chen, L. Zhu, and Y. Zhang, "Webpage fingerprinting using only packet length information," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[8] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6000–6010, 2017.

[9] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[10] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 562–570.

[11] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019.

[12] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.

[13] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain. Cities Soc.*, vol. 72, 2021, Art. no. 102994.

[14] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of DoH tunnels using time-series classification of encrypted traffic," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Con. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr.*, 2020, pp. 63–70.

[15] T. Gui et al., "A lexicon-based graph neural network for Chinese NER," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 1040–1050.

[16] B. Weisfeiler and A. Leman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *NauchnoTechnicheskaya Informatsiya*, vol. 2, no. 9, pp. 12–16, 1968.

[17] G. Nikolentzos, G. Dasoulas, and M. Vazirgiannis, "Permute me softly: Learning soft permutations for graph representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5087–5098, Apr. 2023.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv: 1609.02907*.

[19] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[20] R. Namazi, E. Ghalebi, S. Williamson, and H. Mahyar, "SMGRL: A scalable multi-resolution graph representation learning framework," 2022, *arXiv:2201.12670*.

[21] Y. Zhang, C. Yang, K. Huang, and Y. Li, "Intrusion detection of industrial Internet-of-Things based on reconstructed graph neural networks," *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: 10.1109/TNSE.2022.3184975.

[22] Chang L. and Branco P., "Graph-based solutions with residuals for intrusion detection: The modified E-GraphSAGE and E-ResGAT algorithms," 2021, *arXiv:2111.13597*.

[23] J. Xiao, L. Yang, F. Zhong, H. Chen, and X. Li, "Robust anomaly-based intrusion detection system for in-vehicle network by graph neural network framework," *Appl. Intell.*, vol. 53, no. 3, pp. 3183–3206, 2022.

[24] R. F. Bikmukhamedov and A. F. Nadeev, "Multi-class network traffic generators and classifiers based on neural networks," in *Proc. Syst. Signals Generating Process. Field Board Commun.*, 2021, pp. 1–7.

[25] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 179–186.

[26] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[27] T. Kovanen, G. David, and T. Hämäläinen, "Survey: Intrusion detection systems in encrypted traffic," in *Proc. Internet Things, Smart Spaces, Next Gener. Netw. Syst.: 16th Int. Conf., NEW2AN, 9th Conf., ruSMART*, 2016, pp. 281–293.

[28] C. F. T. Pontes, M. M. C. de Souza, J. J. C. Gondim, M. Bishop, and M. A. Marotta, "A new method for flow-based network intrusion detection using the inverse potts model," *IEEE Trans. Netw. Serv. Manage.*, vol. 18, no. 2, pp. 1125–1136, Jun. 2021.

[29] A. Fais, G. Lettieri, G. Procissi, S. Giordano, and F. Oppedisano, "Data stream processing for packet-level analytics," *Sensors*, vol. 21, no. 5, 2021, Art. no. 1735.
[30] N. Bayat, W. Jackson, and D. Liu, "Deep learning for network traffic classification, 2021, *arXiv:2106.12693*.
[31] R. W. Liu et al., "STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multigraph convolutional network," *IEEE Trans. Ind. Inform.*, vol. 18, no. 11, pp. 7977–7987, Nov. 2022.
[32] Z. Huang, M. Xia, M. Lu, L. Pan, and J. Liu, "AWL-GCN: Branch parameter identification considering grid spatial structure constraints," *IEEE Trans. Ind. Inform.*, vol. 19, no. 5, pp. 6939–6949, May 2023.
[33] W. Chen, F. Lyu, F. Wu, P. Yang, G. Xue, and M. Li, "Sequential message characterization for early classification of encrypted Internet traffic," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3746–3760, Apr. 2021.
[34] Z. Jia and W. Lu, "An End-to-End hyperspectral image classification method using deep convolutional neural network with spatial constraint," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 10, pp. 1786–1790, Oct. 2021.
[35] X. Wu, Z. Luo, Z. Du, J. Wang, C. Gao, and X. Li, "TW-TGNN: Two windows graph-based model for text classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2021, pp. 1–8.
[36] Y. Zhao, Y. Yang, B. Tian, J. Yang, T. Zhang, and N. Hu, "Edge intelligence based identification and classification of encrypted traffic of Internet of Things," *IEEE Access*, vol. 9, pp. 21895–21903, 2021.
[37] F. Baig, D. Teng, J. Kong, and F. Wang, "SPEAR: Dynamic spatio-temporal query processing over high velocity data streams," in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 2279–2284.
[38] Z. Zhao et al., "ERNN: Error-resilient RNN for encrypted traffic detection towards network-induced phenomena," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2023.3242134.

Xinlei Wang is currently working toward the Ph.D. degree in electronic science and technology with the Beijing University of Posts and Telecommunications, Beijing, China.

Her current research interests include machine learning, network security, data poisoning, and deep learning architectures.

Xiaojuan Wang received the Ph.D. degree in electronic science and technology from the University of Beijing University of Posts and Telecommunications, Beijing, China, in 2015.

She is currently an Associate Professor with the School of Electronic Engineering, Beijing University of Posts and Telecommunications. Her research interests include deep learning, complex networks, and human gesture recognition.

Mingshu He (Member, IEEE) received the Ph.D. in electronic science and technology degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2022.

He is currently doing research with the School of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interests include network security, anomaly detection, and machine learning.

Min Zhang is currently working toward the Ph.D. degree in electronic science and technology from the Beijing University of Posts and Telecommunications, Beijing, China.

Her research interests include artificial intelligence, network science, and differential equation.

Zikui Lu is currently working toward the Ph.D. degree in electronic science and technology with the Beijing University of Posts and Telecommunications, Beijing, China.

He has also actively participated in numerous national-level research projects. His research interests include traffic analysis, representation learning, vulnerability mining, and model acceleration and optimization.