Uncertainty-Aware Heterogeneous Graph Meta-Learning with Hybrid Attention for IoT Intrusion Detection

Fengzhao Li[1,2], Wei Zhang[1,2*], Huiling Shi[1,2]
[1] *Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China*
[2] *Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan, China*
Email: 10431240235@stu.qlu.edu.cn, wzhang@sdas.org*, shihl@sdas.org

***Abstract*—With the rapid expansion of the Internet of Things (IoT), heterogeneous and noisy traffic enlarges the attack surface, increasing the risks of malicious communication, denial-of-service attacks, and malware propagation. Existing methods often model traffic samples independently, failing to capture the interaction structure between traffic flows and network resources, and they generalize poorly to emerging or few-shot attacks under scarce labels and evolving attack patterns. To address these issues, this paper proposes an uncertainty-aware heterogeneous graph meta-learning framework, termed UA-MetaHGN, for few-shot IoT malicious traffic detection. Specifically, network flows, IPs, and ports are modeled as heterogeneous nodes with multiple relations; a hybrid-attention hierarchical representation learning scheme is designed to capture high-order dependencies and enhance discriminative representations; and task-adaptive initialization and dynamic task weighting are incorporated into meta-learning, leveraging loss discrepancies, heteroscedastic uncertainty, and Bayesian regularization to improve fast adaptation and robust generalization. Experiments on the ToN-IoT dataset show that UA-MetaHGN consistently outperforms state-of-the-art baselines under both few-shot (3/5/10-shot) and standard supervised settings.**

***Index Terms*—Internet of Things, malicious traffic detection, heterogeneous graph, meta-learning, heteroscedastic uncertainty.**

I. Introduction

With the continuous expansion of the Internet of Things (IoT) ecosystem, the number of connected devices is rapidly increasing [1]. This growth generates massive and heterogeneous traffic from the network-layer, which broadens the attack surface and facilitates malicious communications, denial-of-service attacks, and malware propagation [2]. Consequently, network-traffic-based detection remains a critical technique for the protection of IoT security [3].

Existing studies typically derive statistical or behavioral features from network traffic and apply machine learning or deep learning models for classification. However, most methods treat traffic flows as independent samples, making it difficult to explicitly model interaction structures between flows and network resources [4]. In addition, approaches that rely heavily on handcrafted features are susceptible to adversarial traffic manipulation, which compromises robustness in real-world deployments [5].

To better exploit cross-entity relational information, heterogeneous graph neural networks (HGNNs) offer a natural and expressive modeling paradigm. By representing network entities as nodes and their interactions as multiple node and relation types, heterogeneous graphs integrate local traffic attributes with global relational dependencies in a unified structure. Attention mechanisms can also adaptively weigh different relations, yielding more robust and discriminative representations for IoT traffic analysis [6].

In practical deployments, malicious traffic detection often faces scarce labeled data and rapidly evolving attack patterns, which substantially weakens the generalization of supervised models, particularly for novel or rare attacks [7]. Recent work suggests that combining heterogeneous graph representation learning and meta-learning enables fast adaptation to unseen classes in few-shot contexts via task-level optimization [8]. However, for heterogeneous graphs with diverse relation types and task distributions, existing methods typically use simplistic relation aggregation and globally shared initializations. They tend to treat all tasks as equally important during meta-training, which can downplay critical relations and overfit to high-uncertainty tasks, reducing robustness and generalization [9], [10].

To address these challenges, we develop UA-MetaHGN, an uncertainty-aware heterogeneous graph meta-learning framework with a hybrid attention mechanism for few-shot IoT malicious traffic detection. By unifying heterogeneous graph modeling and meta-learning, UA-MetaHGN enhances the representation of complex relational structures and enables rapid adaptation in few-shot and emerging attack scenarios.

In summary, the following are the primary contributions of our work:

- We model traffic flows, IP addresses, and ports as distinct node types and define multiple relation types, including *flow–source/destination IP* and *flow–source/destination*

port, to capture diverse interaction patterns among network entities.

- We propose a relation-aware attention mechanism that learns adaptive relation weights and node-pair attention for aggregating heterogeneous relations. We further introduce a multi-hop node attention module to capture higher-order dependencies by constructing a hop-expanded sparse adjacency with decayed hop importance, scoring candidate neighbors, and stabilizing long-range aggregation via iterative attention diffusion. In addition, a semantic attention module is used to fuse relation-level and multi-hop representations into more discriminative embeddings.
- In the meta-learning stage, we propose a task-adaptive, uncertainty-calibrated meta-optimization framework, featuring task-adaptive initialization via a parameter buffer, difficulty–uncertainty joint task reweighting based on the support–query loss gap and heteroscedastic uncertainty, and uncertainty regularization with a unified meta-objective, which improves fast adaptation and generalization under task uncertainty and limited labeled data.

The remainder of this paper is structured as follows: Section II introduces the relevant background and formulates the problem. Section III presents the heterogeneous graph construction and representation learning method and elaborates the uncertainty-aware meta-learning optimization strategy. Section IV conducts extensive experiments to validate the effectiveness of the proposed approach for few-shot IoT malicious traffic detection; and Section V concludes the paper.

II. Background and Related Work

A. Machine Learning and Deep Learning-Based IoT Anomaly Traffic Detection

In recent years, machine learning techniques have been widely applied to IoT anomalous traffic detection. Most early studies adopt supervised frameworks based on handcrafted traffic features. These features are then fed into conventional classifiers such as decision trees, random forests, and support vector machines (SVMs) to identify malicious traffic. For example, Gu *et al.* [11] proposed an NB-SVM framework that uses Naive Bayes feature transformation with SVM classification to enhance feature discriminability. To mitigate the reliance on manual feature engineering and improve representation capacity, deep learning-based detection models have been further explored for IoT traffic analysis. Existing approaches include sequence modeling with CNNs, RNNs, and LSTMs, as well as end-to-end classification frameworks for multi-class attack recognition. Li *et al.* [12] proposed MTCD-Model, which combines CNN–Bi-SRU hierarchical feature learning with a capsule network for malicious traffic classification and detection; focal and reconstruction losses further enable end-to-end learning from variable-length raw traffic, reducing feature-engineering dependence.

Although existing studies achieve good performance on standard datasets and specific attack types, most of them still treat network traffic as independent samples and fail to explicitly model the interaction structures between traffic flows and network resources. As a result, their robustness and transferability are limited in scenarios involving complex communication relationships, cross-entity coordinated attacks, and evolving attack patterns.

B. IoT Anomaly Traffic Detection Based on Graph Neural Network

To overcome the limitations of the "independent sample" assumption, researchers have abstracted network communication processes into graph structures to explicitly model the interactions among entities. Building upon this, Lo *et al.* [13] proposed a Graph Neural Network (GNN) method, E-GraphSAGE, which integrates edge features and topological information, and is the first to implement edge classification based on flow data for IoT anomaly detection. Deng *et al.* [14] introduced the FT-GCN method, which constructs interval-constrained traffic graphs and designs node-level spatial attention, achieving good performance in IoT intrusion detection scenarios with limited labeled data.

However, in IoT scenarios, existing graph-based detection methods primarily rely on homogeneous graph modeling and local neighborhood aggregation, making it difficult to fully represent the complex interactions between multiple types of nodes and their relational semantics in network traffic. These methods also struggle to capture high-order dependencies and propagation effects across multi-hop links, limiting the models' ability to detect anomalous traffic effectively.

To address the limitations of current methods, La and Li *et al.* [15], [16] proposed the ML-HAN and MSHGT methods based on Heterogeneous Graph Neural Networks (HGNNs). By learning the contribution weights of different relations at the relation/edge type level and performing fusion of representations generated from various relations or multiple meta-paths at the semantic level, these methods improve the expressive power for heterogeneous structures and multi-semantic dependencies, providing a more natural framework for modeling complex interaction systems.

C. Applications of Few-Shot Learning and Meta-Learning in Traffic Detection

In real-world deployments, IoT malicious traffic detection often faces the challenges of limited labeled data and the rapid emergence of new attack types, which makes it difficult for traditional supervised learning models to quickly adapt to novel classes or few-shot conditions. Few-shot learning and meta-learning provide feasible solutions to these challenges. Xu *et al.* [17] proposed a meta-learning–based few-shot intrusion detection method that uses an FC-Net to measure the similarity between pairs of network traffic samples, mitigating limited labeled data in zero-day scenarios and improving detection of emerging attacks. Pawlicki *et al.* [18] proposed a Siamese-network-based real-time intrusion detection approach using NetFlow features, and introduce a k-NN-driven border extraction sampling strategy to select informative boundary

samples, mitigating pairwise-combination explosion and improving generalization to previously unseen attacks. Furthermore, Wang *et al.* [19] proposed a graph meta-learning model (AG-Meta), which introduces a consistency mechanism into the graph representation to improve few-shot generalization. These studies improve detection performance to some extent under few-shot settings; however, when facing tasks with large discrepancies and high uncertainty, the generalization capability of the models remains limited.

III. System model and problem formulation

We study few-shot IoT intrusion detection under heterogeneous relational structures, where tasks exhibit varying difficulty and uncertainty. Our goal is to learn a meta-learner that can (i) extract relation- and hop-aware representations, and (ii) adapt rapidly to unseen attack types with uncertainty-calibrated updates.

Fig. 1 illustrates the overall framework, which contains three stages: (1) heterogeneous graph construction from IoT traffic data; (2) hierarchical heterogeneous representation learning that models relation-, hop-, and semantic-level dependencies; and (3) a task-adaptive meta-training stage equipped with uncertainty-aware task reweighting.

A. Node–Relation Construction in Heterogeneous Graphs

To characterize interactions between traffic and network resources, we model raw data as a heterogeneous graph with three node types: IP address nodes, network flow nodes, and port nodes. The associations among node types are represented by multiple relation-specific adjacency matrices, where each relation corresponds to an independent matrix describing connectivity under a specific semantic relation. Formally, given a graph with N nodes and K relation types, the relation adjacency matrices are denoted as:

$$\mathbf{A} = \left\{ \mathbf{A}^{(k)} \in \mathbb{R}^{N\times N} \,\middle|\, k = 1, \ldots, K \right\}. \tag{1}$$

where $\mathbf{A}^{(k)}$ denotes the adjacency matrix corresponding to the k-th relation type, and $\mathbb{R}$ is the set of all relation types, and N represents the number of nodes in the graph.

B. Relation Attention Mechanism

Since a single adjacency matrix only encodes a single relation type, it is insufficient to capture complex cross-relation semantics. To comprehensively model the relational information in the graph structure, we first aggregate multiple relation adjacency matrices to obtain a unified adjacency matrix $\tilde{\mathbf{A}}$ via an MLP:

$$\tilde{\mathbf{A}} = \mathrm{MLP}\left(\sum_{k=1}^{K} \omega_k \mathbf{A}^{(k)} \right). \tag{2}$$

where ω_k denotes the weight coefficient for the k-th relation type.

To better capture relational information in the heterogeneous graph, we introduce a relation-attention mechanism. For node representations $\mathbf{h}_i$ and $\mathbf{h}_j$, we calculate the relation-attention coefficient through a deep neural network $\mathrm{Attn}_{\mathrm{rel}}$:

$$e_{ij}^{(k)} = \mathrm{Attn}_{\mathrm{rel}}^{(k)}(\mathbf{h}_i, \mathbf{h}_j), \quad j \in N_i. \tag{3}$$

where $e_{ij}^{(k)}$ is used to quantify the relative importance of node representations $\mathbf{h}_i$ and $\mathbf{h}_j$ under relation k.

To normalize the attention scores across all nodes, the softmax function is applied as follows:

$$p_{ij}^{(k)} = \mathrm{softmax}(e_{ij}^{(k)}) = \frac{\exp\big(\mathrm{Attn}_{\mathrm{rel}}^{(k)}(\mathbf{h}_i, \mathbf{h}_j)\big)}{\sum_{l\in N_i} \exp\big(\mathrm{Attn}_{\mathrm{rel}}^{(k)}(\mathbf{h}_i, \mathbf{h}_l)\big)}. \tag{4}$$

In heterogeneous graphs, different relation types are crucial for node representations. To better represent each node, we integrate these relational features into the model, generating the embedding of node i through a linear combination of the adjacency matrix and attention coefficients.

$$\mathbf{h}_i^{(k)} = \sigma\left(\sum_{j\in N_i} p_{ij}^{(k)} \, \mathbf{W}^{(k)} \, \mathbf{h}_j \right). \tag{5}$$

where σ is the nonlinear activation function, ReLU. $\mathbf{W}^{(k)}$ is a relation-specific transformation matrix, and $\mathbf{h}_j$ represents the feature vector of node v_j.

C. Multi-Hop Node Attention Mechanism

In IoT graphs, informative dependencies may span multiple hops. However, explicitly computing matrix powers can densify the graph and is computationally expensive. We therefore model multi-hop information via sparse iterative propagation, and further refine it with an attention diffusion mechanism to stabilize long-range aggregation.

1) Hop-Expanded Adjacency $\mathbf{A}^{(\ell)}$: Starting from a sparse normalized adjacency $\mathbf{A}$, we iteratively expand neighbors up to ℓ hops and keep at most X candidates per hop. The resulting candidate edge set is represented by $\mathbf{A}^{(\ell)}$ and its corresponding neighbor set $N_i^{(\le \ell)}$. We use a decayed hop-importance factor $0 < \rho < 1$ to down-weight distant hops; conceptually, $\mathbf{A}^{(\ell)}$ aggregates hop information with decay, while remaining sparse by construction: $\mathbf{A}^{(\ell)}$ encodes $N_i^{(\le \ell)}$ obtained by iterative sparse expansion with decay factor ρ.

2) Multi-Hop Attention and Attention Diffusion: We compute attention logits over the multi-hop candidate neighborhood:

$$s_{ij}^{(\ell)} = \delta\Big(\mathbf{v}^{(\ell)\top} \tanh\big(\mathbf{W}_q^{(\ell)} \mathbf{h}_i^{(\ell)} + \mathbf{W}_k^{(\ell)} \mathbf{h}_j^{(\ell)} \big) \Big), \quad j \in N_i^{(\le \ell)}. \tag{6}$$

where $\delta(\cdot)$ is a nonlinear activation function (LeakyReLU), $\mathbf{W}_q^{(\ell)}$ and $\mathbf{W}_k^{(\ell)}$ are trainable projections, and $\mathbf{h}_i^{(\ell)}$ and $\mathbf{h}_j^{(\ell)}$ denote the embeddings of nodes i and j at layer ℓ, respectively.

We further introduce an attention diffusion module, which diffuses information over the graph using the adjacency structure and enables propagation beyond direct neighbors across multiple hops. The attention diffusion process is iteratively performed by the following equation:

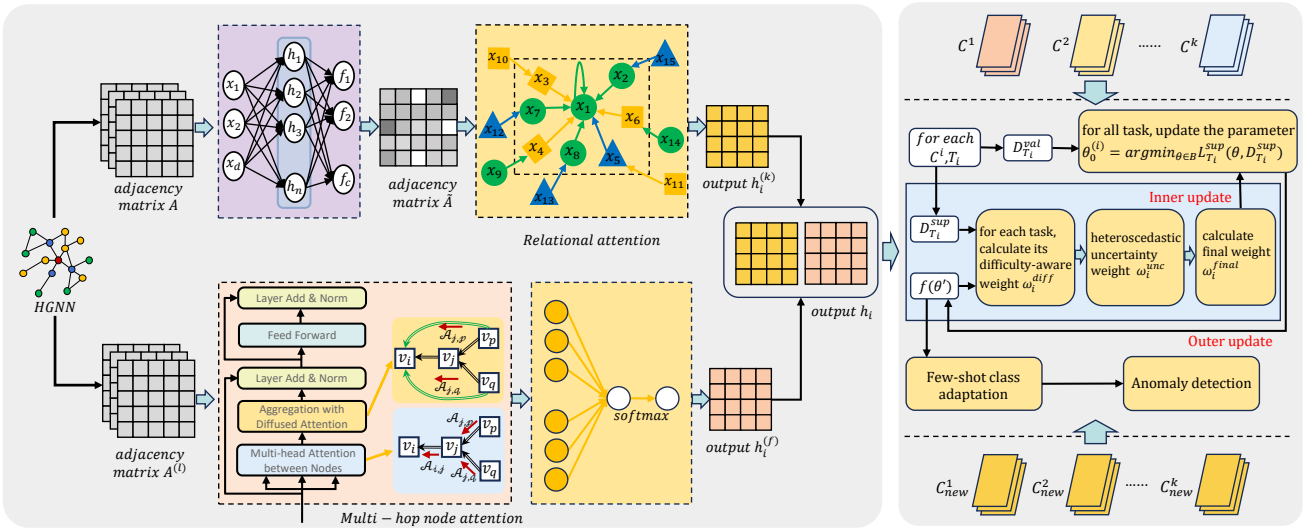

Fig. 1. Overall framework of the proposed method.

$$\mathbf{Z}^{(t+1)} = (1-\gamma)\mathbf{P}\mathbf{Z}^{(t)} + \gamma\mathbf{Z}^{(1)}, \tag{7}$$

where $\gamma \in (0,1]$ is the teleport probability, $\mathbf{P}$ is the normalized attention matrix, and $\mathbf{Z}^{(t)}$ denotes the diffused attention matrix after t diffusion steps.

To obtain consistent scores across different nodes, we apply the softmax function to all available neighboring nodes for normalization, yielding normalized weight coefficients:

$$\phi_{ij}^{(\ell)} = \frac{\exp\left(\mathbf{Z}^{(t)}\right)}{\sum_{l\in N_i^{(\leq\ell)}} \exp\left(\mathbf{Z}^{(t)}\right)}, \quad j \in N_i^{(\leq\ell)}. \tag{8}$$

where $\phi_{ij}^{(\ell)}$ denotes the final multi-hop attention weight from node j to node i at layer ℓ.

After obtaining the attention weights over multi-hop neighborhoods, we aggregate the multi-hop neighbor information of each node according to the attention scores, thereby obtaining the final embedding representation of the target node:

$$\mathbf{h}_i^{(f)} = \sum_{j\in N_i} \phi_{ij}^{(\ell)} \mathbf{W}^{(f)}\ \mathbf{h}_j. \tag{9}$$

D. Feature Fusion

Based on the above two attention mechanisms, we introduce a semantic attention module to fuse node and relation information.

$$\pi_i = \mathrm{softmax}\left(\mathbf{W}_s\left[\mathbf{h}_i^{(k)} \parallel \mathbf{h}_i^{(f)}\right] + \mathbf{b}_s\right), \tag{10}$$

$$\mathbf{h}_i = \pi_i^{(k)}\mathbf{h}_i^{(k)} + \pi_i^{(f)}\mathbf{h}_i^{(f)}. \tag{11}$$

where π_i is the semantic attention weight vector for node i, $\pi_i^{(k)}$ and $\pi_i^{(f)}$ denote the weights associated with $\mathbf{h}_i^{(k)}$ and $\mathbf{h}_i^{(f)}$, respectively. $\mathbf{W}_s$ are the learnable weight matrix and $\mathbf{b}_s$ are the bias term.

Finally, the fused features are processed through a fully connected layer. This yields graph embeddings with high-order semantics for downstream tasks.

E. Meta-Learning Stage

Through the above stages, we obtain an embedding for each traffic sample by aggregating heterogeneous neighbors and semantic information. In real-world IoT deployments, intrusion detection often faces label scarcity and distribution shift, and few-shot tasks may vary substantially in difficulty and noise. Therefore, we cast IoT intrusion detection as a meta-learning problem and propose a task-adaptive, uncertainty-calibrated meta-optimization strategy to enable fast and stable adaptation to unseen attack types.

1) Task-Adaptive Initialization: Tasks in IoT traffic detection can exhibit heterogeneous distributions. Using a single shared initialization may be suboptimal for rapid adaptation. Therefore, we maintain a parameter buffer $\mathcal{B} = \{\theta^{(1)}, \ldots, \theta^{(M)}\}$, which stores the model parameters from the most recent M outer-loop updates. For each sampled task $\mathcal{T}_i$, we select an initialization from $\mathcal{B}$ that yields the smallest support loss:

$$\theta_0^{(i)} = \arg\min_{\theta\in\mathcal{B}} \mathcal{L}_{\mathcal{T}_i}^{\mathrm{sup}}\left(\theta, \mathcal{D}_{\mathcal{T}_i}^{\mathrm{sup}}\right). \tag{12}$$

where M is a constant ranging from 5 to 10, and $\mathcal{L}_{\mathcal{T}_i}^{\mathrm{sup}}$ is the loss evaluated on the support set $\mathcal{D}_{\mathcal{T}_i}^{\mathrm{sup}}$.

2) Difficulty–Uncertainty Joint Task Reweighting: Different few-shot tasks contribute unequally to meta-training. In particular, hard tasks may require more emphasis, while noisy tasks with high predictive uncertainty should contribute less to avoid destabilizing meta-updates. We therefore jointly model task *difficulty* and *reliability* and use a unified weight to reweight the meta-loss of each task.

Task difficulty. We measure task difficulty by the support–query loss gap:

$$\Delta_i = \mathcal{L}_{\mathcal{T}_i}^{\text{qry}} - \mathcal{L}_{\mathcal{T}_i}^{\text{sup}}, \tag{13}$$

and assign a difficulty-aware weight via softmax normalization:

$$\omega_i^{\text{diff}} = \frac{\exp(\tau \Delta_i)}{\sum_{j=1}^{B} \exp(\tau \Delta_j)}. \tag{14}$$

where B is the number of tasks in the current meta-batch and $\tau > 0$ controls the sharpness of the weighting distribution.

Task reliability from heteroscedastic uncertainty. We estimate sample-level heteroscedastic uncertainty by learning a variance term $\sigma^2(x)$ for each sample x. The uncertainty-weighted loss for task $\mathcal{T}_i$ is:

$$\mathcal{L}_{\mathcal{T}_i}^{\text{het}} = \frac{1}{\left|\mathcal{D}_{\mathcal{T}_i}^{\text{qry}}\right|} \sum_{(x,y)\in\mathcal{D}_{\mathcal{T}_i}^{\text{qry}}} \left(\frac{\ell(z(x), y)}{\sigma^2(x)} + \alpha \log \sigma(x) \right), \tag{15}$$

where $\ell(z(x), y)$ is the standard classification loss, α is a scaling factor, and larger uncertainty $\sigma^2(x)$ yields a smaller effective weight, reducing the influence of outliers samples.

To aggregate uncertainty at the task level, we define the task uncertainty as the average variance across query samples:

$$\bar{\sigma}_i^2 = \frac{1}{\left|\mathcal{D}_{\mathcal{T}_i}^{\text{qry}}\right|} \sum_{(x,y)\in\mathcal{D}_{\mathcal{T}_i}^{\text{qry}}} \sigma^2(x). \tag{16}$$

We then convert task uncertainty into an uncertainty-based reliability weight, and tasks with higher uncertainty receive lower weights:

$$\omega_i^{\text{unc}} = \frac{\exp\left(-\lambda \bar{\sigma}_i^2\right)}{\sum_{j=1}^{B} \exp\left(-\lambda \bar{\sigma}_j^2\right)}. \tag{17}$$

Finally, we combine the two signals into a unified task weight:

$$\omega_i^{\text{final}} = \frac{\omega_i^{\text{diff}} \cdot \omega_i^{\text{unc}}}{\sum_{j=1}^{B} \omega_j^{\text{diff}} \cdot \omega_j^{\text{unc}}}. \tag{18}$$

This unified weighting explicitly emphasizes difficult tasks while down-weighting high-uncertainty tasks, leading to more stable and generalizable meta-updates.

3) Uncertainty Regularization and Meta-Objective: To avoid extreme uncertainty values and encourage stable uncertainty estimation, we introduce a regularization term:

$$\mathcal{L}_{\mathcal{T}_i}^{\text{reg}} = \frac{1}{\left|\mathcal{D}_{\mathcal{T}_i}^{\text{qry}}\right|} \sum_{(x,y)\in\mathcal{D}_{\mathcal{T}_i}^{\text{qry}}} \left(\lambda \log \sigma(x) - \mu_0\right)^2, \tag{19}$$

where μ_0 is a prior mean and λ is a regularization parameter to control the uncertainty range.

After applying the above steps, the final meta-objective is defined as:

$$\mathcal{L}_{\text{meta}}(\theta) = \sum_{\mathcal{T}_i \in \mathcal{B}_{\text{task}}} \omega_i^{\text{final}} \left(\mathcal{L}_{\mathcal{T}_i}^{\text{het}} + \eta\, \mathcal{L}_{\mathcal{T}_i}^{\text{reg}} \right). \tag{20}$$

where $\mathcal{B}_{\text{task}}$ denotes the current meta-batch of tasks, and η controls the strength of the uncertainty regularization.

The overall training procedure of UA-MetaHGN is summarized in Algorithm 1.

Algorithm 1: Training Procedure of UA-MetaHGN

Input : Relation adjacencies $\{\mathbf{A}^{(k)}\}_{k=1}^{K}$; node features $\mathbf{X}$; max hop L; diffusion steps t; teleport prob. γ; hop decay ρ; buffer size M; task distribution $p(\mathcal{T})$; uncertainty params $\alpha, \lambda, \eta, \mu_0$.

Output: Meta-trained parameters θ.

Compute unified adjacency $\tilde{\mathbf{A}} \leftarrow \sum_{k=1}^{K} \omega_k \mathbf{A}^{(k)}$

foreach $i \in V$ **do**

  Find one-hop neighbors N_i from $\tilde{\mathbf{A}}$

  for $k = 1$ **to** K **do**

    foreach $j \in N_i$ **do**

      $p_{ij}^{(k)} \leftarrow \text{softmax}\Big(\text{Attn}_{\text{rel}}^{(k)}(\mathbf{h}_i, \mathbf{h}_j)\Big)$

    end

    $\mathbf{h}_i^{(k)} \leftarrow \sigma\Big(\sum_j p_{ij}^{(k)} \mathbf{W}^{(k)} \mathbf{h}_j\Big)$

  end

  Build multi-hop candidates $N_i^{(\le L)}$ with decay ρ, obtain attention matrix $\mathbf{P}$

  Initialize $\mathbf{Z}^{(1)} \leftarrow \mathbf{P}$

  for $r = 1$ **to** $t - 1$ **do**

    $\mathbf{Z}^{(r+1)} \leftarrow (1-\gamma)\mathbf{P}\mathbf{Z}^{(r)} + \gamma \mathbf{Z}^{(1)}$

  end

  $\phi_{ij} \leftarrow \text{softmax}(\mathbf{Z}^{(t)})$

  $\mathbf{h}_i^{(f)} \leftarrow \sum_j \phi_{ij} \mathbf{W}^{(f)} \mathbf{h}_j$

  $\boldsymbol{\pi}_i \leftarrow \text{softmax}\big(\mathbf{W}_s[\mathbf{h}_i^{(k)} \| \mathbf{h}_i^{(f)}] + \mathbf{b}_s\big)$

  $\mathbf{h}_i \leftarrow \pi_i^{(k)} \mathbf{h}_i^{(k)} + \pi_i^{(f)} \mathbf{h}_i^{(f)}$

end

Initialize buffer $\mathcal{B} = \{\theta^{(1)}, \ldots, \theta^{(M)}\}$

while *not converged* **do**

  Sample meta-batch tasks $\mathcal{T}_i \sim p(\mathcal{T})$

  foreach $\mathcal{T}_i$ **do**

    Select initialization $\theta_0^{(i)} \leftarrow \arg\min_{\theta \in \mathcal{B}} \mathcal{L}_{\mathcal{T}_i}^{sup}(\theta)$

    Compute difficulty $\Delta_i \leftarrow \mathcal{L}_{\mathcal{T}_i}^{qry} - \mathcal{L}_{\mathcal{T}_i}^{sup}$

    $\omega_i^{diff} \leftarrow \text{softmax}(\tau \Delta_i)$

    Compute heteroscedastic loss $\mathcal{L}_{\mathcal{T}_i}^{het}$ and task uncertainty $\bar{\sigma}_i^2$ $\omega_i^{unc} \leftarrow \text{softmax}(-\lambda \bar{\sigma}_i^2)$

    $\omega_i^{final} \propto \omega_i^{diff} \cdot \omega_i^{unc}$

    Compute uncertainty regularization $\mathcal{L}_{\mathcal{T}_i}^{reg}$

  end

  Update Θ by minimizing $\sum_{\mathcal{T}_i} \omega_i^{final} \big(\mathcal{L}_{\mathcal{T}_i}^{het} + \eta \mathcal{L}_{\mathcal{T}_i}^{reg}\big)$ Refresh buffer $\mathcal{B}$ with latest θ

end

Adapt on support of target task; predict labels on query using $f_{\theta'}$

IV. Experiments

In this section, we introduce the implementation details and experimental settings of UA-MetaHGN, including the ToN-IoT dataset preprocessing, N-way K-shot task construction,

and evaluation protocols. We then report comprehensive experimental results on ToN-IoT, where we (1) compare UA-MetaHGN with representative baselines under few-shot settings with different support sizes (3/5/10-shot), (2) further evaluate its performance in the ordinary supervised learning scenario, and (3) conduct ablation studies analyses to verify the effectiveness and robustness of key components.

A. Experimental Settings

1) Dataset and Preprocessing: We conducted experiments on the ToN-IoT public dataset. First, we perform duplicate removal to handle repeated flows in the dataset. Considering that attack traffic is typically scarce in real network environments, we filter and partition the samples following the principle of "dominantly normal traffic and scarce malicious traffic," which better reflects the class imbalance characteristic of actual deployment. The preprocessed ToN-IoT dataset contains the categories shown in Table I.

TABLE I
STATISTICS AND SPLIT OF THE TON-IOT DATASET.

Dataset	Classes	Samples
ToN-IoT	Benign	27,027
	xss	9,994
	password	15,629
	ransomware	142
	scanning	2146
	mitm	1295
	backdoor	1724
	injection	46,853
	ddos	32,634
	dos	1771

2) Few-Shot Task Construction: We formulate malicious traffic detection as a meta-learning problem on a set of N-way K-shot tasks. We choose two classes in datasets at random to serve as meta-testing classes, and the remaining classes are utilized to build meta-training tasks. There are only K samples for each class in the support set of the meta-training and meta-testing tasks ($K = 3$, $K = 5$, or $K = 10$) for all datasets. To reduce the effect of class sampling in few-shot settings, we evaluate all models in the same data batches and report average accuracy. We also use stratified sampling when splitting train/validation/test subsets with a ratio of 80%/10%/10%, preserving the original class distribution.

3) Baselines: To validate the effectiveness of our proposed method in the few-shot setting, we compare UA-MetaHGN with several representative models in graph neural networks, including LIGHTGBM, GraphSAGE, GCN, SGC, HAN, Meta-GCN, and Meta-SGC. We conducted comparisons under the same dataset and optimal parameter settings. It is worth noting that we only modified the dataset partition to meet the few-shot learning requirements; all other model variables remain unchanged from their initial implementation.

4) Experiments and Parameter Setup: Our method is implemented based on PyTorch and combines DGL and PyTorch Geometric (PyG). The relation attention and multi-hop node attention both use a two-layer graph encoder, with a hidden dimension set to $d = 64$. During training, we use a Dropout rate of 0.5 and weight decay of 0.0001.

In the multi-hop aggregation module, we set the maximum hop number to $L = 3$ and the decay factor to $\rho = 0.8$. Attention diffusion uses a jump probability of $\gamma = 0.9$ with $t = 3$ iterations. For meta-learning, the outer-loop and inner-loop learning rates are $lr_1 = 0.0005$ and $lr_2 = 0.02$, respectively, with $K = 3$ inner updates. We apply early stopping based on validation loss and restore the best checkpoint.

For task-adaptive initialization, we maintain a candidate set $\{\theta\}$ from the last few outer-loop updates and select the parameters with the lowest support-set loss as initialization. In task-dynamic weighting, the support–query loss-difference threshold and the uncertainty-regularization hyperparameters (α, λ, η) are tuned in the meta-validation set. All experiments use the same hardware and a fixed random seed for reproducibility.

B. Detection Results Analysis

1) Few-shot Sample Learning Ability: As shown in Fig. 2, the proposed method consistently achieves the best F1-score under different K-shot settings, with a more pronounced advantage in the 3-shot case, indicating stronger generalization and faster adaptation under extremely limited annotations. Overall, heterogeneous-graph methods outperform homogeneous-graph methods, suggesting that explicitly modeling the "traffic–IP–port" interaction structure provides more discriminative contextual information. Furthermore, by incorporating a hybrid attention mechanism and task-adaptive dynamic weighting on top of heterogeneous relation modeling, our model can more effectively aggregate critical relations and higher-order dependencies, thereby achieving stable and superior performance in most settings.

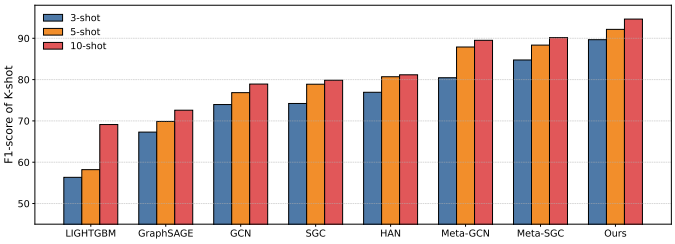

Fig. 2. Performance comparisons of all methods with different support data sizes

2) Learning Ability from Ordinary Sample Scenarios: Table II details the detection performance comparison between our method and representative existing models on the ToN IoT dataset. The results indicate that, compared with the other models, our model achieves the best overall performance across the evaluated metrics. Fig. 3 shows the complete confusion matrix of our classification results, which illustrates the distribution of the ground-truth and predicted classes. The experimental results demonstrate that the proposed method not only enables fast adaptation in few-shot settings, but also

provides robust and effective detection capability under the standard supervised learning setting.

TABLE II
PERFORMANCE COMPARISON WITH EXISTING METHODS USING THE OVERALL DATASET.

Model	A(%)	P(%)	R(%)	F(%)
Mao et al. [7]	94.32	74.28	72.46	73.31
Booij et al. [20]	98.07	–	–	97.26
Mehedi et al. [21]	87.25	88.18	86.20	86.32
Duan et al. [22]	93.37	92.46	93.68	92.55
Jiang et al. [23]	97.11	94.50	97.74	96.10
Ours	**99.57**	**98.62**	**97.86**	**98.16**

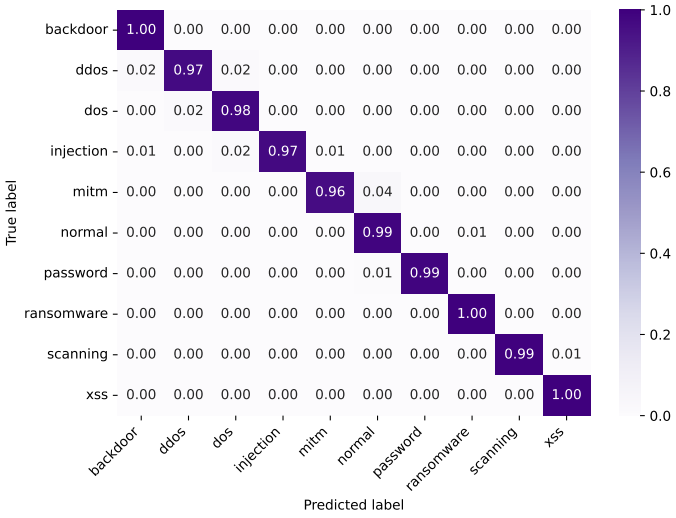

Fig. 3. Performance comparison with existing methods

3) Ablation Study: To verify the contribution of key component in the proposed model and its impact on the overall performance, we conduct three ablation studies. Table III reports the F1-score comparisons of different variants under 3/5/10-shot settings. The full model achieves F1-scores of 89.66/92.16/94.65. Removing multi-hop attention reduces F1 by 5.43/4.80/4.92 points, indicating its role in capturing cross-hop dependencies and propagation relations for better feature aggregation. Removing task adaptation causes larger drops of 6.80/6.45/7.28 points, highlighting the importance of task-aware reweighting for feature selection and generalization across shots. Finally, regarding the heteroscedastic uncertainty model, its performance is also quite evident. When removing the uncertainty modeling component, the F1-scores under different K-shot settings drop to 77.61/81.06/82.19, suggesting that uncertainty modeling suppresses hard/noisy samples, stabilizes training, and improves robustness in low-data, high-noise scenarios.

V. CONCLUSION

In this paper, we present an uncertainty-aware heterogeneous graph meta-learning framework for few-shot IoT malicious traffic detection, namely UA-MetaHGN. UA-MetaHGN first constructs a heterogeneous interaction graph to explicitly model the structural dependencies among network flows, IP addresses, and ports, thereby capturing the fine-grained traffic–resource relationships that are often overlooked by sample-independent detectors. Then, it adopts a hybrid-attention hierarchical representation learning scheme to learn discriminative graph representations by aggregating relation-aware, multi-hop, and semantic-level information, which enables the model to better exploit high-order dependencies in complex IoT traffic. Furthermore, we integrate task-adaptive initialization and dynamic task weighting into the meta-learning procedure, jointly leveraging loss discrepancy, heteroscedastic uncertainty, and Bayesian-style regularization to facilitate fast adaptation and robust generalization under scarce labels and evolving attack patterns. Extensive experiments on the ToN-IoT benchmark demonstrate that UA-MetaHGN consistently outperforms state-of-the-art baselines in few-shot settings with varying support sizes (3/5/10-shot) and also achieves superior performance under standard supervised learning; ablation studies further verify the effectiveness of the proposed hybrid-attention aggregation and uncertainty-aware task reweighting.

TABLE III
ABLATION RESULTS UNDER DIFFERENT K-SHOT SETTINGS.

Model Variants	3-shot F1-score	5-shot F1-score	10-shot F1-score
Full Model	**89.66**	**92.16**	**94.65**
w/o MultiHop Attention	84.23	87.36	89.73
w/o Task Adaptation	82.86	85.71	87.37
w/o Uncertainty Method	77.61	81.06	82.19

ACKNOWLEDGMENT

This work was supported in part by the Shandong Provincial Natural Science Foundation under Grant No.ZR2023LZH011, the Taishan Scholar Program of Shandong Province in China under Grant No.TSQN202312230.

REFERENCES

[1] E. Lee, Y.-D. Seo, S.-R. Oh, and Y.-G. Kim, "A survey on standards for interoperability and security in the internet of things," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1020–1047, 2021.

[2] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[3] R. Kumar, M. Swarnkar, G. Singal, and N. Kumar, "Iot network traffic classification using machine learning algorithms: An experimental analysis," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 989–1008, 2021.

[4] Y. Luo, M. He, and X. Wang, "Analyzing the semantic structure of network flow: a threat detection method with independent generalization capabilities," *IEEE Transactions on Network Science and Engineering*, 2024.

[5] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[6] G. Wang, L. Zhang, J. Wang, T. Wo, X. Wang, and C. Hu, "Logad: A multi-feature fusion approach for log anomaly detection," in *2025 IEEE International Conference on Joint Cloud Computing (JCC)*. IEEE, 2025, pp. 83–90.

[7] Q. Mao, X. Lin, W. Xu, Y. Qi, X. Su, G. Li, and J. Li, "Fecograph: Label-aware federated graph contrastive learning for few-shot network intrusion detection," *IEEE Transactions on Information Forensics and Security*, 2025.

[8] H. Li, X. Fu, Y. Zhu, R. Yang, and C. Li, "Metaiot: Few shot malicious traffic detection in internet of things networks based on hin," in *2023 IFIP Networking Conference (IFIP Networking)*. IEEE, 2023, pp. 1–9.

[9] D. He, J. Cui, X. Wang, G. Song, Y. Huang, and L. Wu, "Dual enhanced meta-learning with adaptive task scheduler for cold-start recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[10] Q. Chen, S. Li, Y. Liu, S. Pan, G. I. Webb, and S. Zhang, "Uncertainty-aware graph neural networks: A multihop evidence fusion approach," *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[11] J. Gu and S. Lu, "An effective intrusion detection approach using svm with naïve bayes feature embedding," *Computers & Security*, vol. 103, p. 102158, 2021.

[12] Z. Li, Z. Cheng, T. Zang, and Y. Li, "Mtcd-model: A two-layer model for malicious traffic classification and detection based on hierarchical feature learning," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[13] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-graphsage: A graph neural network based intrusion detection system for iot," *arXiv preprint arXiv:2103.16329*, 2021.

[14] 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.

[15] Z. La, J. Shen, Y. Song, S. Tian, and W. Gong, "Ml-han: Multi-level heterogeneous graph attention network for representation learning with semantic diversity via feature-node-semantic attention," *Neurocomputing*, p. 131962, 2025.

[16] S. Li, J. Gong, S. Ke, and S. Tang, "Graph transformer-based heterogeneous graph neural networks enhanced by multiple meta-path adjacency matrices decomposition," *Neurocomputing*, vol. 629, p. 129604, 2025.

[17] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3540–3552, 2020.

[18] M. Pawlicki, R. Kozik, and M. Choraś, "Improving siamese neural networks with border extraction sampling for the use in real-time network intrusion detection," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[19] Y. Wang, C. Huang, M. Li, Q. Huang, X. Wu, and J. Wu, "Ag-meta: Adaptive graph meta-learning via representation consistency over local subgraphs," *Pattern Recognition*, vol. 151, p. 110387, 2024.

[20] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. Den Hartog, "Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, 2021.

[21] S. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable intrusion detection system for iot: A deep transfer learning based approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1006–1017, 2022.

[22] G. Duan, H. Lv, H. Wang, and G. Feng, "Application of a dynamic line graph neural network for intrusion detection with semisupervised learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 699–714, 2022.

[23] Z. Jiang, J. Li, Q. Hu, W. Meng, W. Pedrycz, and Z. Su, "Scalable graph-aware edge representation learning for wireless iot intrusion detection," *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 26 955–26 969, 2024.