Adversarial Optimization Scheme for Threat Detection Based on Hierarchical Oracle Supervision

Yiqing Luo , Mingshu He* , Xiaojuan Wang
Beijing University of Posts and Telecommunications
{luoyiqing, hemingshu, wj2718}@bupt.edu.cn

Abstract

Conventional threat detection methodologies encounter substantial limitations in real-world complex network settings, owing to their dependence on single-dimensional analysis, vulnerability to adversarial perturbations, and propensity for overfitting. This study introduces an adversarial training optimization framework that incorporates hierarchical label encoding and prompt learning, designed to enhance model robustness and generalization in threat detection. The framework first establishes a hierarchical taxonomy of attack scenarios and types, leveraging a graph attention network to encode semantic and structural dependencies among labels. Subsequently, the classification task is reformulated as a masked language modeling problem through prompt learning, enabling effective semantic alignment. Furthermore, a novel adversarial training mechanism is proposed, which utilizes local hierarchical information as a potent regularization signal. Within a game-theoretic architecture comprising a generator, an encoder, and a discriminator, the encoder is steered to integrate authentic hierarchical priors and produce high-fidelity oracle representations. This process encourages the generator to implicitly assimilate sample-specific hierarchical knowledge during adversarial learning, thereby improving the model's resilience to noisy inputs and its capacity to detect infrequent attacks. Evaluation results show that this method achieves an accuracy of 0.997–1.000 in complex mixed scenarios for scenario detection and 0.998–0.999 for attack category detection, while exhibiting strong robustness against interference.

1 Introduction

Network security is intrinsically linked to operational continuity and public safety, underscoring the critical importance of deploying threat detection systems that can effectively counteract sophisticated cyberattacks. Nevertheless, current threat detection approaches continue to encounter substantial limitations. A primary issue lies in the unidimensional nature of threat analysis. Predominant approaches frequently reduce threat detection to a flattened multiclass classification problem [Zhao *et al.*, 2025b; Shen *et al.*, 2025; Zhao *et al.*, 2025a], wherein only the attack type of network flow is identified, while the essential contextual dimension pertaining to attack scenarios is neglected. This results in fragmented threat intelligence and complicates forensic traceability. A further challenge involves the brittleness of model decision boundaries [Ennaji *et al.*, 2025; He *et al.*, 2023; Yan *et al.*, 2025]. Real-world network settings are characterized by considerable noise, and emergent threat variants [Ben Atitallah *et al.*, 2025], imposing stringent demands on model robustness. Conventional models are susceptible to overfitting to superficial patterns in training data [Agarwal *et al.*, 2024; Musthafa *et al.*, 2024b; Musthafa *et al.*, 2024a] and exhibit limited stability at their decision boundaries when confronted with noise perturbations, flow shifts, or novel threat mutations, thereby undermining their practical applicability and reliability in dynamic adversarial environments.

This paper proposes a novel framework for threat detection that integrates hierarchical labels, prompt learning, and adversarial training. The method first models the hierarchical relationships between attack scenarios and types using graph networks, enabling fine-grained multi-label threat perception and traceability analysis. It then employs prompt learning to reframe the classification task as a mask prediction task familiar to pre-trained models, thereby improving knowledge transfer efficiency. Finally, an adversarial training mechanism supervised by hierarchical labels is introduced, forcing the model to learn robust features resistant to interference, which effectively enhances the generalization capability and stability of the system in dynamic adversarial environments.

The main contributions of this paper are as follows.

- We introduce a multi-label classification framework designed to facilitate the collaborative analysis of security scenarios and attacks. By leveraging graph networks to model their hierarchical dependencies, this method not only categorizes attack types but also deduces the corresponding scenarios, thereby supplying essential contextual information for threat attribution and supporting more granular threat awareness.
- A prompt learning-based method for classifier construction has been proposed, which reframes the classification task as a masked language modeling problem com-

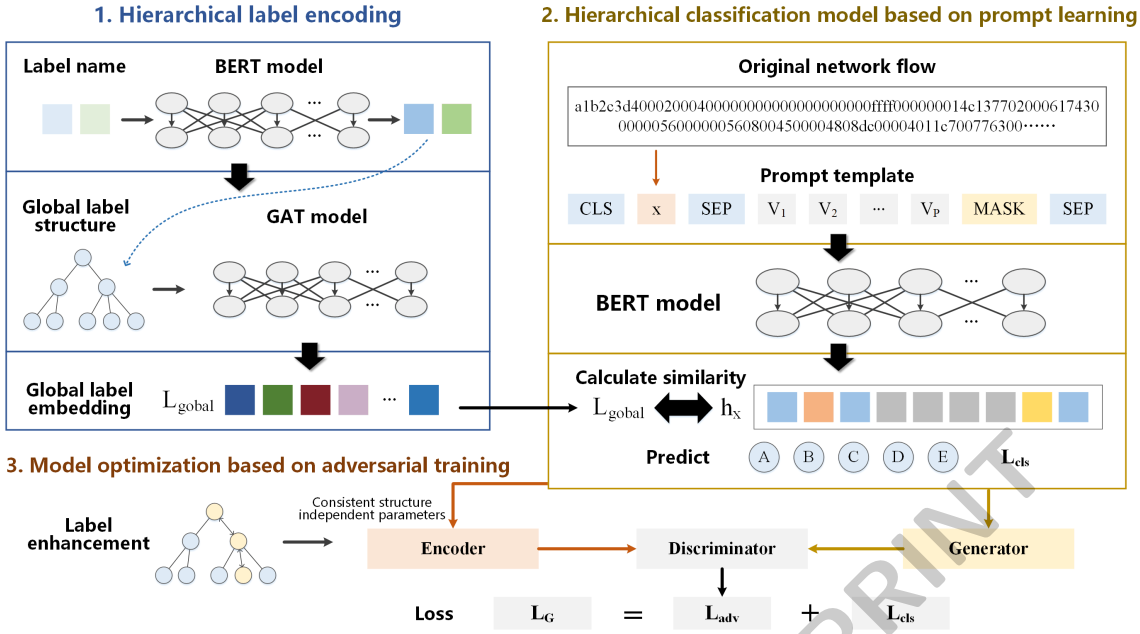


Figure 1: Model architecture.

patible with pre-trained models. By optimizing continuous prompt embeddings, this method enables efficient and stable transfer of pre-trained knowledge to the domain of threat detection.

- We introduce an adversarial training strategy that leverages genuine hierarchical information as a robust supervisory signal. This method formulates a game-theoretic framework comprising a generator, an encoder, and a discriminator, wherein instance-specific hierarchical label information acts as a regularizer to directly steer the model toward learning feature representations consistent with hierarchical constraints. In contrast to conventional methods, our design explicitly integrates structural knowledge of the hierarchy, thus improving the model's resilience to noise, adversarial perturbations, and unforeseen threats.

2 Problem and Threat Model Assumptions

2.1 Problem Definition

Current threat detection methods face two core challenges. Firstly, threat analysis suffers from a lack of dimensionality. Flat, single-label classification [Zhao *et al.*, 2025b; Shen *et al.*, 2025; Zhao *et al.*, 2025a] leads to isolated threat intelligence, making it difficult to correlate attack tactics, stages, and target environments, thereby hindering in-depth traceability and response decisions. Secondly, model decision boundaries are fragile [Ennaji *et al.*, 2025; He *et al.*, 2023; Yan *et al.*, 2025]. Traditional models are prone to overfitting superficial features of the training data [Agarwal *et al.*, 2024; Musthafa *et al.*, 2024b; Musthafa *et al.*, 2024a], resulting in significant performance degradation when confronted with input perturbations or unknown threat variants, and making them ineffective against adversarial attacks or incapable of adapting to novel threats.

2.2 Threat Model Assumption

This method assumes that the attacker possesses the capability to launch multi-stage persistent attacks, with the objectives of compromising service availability, data confidentiality/integrity, or escalating privileges, while the target network contains security flaws such as unpatched vulnerabilities and weak password configurations.

In this setting, given a network flow dataset $D = \{x_i, y_i\}_{i=1}^{N}$, where $x_i \in R^d$ is the feature vector and $y_i \in \{1, 2, \ldots, K\}$ is the threat label. The objective of the task is to learn a robust classification function $f : R^d \to \{1, 2, \ldots, K\}$ that can maintain prediction stability and accurately output the threat category $y = f(x)$ when confronted with malicious flow x' crafted by attackers through adversarial examples.

3 Method

The proposed method comprises three core components: hierarchical label encoding, a prompt learning-driven hierarchical classification model, and adversarial training-based optimization, as depicted in Fig.1.

3.1 Hierarchical Label Encoding

This module employs a progressive encoding strategy to transform discrete labels into distributed vector representations that concurrently capture lexical semantics, global hierarchical relationships, and local contextual information.

First, we developed a hierarchical labeling framework. In contrast to conventional threat detection approaches, which typically categorize attacks by type alone, our system not

only incorporates detailed attack classifications but also introduces attack scenarios as higher-level labels, thereby constructing a more thorough descriptive framework for characterizing threats, as depicted in Fig.2.

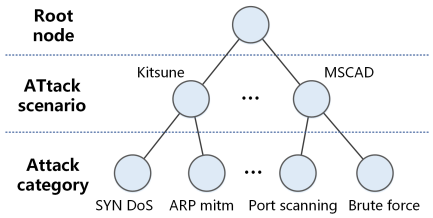


Figure 2: Label system.

Subsequently, we need to convert the textual label names into a numerical representation that can be processed by machine learning models. Labels initially exist as natural language entities. We utilize the pre-trained language model BERT to encode the semantic information of the labels, transforming each label name y into a distributed representation. Specifically, each label name y is first processed by BERT's WordPiece tokenizer:

$$t_y = [t_1, t_2, \cdots, t_k] = Tokenizer_{BERT}(y). \tag{1}$$

In the formulation, t_y represents the subword sequence obtained after tokenization, with t_k denoting the k-th subword token. For instance, the label "port scanning" is decomposed into the subword sequence ["port", "scanning"]. This tokenization strategy effectively addresses out-of-vocabulary compound words and domain-specific terminology, thereby preserving the full semantic content of the label. The resulting token sequence is subsequently mapped to vector representations via BERT's embedding layer:

$$\begin{aligned} e^{(k)} = {} & W_{token} \cdot onehot(t_k) + W_{pos} \cdot onehot(k) \\ & + W_{seg} \cdot onehot(0). \end{aligned} \tag{2}$$

W_{token} is the token embedding matrix, W_{pos} is the position embedding matrix, and W_{seg} is the segment embedding matrix. The embedding layer integrates lexical semantics, positional information, and segment type to generate a context-independent initial representation for each token.

The final semantic representation of the label is generated using an average pooling strategy:

$$l_y^{sem} = \frac{1}{K} \sum_{k=1}^{K} e^{(k)}. \tag{3}$$

l_y^{sem} is the semantic representation vector of the label, and K is the number of tokens in the sequence. Average pooling considers the contribution of each subword equally, and is insensitive to variations in sequence length, making it suitable for handling label names of different lengths.

However, l_y^{sem} only contains the semantic meaning of the label itself and fails to capture its position within the hierarchical structure. To address this, we introduce a Graph Attention Network (GAT), treating it as a graph for encoding. The GAT employs a message-passing mechanism, allowing the representation of each label node to aggregate information from its neighbors (parent and child classes). Specifically, for each label node i in the label hierarchy, its global representation is iteratively updated through a multi-layer graph attention mechanism:

$$l_{global}^{(l)} = \sigma\Big(\sum_{j \in N(i) \cup \{i\}} \alpha_{ij}^{(l)} W^{(l)} l_{global_j^{(l-1)}} \Big). \tag{4}$$

In this context, $\alpha_{ij}^{(l)}$ represents the attention weight at the l-th layer, $N(i)$ denotes the set of neighboring nodes of node i, $W^{(l)}$ is the transformation matrix, and σ is the activation function. This process ensures that semantically similar or hierarchically adjacent labels are positioned closely in the vector space, thereby enhancing the discriminative power and semantic richness of the label representations. Furthermore, the hierarchical encoding of GAT allows each label's representation to incorporate not only its own lexical semantics but also the structural contextual information from the global hierarchy. For example, "port scanning" and "uploading injection" both belonging to the parent category of the Edge-IIoTset dataset scenario, will share certain common features in their representations after GAT encoding, while retaining their distinct characteristics.

After propagation through the L layers, we obtain the final global label embedding matrix L_{global}. For the complete label system comprising N tags, we derive the hierarchical representation $l_{global}^i \in \mathbb{R}^d$ for each tag through GAT encoding, where $i \in \{1, 2, \ldots, N\}$ is the tag index and d is the embedding dimension. These vectors are combined row-wise to construct the global tag embedding matrix:

$$L_{global} = \begin{bmatrix} l_{global}^1 \\ l_{global}^2 \\ \vdots \\ l_{global}^N \end{bmatrix} \in \mathbb{R}^{N \times d} \tag{5}$$

This matrix will serve as the learnable label representation in the next section's classification model, used to compute the semantic similarity between input features and all labels.

3.2 Hierarchical Classification Model Based on Prompt Learning

Upon acquiring the hierarchical vector representations of the labels, the central objective of this module is to construct a classification model capable of mapping raw flow into the same semantic space. Hierarchical classification is subsequently achieved by computing the similarity between the input data and the label representations.

We first fill the original flow data x into a carefully designed prompt template T to construct the final input sequence $T(x)$ for the model, which can be described as:

$$T(x) = [CLS]x[SEP][V_1][V_2] \cdots [V_P][MASK][SEP]. \tag{6}$$

In this framework, $[V_1], [V_2], \cdots, [V_P]$ represent a set of learnable continuous virtual tokens. These tokens are not as-

sociated with any natural language vocabulary but are optimized through gradient descent during training. They function as task-specific contextual cues, enabling the pre-trained model to better adapt to the downstream threat classification task. The placement of the $[MASK]$ token is critical within the overall template, as the model is tasked with generating predictions at this specific position.

Then, the BERT model is used to encode the prompt-processed sequence $T(x)$ into a sequence of hidden states:

$$H = BERT(T(x)). \tag{7}$$

We take the hidden state corresponding to the $[CLS]$ token or perform a pooling operation on the entire sequence to obtain the aggregated representation $h_x \in \mathbb{R}^d$ of the data x:

$$h_x = Pooler(H_{[CLS]}). \tag{8}$$

This representation effectively integrates raw flow data with contextual information derived from learnable prompts, while the BERT model captures the deep semantic features of the input via its multi-layer Transformer encoder.

Classification decisions are made by calculating the similarity between the text representation h_x and each label vector in the global label embedding matrix L_{global} generated in the previous section. Specifically, we employ the dot product operation to measure the proximity in the semantic space.

$$s_i = h_x \cdot l^i_{global}. \tag{9}$$

l^i_{global} represents the label vector of the i-th label in the global label embedding matrix L_{global}. The similarity score is then transformed into the predicted probability for each label y using the sigmoid function:

$$P(y_i|x) = \sigma(s_i + b_i) = \frac{1}{1 + e^{-(h_x \cdot l^i_{global} + b_i)}}. \tag{10}$$

where b_i is the label-specific bias term.

The model learns its parameters by minimizing the binary cross-entropy loss L_{cls} between the predicted probabilities and the true labels, allowing the predicted probability $P(y|x)$ to gradually approximate the distribution of the true labels.

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log P(y_i|x) + (1-y_i) log(1 - P(y_i|x))]. \tag{11}$$

During the inference stage, we employ an optimized single-model architecture that retains only the trained generator to ensure efficient forward propagation. For an input sample x_{test}, it is initially formatted via a prompt template $T(x_{\text{test}})$, incorporating pre-trained continuous prompt tokens and the target $[MASK]$ position. The model then computes the hidden state $h_{[\text{MASK}]}$ at the $[MASK]$ position to serve as the data representation, and performs multi-label prediction by measuring its similarity against a global label embedding matrix L_{global}.

$$\hat{y}_i = \mathbb{I}[P(y_i|x_{test}) > \tau]. \tag{12}$$

The final decision is made using a threshold strategy of τ=0.5, and hierarchical consistency constraints are applied to ensure the prediction results align with the label hierarchy.

3.3 Model Optimization Based on Adversarial Training

The principal aim of this module is to utilize sample-specific, authentic hierarchical information as a robust supervisory signal to regularize the base model, thereby improving its resilience and generalization performance in the presence of noisy perturbations and out-of-distribution data.

The essence of adversarial training is to enable the model to learn through "adversarial gaming" by generating indistinguishable adversarial samples, thereby enhancing the smoothness and stability of the decision boundary. In this module, the adversarial training framework consists of three components: a generator, an encoder, and a discriminator.

The generator, implemented as a hierarchical classification model parameterized by θ_G and trained in the preceding section, processes flow data x to generate a latent representation h_{mix} that incorporates both semantic and prompt-related information. Throughout adversarial training, the generator aims to produce representations that are statistically indistinguishable from the real data distribution, thereby attempting to mislead the discriminator.

The encoder comprises a network architecture analogous to that of the generator, yet endowed with independent parameters. Its distinctive characteristic is the incorporation of both the input data x and the label representation $\hat{L}$, derived via local hierarchical perception, to produce a high-quality oracle representation $\hat{h}_{mix}$ that embeds the true local hierarchical prior. This representation effectively integrates authentic local hierarchical information, thereby furnishing the generator with a well-defined learning objective and a robust regularization signal.

The discriminator is a binary classification neural network tasked with determining whether an input representation originates from the generator or the encoder. It outputs a scalar probability value indicating the likelihood that the input representation is considered to come from the encoder (i.e., the oracle representation).

To generate high-quality oracle representations $\hat{h}_{mix}$, we first enhance the label representations by incorporating sample-specific presence information:

$$\hat{l}_y = l^y_{global} + \mathbb{I}(y \in Y) \cdot e_{present} + \mathbb{I}(y \notin Y) \cdot e_{absent}. \tag{13}$$

Among them, $e_{present}$ and e_{absent} are learnable vectors, and $\mathbb{I}(\cdot)$ is an indicator function. This step assigns each label a clear semantic meaning (presence or absence) within the context of the current sample. The encoder utilizes this set of enhanced label representations $\hat{L}$ and the input data x to generate the final oracle representation $\hat{h}_{mix}$. This process ensures that h_{mix} is a high-quality reference representation that incorporates genuine local hierarchical knowledge.

Adversarial training is performed through alternating optimization. First, the parameters of the generator G and the encoder E are fixed, while the parameters θ_D of the discriminator D are updated. The goal of the discriminator is to maximize its discriminative ability, and its loss function is:

$$L_D = -\mathbb{E}[log D(\hat{h}_{mix})] - \mathbb{E}[log(1 - D(h_{mix}))]. \tag{14}$$

	Attack scenario detection					Attack category detection				
Dataset	ACC	PRE	REC	MaF1	MiF1	ACC	PRE	REC	MaF1	MiF1
Kitsune+ IIoTset	0.999	0.999	0.999	0.999	0.999	0.998	0.996	0.978	0.830	0.987
Kitsune+TFC2016	0.997	0.997	0.997	0.995	0.997	0.998	0.984	0.984	0.816	0.984
Kitsune+MSCAD	1.000	1.000	1.000	1.000	1.000	0.999	0.996	0.996	0.906	0.996
IIoTset+TFC2016	0.999	0.999	0.999	0.999	0.999	0.999	0.998	0.998	0.995	0.998
IIoTset+MSCAD	1.000	1.000	1.000	1.000	1.000	0.999	0.997	0.997	0.927	0.997
TFC2016+MSCAD	0.999	0.999	0.999	0.999	0.999	0.999	0.996	0.995	0.918	0.995
Kitsune+ IIoTset+TFC2016	1.000	1.000	1.000	1.000	1.000	0.999	0.997	0.990	0.863	0.993
Kitsune+ IIoTset+MSCAD	0.999	0.999	0.999	0.999	0.999	0.998	0.988	0.988	0.857	0.988
IIoTset+TFC2016+ MSCAD	0.998	0.998	0.998	0.998	0.998	0.999	0.996	0.995	0.951	0.996
all datasets	0.999	1.000	0.999	0.999	0.999	0.999	0.994	0.994	0.892	0.994

Table 1: The detection effect of attack scenarios and attack categories using this method

Here, the discriminator is trained to correctly distinguish between the oracle representations from the encoder and the "generated" representations from the generator.

Next, with the parameters of the discriminator D and the encoder E held fixed, the parameters θ_G of the generator G are updated. The generator is optimized under a dual objective: minimizing the classification loss L_{cls} to preserve baseline classification performance, and minimizing the adversarial loss L_{adv}, which promotes the generation of representations h_{mix} that are indistinguishable from the oracle representations $\hat{h}_{mix}$, thereby misleading the discriminator. The adversarial loss is defined as:

$$L_{adv} = -\mathbb{E}[logD(h_{mix})]. \tag{15}$$

Therefore, the total loss of the generator is:

$$L_G = L_{cls} + \alpha L_{adv}. \tag{16}$$

Among them, α is a hyperparameter used to balance the weights of the two loss terms.

Through this adversarial training process, the generator is compelled to learn representations that are distributionally aligned with those of the oracle model. Consequently, the internal feature representations of the generator implicitly capture and encode the true hierarchical structure of the data. This approach yields several key benefits: (1) It introduces a strong regularization effect by incorporating hierarchical information as a powerful inductive bias, which mitigates overfitting on limited training data, particularly for rare attack classes; (2) It promotes smoother decision boundaries, encouraging the model to learn a more stable decision function in the vicinity of the boundary between genuine and adversarial examples, thus improving robustness to input variations; (3) It facilitates a form of knowledge distillation, whereby the oracle hierarchical knowledge embedded in the encoder is effectively transferred to the generator (the primary model).

4 Evaluation

4.1 Evaluation Setup

Dataset

We conducted a comprehensive evaluation on four public datasets: Kitsune (for IoT and Industrial Internet of Things environments, including 9 types of attacks), Edge-IIoTset (for large-scale IoT scenarios, including 14 types of attacks), MSCAD (for multi-step network attack scenarios, including 5 types of attacks), and USTC-TFC2016 (for encrypted flow detection scenarios, including 10 type of attack).

Parameter Setting

Unless otherwise stated, all experimental results presented in this study are derived under the following parameter configurations: The AdamW optimizer is employed for parameter optimization during training, with a base learning rate of 310^{-5}. In adversarial training, the weight of the adversarial loss is set to 1.0. The discriminator is trained with a learning rate of 1×10^{-4}, updated once every five generator steps, and a gradient penalty coefficient of 10.0 is applied. During inference, a classification threshold of 0.5 is used. All datasets are partitioned into training, validation, and test sets with a ratio of 64:16:20, respectively.

Evaluating Metrics

We used accuracy, precision, recall, macro F1-score(MaF1) and micro F1-score(MiF1) to evaluate the classification ability of the proposed method.

4.2 Overall Comparative Analysis

To comprehensively evaluate the effectiveness of the proposed method, we designed cross-validation experiments with 10 combinations ranging from 2 to 4 scenarios on four public datasets. As summarized in Table 1, the proposed method exhibits robust performance in both attack scenario and attack category detection under varying scenario configurations. This consistency suggests that hierarchical label encoding and prompt learning effectively capture discriminative feature representations at the macro-scenario level, thereby affirming the distinctiveness of scenario-specific fingerprints embedded within network flow.

Furthermore, based on the fused dataset of four scenarios, Kitsune-IIoTset-TFC2016-MSCAD, we compared this method with other advanced threat detection approaches and evaluated its detection performance across different attack categories, as shown in Table 2. In contrast to approaches that depend on statistical features, our method employs prompt learning and hierarchical label encoding to more effectively model flow semantics and contextual relationships, thereby

Method	Feature	Classification model	ACC	PRE	REC	MaF1
Kitsune [Mirsky *et al.*, 2018]	statistical features	Autoencoder	0.9812	0.9456	0.9321	0.7988
Whisper [Fu *et al.*, 2021]	frequency domain features	Clustering	0.9735	0.9287	0.9105	0.7742
Rosetta [Xie *et al.*, 2023]	Temporal and packet length features	Contrastive learning	0.9889	0.9618	0.9523	0.8015
ERFS [Luo *et al.*, 2025]	Semantic features between bytes and flows	Message passing mechanism	0.9921	0.9725	0.9688	0.8436
SmartDetector [Shen *et al.*, 2025]	Statistical based contextual features	Contrastive learning	0.9948	0.9831	0.9769	0.8710
our	**prompt-based learning and Hierarchical Label Encoding**	**Adversarial training**	**0.9996**	**0.9947**	**0.9942**	**0.8924**

Table 2: Overall comparative analysis of this method with other advanced approaches

Ablation strategy	ACC	PRE	REC	MaF1
Remove GAT	0.9981	0.9875	0.9850	0.8326
Remove prompt -based learning	0.9978	0.9843	0.9821	0.8195
Remove adversarial training	0.9989	0.9905	0.9888	0.8510
All	0.9996	0.9947	0.9942	0.8924

Table 3: Local ablation evaluation of this method

achieving superior generalization. Unlike semantic feature-based techniques, the generative adversarial training framework incorporated in our method substantially improves its capacity to address class imbalance, yielding notable advantages in the identification of infrequent attack types.

4.3 Local Ablation Analysis

In this section, we performed ablation studies by systematically excluding the GAT-based label hierarchy encoding, prompt learning, and adversarial training modules from the complete model, in order to assess the individual contribution of each component to attack category detection on the integrated four-scenario dataset. As summarized in Table 3, the adversarial training module provided the most substantial improvement for recognizing rare attack categories, with the GAT-based label hierarchy encoding also contributing significantly. Optimal performance was achieved when all three components were utilized together.

4.4 Generalization and Induction Performance Analysis

Generalizability

To assess the model's generalization capacity, cross-testing was performed across datasets encompassing diverse network environments, topologies, and service scenarios. As illustrated in Table 4, the model attains a detection accuracy exceeding 80% in the majority of cross-scenario evaluations, indicating its ability to capture common underlying features present in disparate scenario-based datasets and thereby demonstrating a measurable degree of generalization.

Training dataset	Testing dataset	ACC	PRE	REC	MaF1
Kitsune IIoIset	TFC2016 MSCAD	0.872	0.845	0.821	0.833
Kitsune TFC2016	IIoTset MSCAD	0.891	0.863	0.852	0.857
Kitsune MSCAD	IIoTset TFC2016	0.885	0.851	0.841	0.846
IIoTset TFC2016	Kitsune MSCAD	0.906	0.882	0.875	0.878
IIoTset MSCAD	Kitsune TFC2016	0.898	0.871	0.863	0.867
TFC2016 MSCAD	Kitsune IIoTset	0.864	0.832	0.815	0.824

Table 4: The detection F1 value of this method when using different datasets as training and testing data sources

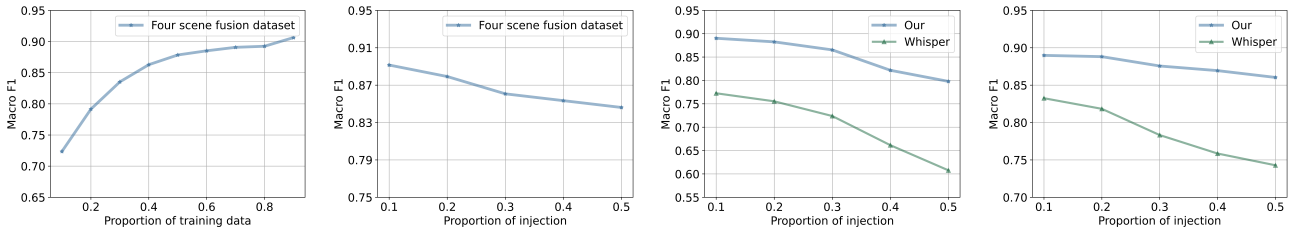
Inductive

To assess the adaptability of the proposed method under inductive learning conditions, we systematically varied the proportion of training data using a multi-scenario integrated dataset, thereby simulating real-world threat detection environments with differing levels of data availability. As illustrated in Fig.3(a), this method exhibits notable robustness in data-scarce inductive settings, sustaining an F1-score exceeding 70% on the integrated dataset even when trained on only 10% of the available data.

4.5 Robust Performance Analysis

Evaluation of Model Robustness Based on Low Rate Strategy

To assess the model's robustness against low-rate attacks, this section emulates an attacker's low-rate strategy by introducing delay sequences—generated via a Poisson process featuring a base delay of 5 seconds superimposed with ±30% random jitter into the network flow at an injection ratio ranging from 0.1% to 0.5%. These sequences exhibit long-period and low-density statistical properties. As illustrated in Fig.3(b), this method maintains strong robustness under such conditions. Although low-rate attacks perturb the temporal distribution of network flows, they leave semantic features of



(a) Scarcity of training data (b) Low rate strategy (c) Temporal noise interference (d) Payload filling interference

Figure 3: Comparison of detection performance of this method under different conditions

payloads and flow-level interaction logic unaltered, thereby preserving the effectiveness of hierarchical label encoding and prompt learning-based semantic representations. Furthermore, through adversarial training, the model learns intrinsic features that are robust to temporal disturbances, thereby improving detection stability against low-rate attacks.

Evaluation of Model Robustness Based on Temporal Noise Interference

This section evaluates this method's resilience against timing-based interference by simulating an attacker's strategy and comparing its performance with existing robustness approaches, such as Whisper[Fu *et al.*, 2021]. In the experiments, random delays drawn from a Gaussian distribution (mean = 0.5 s, standard deviation = 0.2 s) were introduced into the network flow, with the injection rate incrementally raised from 0.1% to 0.5%. As illustrated in Fig.3(c), this method exhibits significantly greater robustness to timing noise compared to Whisper. The latter depends on frequency-domain features that are highly susceptible to disruptions in temporal structure, rendering it prone to performance degradation under noise. By contrast, the proposed method utilizes hierarchical semantic encoding coupled with adversarial training, which facilitates the extraction of deep features that are inherently less sensitive to temporal variations, thus ensuring more consistent detection accuracy in noisy conditions.

Evaluation of Model Robustness Based on Payload Filling Interference

This section simulates an adversarial payload-padding interference strategy and compares this method with the Whisper[Fu *et al.*, 2021]. The strategy involves inserting cryptographically secure random hexadecimal data into the mid-section of packet payloads, constituting 20% of the original payload length (injection ratio: 0.1%–0.5%), to obfuscate flow features while dynamically updating protocol headers to ensure compatibility. As illustrated in Fig.3(d), this method exhibits significantly superior detection performance over the Whisper, which relies on frequency-domain analysis, under conditions of payload-padding interference. The introduction of random hexadecimal data disrupts the frequency-domain feature structure essential to Whisper's operation. In contrast, this method leverages prompt learning and hierarchical encoding to extract deep semantic features that remain robust to content-level perturbations. Furthermore, adversarial training enhances the model's robustness and adaptability in the presence of payload obfuscation.

	Loss			
	R_l=0	R_l=0.05	R_l=0.1	R_l=0.2
	0.9996	0.9942	0.9808	0.9635
	Retransmission			
Four scene fusion dataset	R_r=0	R_r=0.05	R_r=0.1	R_r=0.2
	0.9996	0.9975	0.9831	0.9642
	Out-of-order			
	R_o=0	R_o =0.05	R_o =0.1	R_o =0.2
	0.9996	0.9925	0.9776	0.9350

Table 5: The detection accuracy of this method under various network phenomena

Evaluation of Model Robustness Based on Network Phenomena

To evaluate the impact of network-induced phenomena on the model, this section utilizes the scapy library to manipulate PCAP packets by deleting, duplicating, and reordering packets to simulate three typical network anomalies: packet loss (R_l), retransmission (R_r), and packet reordering (R_o), respectively. As shown in Table 5, packet reordering has the most significant impact on model performance, as it disrupts flow-level temporal dependencies and protocol state machines, to which the hierarchical semantic encoding and prompt learning method employed here are relatively sensitive. In contrast, packet loss and retransmission mainly cause information loss or redundancy but do not alter the relative timing among the remaining packets, thus resulting in a comparatively smaller impact.

5 Conclusion

This method introduces an adversarial training mechanism that utilizes real local hierarchical information as a strong supervisory signal, compelling the model to learn feature representations that are insensitive to noise and adversarial perturbations. Evaluation results demonstrate that this method achieves an accuracy exceeding 0.998 in detecting scenes and attack categories under complex conditions, while exhibiting strong robustness against interference.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62402053 and Grant 62227805, and in part by the Fundamental Research Funds for the Central Universities under Grant 2025KYQD17(BUPT).

References

[Agarwal *et al.*, 2024] Lalit Agarwal, Bhavnesh Jaint, and Anup K. Mandpura. Reducing overfitting in deep learning intrusion detection for power systems with ctgan. *Chaos, Solitons & Fractals*, 188:115603, 2024.

[Ben Atitallah *et al.*, 2025] Safa Ben Atitallah, Maha Driss, Wadii Boulila, and Anis Koubaa. Securing industrial iot environments: A fuzzy graph attention network for robust intrusion detection. *IEEE Open Journal of the Computer Society*, 6:1065–1076, 2025.

[Ennaji *et al.*, 2025] Sabrine Ennaji, Fabio de Gaspari, Dorjan Hitaj, Alicia Kbidi, and Luigi Vincenzo Mancini. Adversarial challenges in network intrusion detection systems: Research insights and future prospects. *IEEE Access*, 13:148613–148645, 2025.

[Fu *et al.*, 2021] Chuanpu Fu, Qi Li, Meng Shen, and Ke Xu. Realtime robust malicious traffic detection via frequency domain analysis. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3431–3446, New York, NY, USA, 2021. Association for Computing Machinery.

[He *et al.*, 2023] Ke He, Dan Dongseong Kim, and Muhammad Rizwan Asghar. Adversarial machine learning for network intrusion detection systems: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 25(1):538–566, 2023.

[Luo *et al.*, 2025] Yiqing Luo, Mingshu He, and Xiaojuan Wang. Erfs: Efficient feature graph representation for intrusion detection based on flow semantic association. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1, 2025.

[Mirsky *et al.*, 2018] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *Proceedings of the 25th Annual Network and Distributed System Security Symposium, NDSS 2018*. The Internet Society, 2018.

[Musthafa *et al.*, 2024a] Muhammad Bisri Musthafa, Samsul Huda, Md. Arshad Ali, Yuta Kodera, and Yasuyuki Nogami. Evaluation of ids model by improving accuracy and reducing overfitting using stacking lstm. In *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–5, 2024.

[Musthafa *et al.*, 2024b] Muhammad Bisri Musthafa, Samsul Huda, Yuta Kodera, Md. Arshad Ali, Shunsuke Araki, Jedidah Mwaura, and Yasuyuki Nogami. Optimizing iot intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques. *Sensors*, 24(13), 2024.

[Shen *et al.*, 2025] Meng Shen, Jinhe Wu, Ke Ye, Ke Xu, Gang Xiong, and Liehuang Zhu. Robust detection of malicious encrypted traffic via contrastive learning. *IEEE Transactions on Information Forensics and Security*, 20:4228–4242, 2025.

[Xie *et al.*, 2023] Renjie Xie, Yixiao Wang, Jiahao Cao, Enhuan Dong, Mingwei Xu, Kun Sun, Qi Li, Licheng Shen, and Menghao Zhang. Rosetta: Enabling robust tls encrypted traffic classification in diverse network environments with tcp-aware traffic augmentation. In *Proceedings of the ACM Turing Award Celebration Conference - China 2023*, ACM TURC '23, page 131–132, New York, NY, USA, 2023. Association for Computing Machinery.

[Yan *et al.*, 2025] Wenhao Yan, Ning An, Wei Qiao, Weiheng Wu, Bo-Sian Jiang, Yuling Liu, Zhigang Lu, and Junrong Liu. Sentient: Multi-scenario behavioral intent analysis for advanced persistent threat detection. *ArXiv*, abs/2502.06521, 2025.

[Zhao *et al.*, 2025a] Haitao Zhao, Dan Li, Jinlong Sun, Xin Li, Haifeng Tang, and Gongrui Huang. Dtf-vpp: A dynamic intrusion detection method combining transformer and feature filtering for virtual power plant network security. *IEEE Transactions on Information Forensics and Security*, 20:11080–11092, 2025.

[Zhao *et al.*, 2025b] Jianjin Zhao, Qi Li, Zewei Han, Junsong Fu, Guoshun Nan, Meng Shen, and Bharat K. Bhargava. Retrial: Robust encrypted malicious traffic detection via discriminative relation incorporation and misleading relation correction. *IEEE Transactions on Information Forensics and Security*, 20:677–692, 2025.