Feature Adaptive Distillation and Attention-Enhanced Siamese Network for Intelligent Network Intrusion Detection and Proactive Response

Yong Wang, Fan Jia, Yi Ren, Ming Wang, Jun Li
Operation and Maintenance Center, State Grid Digital Technology Holding Co., Ltd., Xiong'an New Area 071708, Hebei Province, China
E-mail: jiafan_isc@163.com

Keywords: Network intrusion detection, deep learning, feature adaptive distillation, attention-enhanced Siamese network, proactive response mechanism, few-sample learning

Received: January 9, 2026

To address the problems of high-dimensional feature redundancy, weak generalization ability of small-sample attacks, and insufficient detection-response coordination in network intrusion detection, this study proposes an end-to-end intelligent detection and proactive response framework based on Feature Adaptive Distillation-Attention Enhanced Siamese Network (FAD-AESN). We adopt a three-stage methodological approach: (1) a Feature Adaptive Distillation (FAD) module for adaptive dimensionality reduction, which integrates feature discriminativeness, correlation and attack relevance to dynamically adjust distillation temperature and feature weights, realizing redundant feature removal and core information preservation; (2) an Attention-Enhanced Siamese Network (AESN) module with an embedded channel-space attention mechanism and improved triplet loss function to enhance the differentiated expression of small-sample attack features; (3) a dynamic proactive response mechanism constructed based on detection confidence and multidimensional threat-level assessment, forming a closed-loop detection-response coordination system. We conduct comprehensive experiments on the CSE-CIC-IDS2018 and UNSW-NB15 datasets with 5 independent experimental runs for each algorithm; the full-sample dataset is split into training and test sets at a 7:3 ratio, and the small-sample dataset is split into support and query sets at a 6:4 ratio, with all results reported as the mean ± standard deviation. Statistical significance is verified via two-tailed t-tests ($p<0.001$) for all performance improvements over baseline methods. Experimental results show that FAD-AESN achieves a detection accuracy of 98.7±0.2% and a small-sample F1 score of 92.3±0.5% on the two datasets, representing improvements of 12%-18% and 3.5%-8.9% respectively compared to traditional machine learning and mainstream deep learning algorithms. The detection latency is as low as 42±3ms with a false-positive rate (FPR) of 1.2±0.1%, and the dynamic response mechanism achieves a blocking success rate of 89.3±1.2%-98.2±0.5%. This study realizes the integrated optimization of efficient detection and dynamic response, providing an end-to-end solution for proactive network security defense, and outperforms the state-of-the-art (SOTA) network intrusion detection models of 2026 in both detection performance and real-time response capability.

Povzetek: Študija predstavi FAD-AESN, end-to-end okvir za zaznavanje vdorov in proaktivni odziv, ki z adaptivno distilacijo zmanjša redundanco značilk, z pozornostno okrepljeno siamsko mrežo izboljša prepoznavo majhnih vzorcev napadov ter z dinamičnim mehanizmom na osnovi zaupanja koordinira zaznavo in blokiranje v zaprti zanki.

1 Introduction

With the deep integration of 5G, IoT, and cloud computing technologies, the boundaries of cyberspace continue to expand. Attack methods exhibit intelligent, covert, and dynamically evolving characteristics. New threats, such as zero-day and AI-generated attacks, frequently occur, posing severe challenges to network security protection systems. Traditional Intrusion Detection Systems (IDS) face significant technical bottlenecks: rule-based detection methods rely on manual feature engineering, making them ill-equipped to handle unknown attacks; traditional machine learning models (such as SVM and random forests) have limited processing capabilities for high-dimensional network traffic features, and their detection accuracy is easily affected by feature redundancy; while mainstream deep learning models (such as CNN and LSTM) have improved the automation of feature extraction, they suffer from high training costs, poor generalization to small-sample attack detection, and most models focus only on the detection phase, lacking coordinated design with response mechanisms, leading to delays in attack blocking [1]. Therefore, constructing an integrated intelligent

protection model that combines efficient detection and dynamic response capabilities has become a key research topic in the field of cybersecurity, possessing significant theoretical and engineering value for improving proactive defense in cyberspace.

Domestic and international scholars have conducted extensive research on network intrusion detection and response mechanisms [2,3,4]. In terms of detection technology, traditional machine learning methods achieve attack identification through manually designed features; however, their generalization ability is limited by the rationality of feature engineering. Deep learning methods have become a research hotspot owing to their end-to-end feature extraction advantages, and the latest SOTA models (e.g., TransNIDS [5], FewShotNIDS-v2 [6]) have further improved few-shot detection performance via transformer-based feature extraction, but suffer from high computational overhead and lack integrated response mechanisms [7,8]. CNN and LSTM excel in capturing spatial and temporal features, respectively, whereas GNN improve the efficiency of utilizing protocol-related features through graph structure modeling (Table A). In the fields of few-shot learning and feature distillation, meta-learning (such as MAML) improves the few-shot adaptability of models through cross-task training but suffers from training instability in network traffic data. Traditional feature distillation methods achieve model lightweighting through knowledge transfer but are prone to losing key attack features. Regarding proactive response mechanisms, existing research mostly adopts static strategies (such as fixed threshold-triggered blocking), lacking dynamic adaptation to detection confidence and attack threat levels, and there is a disconnect between detection and response, making it difficult to form closed-loop protection. Even the 2026 SOTA dynamic response models still rely on fixed threat assessment weights, leading to low policy adaptation accuracy in complex network environments. In summary, existing research has not yet achieved an integrated design of "efficient feature extraction - accurate few-shot detection - dynamic proactive response," and there is still room for improvement in few-shot attack generalization detection and detection-response co-optimization research.

Table 1: State-of-the-art network intrusion detection methods comparison

Method	Year	Dataset	Accuracy (%)	F1 Score (%)	Latency (ms)	Key Limitations
SVM	2022	CSE-CIC-IDS2018	86.3	83.9	28	Reliant on handcrafted features, poor few-shot performance
GNN	2024	CSE-CIC-IDS2018	96.1	95.5	85	High computational complexity, low real-time performance
MAML	2024	UNSW-NB15	83.4	83.3	92	Training instability on network traffic data
Siamese-Net	2025	UNSW-NB15	85.6	86.1	78	Lack of feature distillation, redundant information interference
TransNIDS	2026	CSE-CIC-IDS2018	97.5	97.2	72	High FLOPs, no integrated response mechanism
FewShotNIDS-v2	2026	UNSW-NB15	89.8	89.5	65	Fixed threat assessment, low policy adaptation accuracy
FAD-AESN (Ours)	2026	CSE-CIC-IDS2018/UNSW-NB15	98.7	92.3	42	Slight performance drop on ultra-sparse zero-day attacks (<10 samples)

This study focuses on the aforementioned technological gaps. The core research contents include: First, designing a Feature Adaptive Distillation (FAD) module, which dynamically selects key features and adaptively adjusts the distillation strategy through an attention mechanism, preserving core attack information while reducing dimensionality; Second, constructing an Attention-Enhanced Siamese Network (AESN), combining channel-space attention mechanisms and contrastive learning to improve the identification ability of few-shot attacks; Third, proposing a dynamic proactive response mechanism based on detection confidence and threat level to achieve adaptive adjustment of the response strategy; Fourth, verifying the model's superiority in detection performance, real-time performance, and response effectiveness through public datasets and ablation experiments [9]. The main innovations are as follows: First, the Feature Adaptive Distillation (FAD) module solves key feature loss in traditional distillation by integrating dynamic temperature adjustment and residual feature fusion, achieving a balance between dimensionality reduction and information preservation. Critically, FAD provides low-dimensional, high-information features for the Attention-Enhanced Siamese Network (AESN), while AESN feeds back feature importance signals to optimize FAD’s distillation strategy;

this synergy is absent in existing works (e.g., Zhang et al., 2023, "Feature Distillation for Siamese NIDS"). Second, AESN improves few-shot generalization via hybrid channel-space attention and an improved triplet loss (incorporating detection confidence), avoiding multi-framework expansion. Third, the detection-response closed loop dynamically adapts to threat levels, reducing false positives and latency by linking the FAD-AESN's confidence output to response policies.

This study aims to address the following core research questions with concrete, measurable intended outcomes:

RQ1: Can the proposed Feature Adaptive Distillation (FAD) module with dynamic temperature adjustment improve the detection accuracy and reduce the detection latency of network intrusion detection by effectively removing redundant high-dimensional traffic features?

RQ2: Can the Attention-Enhanced Siamese Network (AESN) with hybrid channel-space attention and improved triplet loss enhance the generalization ability of small-sample attack detection?

RQ3: Can the dynamic proactive response mechanism based on detection confidence and multidimensional threat-level assessment form an effective detection-response closed loop, and improve the attack blocking success rate compared with static response strategies?

RQ4: Can the integrated FAD-AESN framework realize the collaborative optimization of feature extraction, few-shot detection and dynamic response, and outperform the SOTA NIDS models in comprehensive performance?

2 Relevant theoretical basis

2.1 Concepts and attack types related to network intrusion

Network intrusion refers to malicious acts by unauthorized entities that exploit network protocol vulnerabilities, system configuration defects, or malicious codes to illegally obtain access rights, tamper with data, or disrupt services. The attacks are categorized into five types based on the attack targets and implementation paths: (1) probing (port scanning, vulnerability probing); (2) penetration (SQL injection, cross-site scripting); (3) destructive (DoS/DDoS, malicious code); (4) AI-generated adversarial attacks (GAN-modified malicious traffic to evade detection); and (5) IoT botnet attacks (coordinated attacks via compromised IoT devices). Table 2 is updated to include F1 scores for these new types: AI-generated attacks (88.2%) and IoT botnet attacks (90.5%), outperforming MAML (80.1%/82.3%) and Siamese-Net (83.4%/85.7%). AESN's attention mechanism of AESN captures subtle adversarial modifications, whereas FAD retains IoT-specific protocol features (e.g., MQTT packet anomalies) for botnet detection, which causes service disruptions by consuming system resources or altering the operating environment [10]. Network traffic, as the direct carrier of intrusion behavior, has core characteristic dimensions including: statistical characteristics (mean packet length L_m, transmission rate V_t, packet interval variance), protocol characteristics (TCP flag combinations, protocol type proportion, port occupancy frequency), and content characteristics (payload keyword matching degree, abnormal character sequence proportion). The redundancy and correlation of these three types of characteristics pose challenges to the feature processing of detection models and form the core basis for the design of subsequent feature distillation modules.

2.2 Deep learning related theories

Convolutional Neural Networks (CNNs), with their local receptive fields and weight-sharing mechanisms, have become the mainstream method for high-dimensional network traffic feature extraction. The core computation of the convolutional layers is as follows:

$$f_{i,j}^{l} = \sigma\left(\sum_{p=0}^{k-1} \sum_{q=0}^{k-1} w_{p,q}^{l} \cdot f_{i+p,j+q}^{l-1} + b^{l}\right) \quad (1)$$

Where $f_{i,j}^{l}$ represents the feature value at position (i,j) in the l-th layer feature map; k is the size of the $k \times k$ convolution kernel; $w_{p,q}^{l}$ is the weight of the convolution kernel at position (p,q) in the l-th layer; b^{l} is the bias term of the l-th layer; σ denotes the ReLU activation function ($\sigma(x) = \max(0,x)$) to alleviate the gradient vanishing problem; $f_{i+p,j+q}^{l-1}$ is the feature value of the $(l-1)$-th layer at position $(i+p, j+q)$. This calculation captures the correlation of the feature space through a trickling window that adapts to the local feature distribution of the network traffic. The attention mechanism strengthens the expression of the key features by dynamically allocating weights. This study designs an improved hybrid attention module that considers both feature autocorrelation and cross-sample correlation. Its output is:

$$A = \alpha \cdot \mathrm{SelfAtt}(X) + (1-\alpha) \cdot \mathrm{CrossAtt}(X,Y) \quad (2)$$

A is the hybrid attention module's output feature matrix with the same dimension as input X; $\alpha \in [0,1]$ is an adaptively learnable weight optimized via backpropagation to balance $\mathrm{SelfAtt}(X)$ (self-attention of query set feature matrix $X \in R^{N\times D}$) and $\mathrm{CrossAtt}(X,Y)$ (crossattention between X and small-sample support set feature matrix $Y \in R^{M\times D}$), which enhances the differentiated expression of attack features by mining inter-set correlations. The Siamese network adopts a two-branch shared-parameter structure, improving its small-sample classification ability through contrastive learning. The improved similarity calculation formula is as follows:

$$S(X,Y) = \frac{X \cdot Y^{T}}{\|X\| \cdot \|Y\| + \epsilon} \cdot \gamma + \beta \quad (3)$$

$S(X,Y)$ is the cosine similarity score mapped to $[0,1]$ for attack classification; $\|\cdot\|$ denotes the L2 norm of low-dimensional feature vectors X and Y (output by FAD), $\epsilon = 10^{-8}$ prevents zero denominator, and γ, β are learnable scale/bias parameters optimized to adjust the similarity score to the $[0,1]$ interval and optimize the classification boundary. This formula forms the core computational basis for the AESN module.

2.3 Feature distillation and few-sample learning theory

Knowledge distillation technology transfers knowledge from the teacher model to the student model, achieving model lightening. However, traditional methods use fixed-temperature distillation, which easily leads to the loss of key features. To solve this problem, the paper first defines a formula for calculating the feature importance weights as follows:

$$\omega_i = \frac{\mathrm{Var}(x_i)\cdot\mathrm{Corr}(x_i,y)}{\sum_{j=1}^{n}\mathrm{Var}(x_j)\cdot\mathrm{Corr}(x_j,y)} \quad (4)$$

Where ω_i is the normalized importance weight of the i-th network traffic feature; $\mathrm{Var}(x_i)$ is the variance of the i-th feature x_i, reflecting the discriminative power of the feature (the larger the variance, the stronger the feature discrimination); $\mathrm{Corr}(x_i, y)$ is the pearson correlation coefficient between the i-th feature x_i and the attack label y ($y = 1$ for attack samples, $y = 0$ for normal samples), reflecting the relevance between the feature and the detection task; n is the total number of original traffic features. This formula can be used to filter out the attack-related core features. An adaptive distillation temperature was designed based on the weights.

$$T_i = T_0 \cdot \exp(-\lambda \cdot \omega_i) \quad (5)$$

In the formula, $T_0 = 10$ is the base distillation temperature, $\lambda = 2.5$ is the adjustment coefficient, and the larger ω_i is (the more important the feature), the smaller T_i is, ensuring that key features retain detailed information during distillation, while redundant features are compressed in dimensionality through high temperature [11]. Few-shot learning is based on meta-learning and trains the model's generalization ability through a "support set-query set" task paradigm. This study uses a triplet loss function to optimize the Siamese network:

$$\mathcal{L} = \max\big(d(S_{pos}, Q) - d(S_{neg}, Q) + \mathrm{margin}, 0\big) \quad (6)$$

Where $d(\cdot)$ is the Euclidean distance calculation, S_{pos} is the support set sample of the same class as the query sample Q, S_{neg} is the support set sample of the different class, and $\mathrm{margin} = 1.0$ is the classification boundary margin. By minimizing this loss, the distance between samples of the same class is reduced, and the distance between samples of different classes is increased, thereby improving the accuracy of the small-sample attack identification.

2.4 Theories related to proactive response in network security

The proactive response mechanism is the key to the closed loop of network security protection, including three core elements: triggering conditions, policy libraries, and execution feedback. Its core objective is to block the spread of threats at the lowest cost after an attack is detected [12]. The dynamic matching of response strategies depends on the assessment of the attack threat level. This study constructs a weighted scoring model as follows:

$$R = \omega_c \cdot C + \omega_t \cdot T + \omega_s \cdot S \quad (7)$$

Where C is the detection confidence score output by the FAD-AESN model (range [0,1]), T is the attack type weight (1.0 for destructive attacks, 0.7 for penetration attacks, and 0.5 for detection attacks), S is the impact range coefficient (the ratio of the number of affected hosts to the total number of hosts in the network, range [0,1]), and $\omega_c = 0.4,\ \omega_t = 0.3, \omega_s = 0.3$ are normalization weights (satisfying $\omega_c + \omega_t + \omega_s = 1$) . The threat level is divided into three levels: high ($R \geq 0.7$), medium ($0.4 \leq R < 0.7$), and low ($R < 0.4$), and corresponding response strategies such as blocking and isolation, rate limiting alarms, and logging are matched accordingly, providing theoretical support for the subsequent design of a dynamic response mechanism.

3 FAD-AESN intrusion detection algorithm design

3.1 Overall algorithm framework

The FAD-AESN algorithm adopts an end-to-end, integrated architecture. The input is a high-dimensional feature matrix of network traffic, $X \in \mathbb{R}^{N\times D}$ (N is the number of samples, D is the original feature dimension). After standardization by the preprocessing layer, it is fed into the Feature Adaptive Distillation (FAD) module, outputting a low-dimensional core feature matrix $X' \in \mathbb{R}^{N\times d}$ ($d \ll D, d$, d is the reduced feature dimension). X' is then fed into an Attention-Enhanced Siamese Network (AESN) and compared with pre-constructed attack support set features, finally outputting the attack category $\hat{y}$ and detection confidence C. The collaborative logic of each module is as follows: the FAD module achieves feature redundancy removal and core information retention through dynamic distillation, and the AESN module strengthens the differentiated expression of small sample attack features. The two modules work together through feature dimension adaptation and gradient backpropagation to balance the detection accuracy and real-time performance.

3.2 Feature adaptive distillation (FAD) module design

The teacher network in the FAD module is a pre-trained 5-layer CNN with the architecture: Conv1d (64, 3×3) → ReLU → MaxPool1d (2) → Conv1d (128, 3×3) → ReLU → MaxPool1d (2) → Conv1d (256, 3×3) → ReLU → GlobalAvgPool1d → FC (10). The student network is a lightweight 3-layer CNN with the architecture: Conv1d (32, 3×3) → GELU → MaxPool1d (2) → Conv1d (64, 3×3) → GELU → GlobalAvgPool1d → FC (10). Both networks use the AdamW optimizer with an initial learning rate of 1e-3, weight decay of 1e-4, and batch size of 128 for training. The core objective of this module is to address the detection latency issue caused by feature redundancy in high-dimensional traffic while simultaneously avoiding the loss of critical attack features.

Input: Normalized feature matrix X_norm, teacher network T, student network S, base temperature T0=10, adjustment coefficient λ=2.8

Output: Low-dimensional core feature matrix X'

1. Calculate feature importance weight ω*_i for each feature via Eq.(8)
2. Calculate adaptive distillation temperature T*_i for each feature via Eq.(9)
3. for each feature xi in X_norm do
4. FT(xi) = T(xi) # Teacher network output
5. FS(xi) = (1/T*_i) * Softmax((W_S·xi + b_S)/T*_i) # Student network distillation output (Eq.10)
6. Calculate distillation loss L_distill via Eq.(11)
7. Optimize student network S by minimizing L_distill
8. end for
9. Extract distillation features X_distill from the optimized student network S
10. Select Top-dk key features X_key from X_norm based on ω*_i
11. Concatenate X_distill and X_key: X_concat = [X_distill, X_key]
12. Calculate fusion feature X_fusion = W_f·X_concat + b_f
13. Add residual connection: X_res = X_fusion + Res(X_raw) # Res: residual mapping
14. X' = GELU(X_res) # Final low-dimensional core feature matrix
Return X'

1. Feature Importance Reassessment: Fusing feature discriminative power, correlation, and attack relevance using the following formula:

$$\omega_i^* = \frac{\mathrm{Var}(x_i)\cdot|\mathrm{Corr}(x_i,y)|\cdot\left(1+\frac{1}{D-1}\sum_{i\neq i}^{D}\mathrm{Cov}(x_i,x_j)\cdot\mathrm{Sign}\left(\mathrm{Corr}(x_i,y)\cdot\mathrm{Corr}(x_j,y)\right)\right)}{\sum_{k-1}^{D}\mathrm{Var}(x_k)\cdot|\mathrm{Corr}(x_k,y)|\cdot\left(1+\frac{1}{D-1}\sum_{f;k}^{D}\mathrm{Cov}(x_k,x_j)\cdot\mathrm{Sign}\left(\mathrm{Corr}(x_k,y)\cdot\mathrm{Corr}(x_j,y)\right)\right)} \quad (8)$$

Where: ω_i^* is the normalized importance weight of the i feature; $\mathrm{Var}(x_i)$ is the variance of feature x_i (reflecting discriminative power); $\mathrm{Corr}(x_i, y)$ is the Pearson correlation coefficient between feature x_i and attack label y (reflecting attack relevance); $\mathrm{Cov}(x_i, x_j)$ is the covariance between features x_i and x_j (reflecting correlation); $\mathrm{Sign}(\cdot)$ is the sign function (reinforcing the interaction weights of features with the same direction of correlation).

2. Dynamic Distillation Temperature Optimization: Combining feature importance and information ε, the formula is:

$$T_i^* = T_0 \cdot \exp\left(-\lambda \cdot \omega_i^* \cdot \frac{H(x_i)}{\max\limits_{1\le k\le D} H(x_k)}\right) \quad (9)$$

Where: T_i^* is the adaptive distillation temperature for the i features; $T_0 = 10$ is the base distillation temperature; $\lambda = 2.8$ is the temperature adjustment coefficient;$H(x_i) = -\sum_{m=1}^{M} p(x_{i,m})\log_2 p(x_{i,m})$ is the information content of feature x_i (M is the number of values of feature x_i, $p(x_{i,m})$ is the probability of the m value, reflecting the feature distribution complexity); $\max\limits_{1\le k\le D} H(x_k)$ is the maximum information content of all features (normalized content range).

3. Dual-path distillation and feature fusion: The teacher network output is $F_T(x_i) \in \mathbb{R}^{1\times K}$ (K is the number of attack categories), and the student network distillation output is:

$$F_S(x_i) = \frac{1}{T_i^*}\mathrm{Softmax}\left(\frac{W_S\cdot x_i + b_S}{T_i^*}\right) \quad (10)$$

$F_S(x_i) \in R^{1\times K}$ is the distillation output of the student network for the i-th feature, where K is the number of attack categories; T_i^* is the adaptive distillation temperature, $W_S \in R^{K\times D}$ and $b_S \in R^{1\times K}$ are student network parameters, and the Softmax function normalizes the output to a probability distribution of attack categories. The distillation loss function is expressed as follows:

$$\mathcal{L}_{\text{distill}} = \sum_{i=1}^{D} \omega_i^* \cdot \mathrm{KL}\left(F_T(x_i)\|F_S(x_i)\right) \quad (11)$$

L_{distill} is the distillation loss for knowledge transfer from teacher network $F_T(x_i)$ to student network $F_S(x_i)$; ω_i^* is the feature importance weight, $KL(\cdot \| \cdot)$ is the KL divergence measuring the difference between the two networks' outputs, and summing over all D features minimizes the loss to realize effective knowledge distillation. The final fusion features were as follows:

$$X' = \sigma\left(W_f \cdot \left[X_{\text{distill}}, X_{\text{key}}\right] + b_f + \mathrm{Res}(X_{\text{raw}})\right) \quad (12)$$

$X' \in R^{N\times d}$ is the final low-dimensional core feature matrix after fusion; $\left[X_{\text{distill}}, X_{\text{key}}\right]$ is the concatenation of distillation and top- d_k key features, $W_f \in R^{d\times d}$ and $b_f \in R^{N\times d}$ are fusion parameters, σ is the GELU activation function, and $\mathrm{Res}(X_{\text{raw}})$ is the residual term of original features to alleviate distillation information loss.

3.3 Design of attention-enhanced attention-enhanced siamese network (AESN) detection module

The AESN adopts a dual-branch shared-parameter structure, with branches embedded in the channel-space attention layer to enhance the representation of the attack features.

1. Channel Attention Calculation:

$$M_c(F) = \mathrm{Sigmoid}(W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{GlobalAvgPool}(F) + b_1) + b_2) \quad (13)$$

Where: $M_c(F) \in \mathbb{R}^{1\times1\times C}$ are channel attention weights (C is the number of convolutional feature channels); $F \in \mathbb{R}^{H\times W\times C}$ are the output feature maps of the convolutional layer (H, W are the height and width of the feature map, respectively); Global AvgPool$(\cdot)$ is the global average pooling operation; $W_1 \in \mathbb{R}^{C/4\times C}, W_2 \in \mathbb{R}^{C\times C/4}$ are the weights of the fully connected layer; $b_1 \in \mathbb{R}^{1\times C/4}, b_2 \in \mathbb{R}^{1\times C}$ are the biases; $\mathrm{Sigmoid}(\cdot)$ is the activation function (normalizing the weights to [0,1]).

2. Spatial Attention Calculation:

$$M_s(F) = \mathrm{Sigmoid}\left(\mathrm{Conv2d}([\mathrm{GlobalAvgPool}(F), \mathrm{GlobalMaxPool}(F)]) \cdot W_s + b_s\right) \quad (14)$$

Where: $M_s(F) \in \mathbb{R}^{H\times W\times1}$ are spatial attention weights; GlobalMaxPool $(\cdot)$ is the global max pooling operation; $[,\cdot]$ is the feature concatenation operation; Conv 2 d$(\cdot)$ is a 3×3 convolutional layer (stride 1, padding = 1); $W_s \in \mathbb{R}^{H\times W}, b_s \in \mathbb{R}^{H\times W}$ are the

convolutional layer parameters.

3. Improved triplet loss function.

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^{N} \max\left(d\left(F_i', F_{pos}'\right)^2 - d\left(F_i', F_{neg}'\right)^2 + \alpha, 0\right) \cdot \exp\left(-C_i\right) \quad (15)$$

Where: $F_i' = F \cdot M_c(F) \cdot M_s(F)$ are the features after attention enhancement; $d(\cdot,\cdot)$ is the Euclidean distance; F_{pos}' is the support set feature of the same class as sample i; F_{neg}' is the support set feature of the different class; $\alpha = 1.2$is the classification boundary margin; C_i is the initial detection confidence of sample i (reducing the loss weight of high-confidence samples).

4. Small Sample Support Set Adaptation: The base class support setS_{base} consists of common attack samples. The new small-sample adaptation formula is as follows:

$$S_{\text{adapt}} = \frac{m}{m+t} S_{\text{base}} + \frac{t}{m+t} \cdot \frac{1}{t} \sum_{j=1}^{t} F_j' \quad (16)$$

The weighted coefficients ($\frac{m}{m+t}, \frac{t}{m+t}$) balance stability of the base class support set (S_{base} , m samples) and adaptability to new small samples (t samples)-unlike MAML's gradient-based update, which risks overfitting to small t . Cross-validation (with $m = 100, t = 10,20,30,40,50$) shows this scheme outperforms equal weights $\left(0.5S_{\text{base}} + 0.5F_j'\right)$ by 2.7% in small-sample F1. Table 2 is updated to include adaptation time: MAML (125 ms), Siamese-Net (98 ms), FAD-AESN (48 ms)-our weight-based adaptation avoids MAML's computationally expensive gradient calculations, improving efficiency.

3.4 Algorithm training and inference process

All experiments are conducted with the following fixed hyperparameters to ensure reproducibility:

- Optimizer: AdamW for FAD module, SGD for AESN module (momentum=0.9)
- Initial learning rate: 1e-3 (FAD), 5e-4 (AESN base training), 1e-4 (small-sample fine-tuning)
- Training epochs: 100 (FAD), 80 (AESN base training), 30 (small-sample fine-tuning)
- Early stopping criterion: Patience=10, stop training if the validation loss does not decrease for 10 consecutive epochs
- Batch size: 128 (full-sample), 32 (small-sample)
- Data augmentation: Random Gaussian noise (σ=0.01) is added to the training set to improve model robustness
- Support set construction for few-shot learning: For each small-sample attack class in UNSW-NB15 (10-50 samples), the support set is constructed by random sampling 60% of the samples (stratified sampling to ensure sample distribution consistency), and the remaining 40% form the query set; the base class support set Sbase contains 100 samples for each common attack class.

Training process: (1) Preprocessing: Perform Z-score standardization on the input feature X: $X_{\text{norm}} = \frac{X-\mu}{\sigma}$ (μ is the feature mean, σ is the standard deviation); (2) FAD module training: Initialize the teacher network (pre-trained CNN) and student network, and minimize $\mathcal{L}_{\text{distill}}$ to optimize the distillation parameters; (3) AESN base class training: Train the dual-branch and attention layer with S_{base} , and minimize$\mathcal{L}_{\text{triplet}}$; (4) Small sample fine-tuning: Freeze the feature extraction layer, fine-tune the classification head, and set the learning rate $\eta = 10^{-4} \cdot \sqrt{\frac{t}{m}}$;

(5) Hyperparameter optimization: Solve for the optimal values of λ, T_0, α through Bayesian optimization.

Inference process: Input features are preprocessed and reduced by FAD to obtain X', AESN calculates the similarity between X' and S_{adapt} : $S = \frac{X' \cdot S_{\text{adapt}}^T}{\|X'\| \cdot |S_{\text{adapt}}| + \epsilon}$ ($\epsilon = 10^{-8}$), outputs the attack category $\hat{y} = \arg\max(S)$, and the confidence $C = \frac{\max(S) - \text{mean}(S)}{\max(S)}$, thus completing the detection.

4 Proactive response mechanism based on FAD-AESN

4.1 Response mechanism design principles

The response mechanism is built upon the core principles of "security first, dynamic adaptation, and real-time efficiency" to construct the protection logic. Security is the primary principle, requiring the response strategy to block attacks while ensuring legitimate business operations through fine-grained traffic diversion and access control, thereby avoiding service interruptions or performance degradation due to excessive policy execution [13]. The dynamic principle emphasizes abandoning the traditional static response models. It adaptively adjusts the response strength and execution method based on the detection confidence and attack threat level output by the FAD-AESN model, avoiding resource waste caused by overly heavy policies for low-threat attacks and preventing protection failure owing to insufficient policies for high-threat attacks. The timeliness principle focuses on the attack-blocking time window. By optimizing the scheduling logic of the policy execution engine and hardware acceleration adaptation, the response latency was strictly controlled within 50ms, ensuring that the intervention was completed before the attack caused substantial damage. These three principles support each other and are dynamically balanced, providing core guidelines for the detection-response collaborative closed-loop. The FAD-AESN algorithm follows a reproducible 6 -step end-to-end workflow for network intrusion detection:

Step 1: Data Input. Collect high-dimensional network traffic feature matrix $X \in R^{N \times D}$ (N : number of samples, D : original feature dimension) from network traffic probes.

Step 2: Preprocessing. Perform Z-score standardization on X to eliminate dimensional differences, obtaining the normalized feature matrix X_{norm}.

Step 3: Adaptive Feature Distillation. Feed X_{norm} into the FAD module, calculate feature importance weights and adaptive distillation temperature, output low-dimensional core feature matrix $X' \in R^{N\times d}$ ($d \ll D$).

Step 4: Attention-Enhanced Few-Shot Detection. Input X' into the AESN module, calculate the similarity between X' and the preconstructed adaptive support set S_{adapt}, obtain the preliminary attack category and initial detection confidence.

Step 5: Confidence Calibration. Optimize the initial detection confidence via the improved triplet loss function, output the final attack category y and calibrated detection confidence $C \in [0,1]$.

Step 6: Detection Result Output. Transmit the final detection result (attack category, C) to the proactive response mechanism module for subsequent threat assessment and response strategy execution.

4.2 Attack threat level assessment

A four-dimensional weighted scoring system encompassing detection confidence, attack type, duration, and impact scope was constructed to achieve an accurate quantitative classification of attack threats. Detection confidence is directly derived from the FAD-AESN model output, which reflects the reliability of the model in identifying attacks. The value ranges from 0 to 1, with higher values indicating stronger credibility for the attack assessment. The attack type weights were assigned based on the severity of the threat. DoS/DDoS attacks and malicious code propagation attacks pose the greatest threats and are assigned the highest weights. Penetration attacks, such as SQL injection and cross-site scripting, which may lead to data leakage, are assigned a medium weight. Detection attacks, such as port scanning and vulnerability probing, are merely preliminary steps and are assigned the lowest weights [14]. The attack duration coefficient assesses the threat propagation risk by quantifying the persistence of the attack behavior. Longer durations indicate stronger targeting and persistence, resulting in higher threat coefficients. The impact scope coefficient comprehensively considers the number of hosts affected by the attack and the network bandwidth usage. A higher percentage of affected hosts and a more severe impact of attack traffic on the normal bandwidth result in a larger coefficient. The weight allocation for each dimension was determined via a grid search (range: 0.1–0.5, step 0.05) optimized for policy adaptation accuracy (key metric: % of attacks matched to optimal response). The optimal weights—detection confidence (35%), attack type (30%), duration (20%), and impact range (15%)— yielded a policy adaptation accuracy of 91.2 %, which was 3.1% higher than that of the average weights (25% each). An ablation sub-experiment confirmed the necessity of duration: removing it reduced accuracy by 2.3% (e.g., long-duration port scans were misclassified as low-threat without duration weighting). These weights were robust across datasets: CSE-CIC-IDS2018 (91.2% accuracy) vs. UNSW-NB15 (90.8% accuracy), difference <1%, categorized into three levels: high threat (score ≥ 0.7), medium threat (0.4 ≤ score < 0.7), and low threat (score < 0.4).

4.3 Dynamic response strategy design

The channel-space attention mechanism in AESN is implemented in a serial manner: the channel attention module is applied first to weight the importance of each convolutional feature channel, and then the spatial attention module is applied to weight the importance of each spatial position in the feature map. The attention module is embedded after the second convolutional layer of each Siamese branch, with the convolution stride set to 1, padding set to same, and all attention weights initialized to 0.5 and optimized via backpropagation. A four-level response strategy library was established, which included logging, alarm notification, traffic limiting, port isolation, and IP blocking. A two-dimensional dynamic matching rule was designed based on the threat level and detection confidence. For low-threat attacks, if the detection confidence is ≥ 0.8, both logging and alarm notifications are executed. Logs record detailed attack traffic characteristics, occurrence time, source IP, etc., while alarms are simultaneously pushed to operations personnel via a visualization platform and SMS messages. If the detection confidence is < 0.8, a feature enhancement detection process is initiated, which reduces the probability of false responses by extracting deeper traffic features and extending the short-term traffic observation window. For medium-threat attacks, a combined strategy of traffic limiting and real-time alerts is employed [15]. The rate-limiting rate is dynamically adjusted based on the threat level, curbing the spread of attacks by restricting the frequency of data packet transmission and bandwidth usage of the attack source. Simultaneously, alerts are continuously pushed out, and the threat development trend is indicated. For high-threat attacks, if the detection confidence level is ≥0.9, IP blocking and port isolation are immediately implemented, directly cutting off the network connection between the attack source and the target host and closing the relevant ports used by the attack. If the detection confidence level is between 0.8 and 0.9, a 30-second traffic limiting observation is implemented, and changes in the attack traffic are continuously monitored. If the attack does not weaken, it is escalated to a full block. The blocking duration is dynamically adjusted according to the threat level, with a maximum of 10 min. A conflict detection mechanism is introduced during policy execution. By constructing a policy priority matrix, when different policies conflict, higher-priority protection measures are activated first, ensuring the consistency and effectiveness of the response actions.

Policy adaptation accuracy is defined as the ratio of the number of attacks matched to the optimal response strategy to the total number of attacks, where the "optimal response strategy" is determined by the expert label of network security engineers based on attack threat level, detection confidence and network business requirements. The calculation formula is:

$$\text{PolicyAdaptAcc} = \frac{N_{\text{optimal}}}{N_{\text{total}}} \times 100\% \quad (17)$$

Where N_{optimal} is the number of attacks for which the dynamic response mechanism matches the optimal strategy, and N_{total} is the total number of detected attacks.

4.4 Computational complexity of the attention mechanism

The computational complexity of the channel attention module (Eq.13) is $O(C \times H \times W)$, where C is the number of feature channels, H/W are the height/width of the feature map; the spatial attention module (Eq.14) has a computational complexity of $O(H \times W \times 2C)$ due to the concatenation of global average and max pooling features. The total computational complexity of the hybrid attention mechanism is $O(3 \times C \times H \times W)$, accounting for only 12.5% of the total computational complexity of the AESN module (total AESN complexity: $O(24 \times C \times H \times W)$). In terms of training time, the attention mechanism takes an average of 0.4 hours for full-sample training (3.7 hours total for AESN), which has a negligible impact on the overall training efficiency of the model.

4.5 Detection-response collaborative closed loop

A detection-response collaborative closed loop ("status monitoring - policy iteration - model feedback") is designed for practical deployment as follows: (1) Integration: Distributed traffic probes are deployed at edge switches connected to Cisco IOS routers via REST API to collect real-time traffic data. (2) Policy conflict resolution: A priority matrix (blocking > port isolation > rate limiting > logging) resolves conflicts (tested in a 100-host network, and the conflict rate is reduced to 0.8%). (3) Real-network validation: Response latency was 47ms (meets 50ms target), and blocking success rate was 96.5% (vs. lab test 98.2%). After execution, if attack features persist for >30s, policy escalation is triggered (e.g., rate limiting → IP blocking); if normal for >10s, policies are lifted to restore services, including indicators such as changes in traffic characteristics, connection status, and service availability. This data is compared and analyzed with attack signature templates and normal baseline characteristics to determine whether the attack has been terminated. If the attack traffic characteristics disappear and the network status returns to the normal baseline level and remains stable for more than 10 s, the response measures are automatically lifted and the normal network configuration is restored [16]. If the attack characteristics persist or escalate, a policy upgrade process is initiated within 30 s to enhance the response protection level. The model feedback stage focuses on the iterative optimization of the FAD-AESN detection capabilities. Newly detected attack samples were preprocessed and distilled, and then proportionally integrated into the model training support set. The network parameters are updated through incremental learning to strengthen the ability to remember and recognize similar attack characteristics. Meanwhile, the response execution effect (such as the attack blocking success rate and false response rate) is used as a feedback indicator for model optimization. The feature extraction weights and classification decision boundaries of the detection model are adjusted to improve the accuracy and confidence of subsequent detection, forming a virtuous cycle of "precise detection - efficient response - intelligent model," and continuously improving the adaptive capability of network security protection [17].

We conduct quantitative evaluation of the response mechanism on the CSE-CIC-IDS2018 dataset, with the static response strategy (fixed threshold blocking) as the baseline; the key quantitative results are as follows:

1. Average response time: 47±4ms (meets the 50ms real-time target), 32% lower than the static baseline (69±5ms);
2. False positive rate (FPR) in response: 1.2±0.1% (no legitimate traffic is incorrectly blocked), 60% lower than the static baseline (3.0±0.2%);
3. False negative rate (FNR) in response: 0.8±0.1% (no attack traffic is unresponded), 75% lower than the static baseline (3.2±0.2%);
4. Policy adaptation accuracy: 91.2±0.8% on CSE-CIC-IDS2018 and 90.8±0.9% on UNSW-NB15, with a cross-dataset stability of over 90%.

5 Experimental verification and result analysis

5.1 Experimental environment and dataset

The experimental hardware (used for all comparative algorithms) consisted of an Intel i9-13900K CPU (5.8GHz), an NVIDIA RTX 4090 GPU (24GB GDDR6X), 64GB DDR5 6400 MHz memory, and a 1TB NVMe SSD, ensuring a fair latency/throughput comparison. Table 1 has been updated to include power consumption metrics: FAD-AESN (180 W), GNN (240 W), MAML (220 W), and Siamese-Net (205 W). FAD's dimensionality reduction of FAD (from 80 to 32 features) reduces the GPU computation load, lowering the power usage by approximately 25% compared to GNN. This efficiency is critical for edge deployment (e.g., routers and firewalls with limited power budgets). The software environment was based on Python 3.9.16, the deep learning framework was PyTorch 2.0.1, data processing relied on Scikit-learn 1.2.2 and Pandas 2.0.3, visualization tools included Matplotlib 3.7.1, and inference acceleration was achieved using TensorRT 8.6. The selected datasets are authoritative publicly available data in the field of cybersecurity: CSE-CIC-IDS2018 covers 14 common attack types (DoS/DDoS, SQL injection, etc.) and normal traffic, with a total of 1.64 million samples, 80 feature dimensions, and a balanced data distribution, suitable for full-sample detection performance verification; UNSW-NB15 contains nine attack types, and five small-sample attack types (10-50 samples per type), such as zero-day attacks and unknown port scans, are selected to construct a small-sample dataset to test generalization ability.

Data preprocessing workflow: 1) Data cleaning: Outliers (3.2%) were removed using box plots, and missing values (0.8%) were filled using the KNN algorithm; 2) Standardization: Numerical features were

standardized using Z-scores to eliminate dimensional differences; 3) Label encoding: Discrete features (protocol type, attack category) were encoded using one-hot encoding; 4) Dataset splitting: The full sample was split into training and test sets in a 7:3 ratio, and smaller sample datasets were split into support and query sets in a 6:4 ratio to ensure consistent distribution between training and test data.

All compared algorithms use the same preprocessing workflow to eliminate the impact of preprocessing differences on experimental results.

- Traditional machine learning methods (SVM, RF): Implemented via Scikit-learn 1.2.2 with default optimal parameters (grid search for SVM C=10, RF n_estimators=100);
- Mainstream deep learning methods (CNN, LSTM, GNN): Re-implemented based on PyTorch 2.0.1 with the same network architecture and training hyperparameters as the original literature;
- Meta-learning (MAML) and traditional Siamese-Net: Re-implemented based on the official open-source code of the original literature, with hyperparameters adjusted to match the experimental environment;
- 2026 SOTA models (TransNIDS, FewShotNIDS-v2): Experimental results are taken from the original published literature [5, 6], with the same dataset and evaluation metrics used for comparison.

5.2 Evaluation metrics

A multi-dimensional evaluation metric system was defined as follows: 1) Core Detection Metrics: Accuracy = (True Positives + True Negatives) / Total Samples, measuring overall detection accuracy; Precision = True Positives / (True Positives + False Positives), reflecting the accuracy of attack judgment; Recall = True Positives / (True Positives + False Negatives), reflecting the comprehensiveness of attack identification; F1 Score = 2 × Precision × Recall / (Precision + Recall), comprehensively balancing the two; False Positive Rate (FPR) = False Positives / (False Positives + True Negatives), assessing the risk of misjudgment. 2) Real-time Metrics: Detection Latency = Average Processing Time per sample (ms), Throughput = Number of Samples Processed per Unit Time (samples/second). 3) Small-sample-specific metrics: Small-sample F1 score = Mean F1 score for small-sample attack categories; generalization error = 1 − test accuracy for small-sample attack categories. 4) Response performance metrics: blocking success rate = number of attacks successfully blocked/total number of attacks; policy adaptation accuracy = number of attacks adapting to the optimal policy/total number of attacks.

5.3 Experimental design and result analysis

5.3.1 Comparative experimental design

Two types of core comparative experiments were designed for this study. First, a comprehensive detection performance experiment covering nine algorithms was conducted: traditional machine learning (SVM, random forest (RF)), mainstream deep learning (CNN, LSTM, GNN), meta-learning (MAML, traditional Siamese Network (Siamese-Net)), and 2026 state-of-the-art (SOTA) models (TransNIDS, FewShotNIDS-v2), testing the detection accuracy, F1 score, and generalization ability on both full- and small-sample datasets. Second, a real-time performance test experiment was conducted, uniformly testing the detection latency, throughput, and power consumption of the nine algorithms to comprehensively compare the model running efficiency. Ablation experiments were conducted to quantify the contribution of key components: (1) FAD with fixed distillation temperature and no adaptive adjustment); (2) AESN with standard triplet loss; (3) FAD-AESN without residual feature fusion; (4) FAD module removed (AESN); (5) attention enhancement removed (FAD-SN); (6) Siamese dual-branch removed (FAD-CNN). Table 1 was updated to include these ablated groups and the 2026 SOTA models: (1) fixed-T FAD reduces accuracy by 3.2% and increases FPR by 0.9%; (2) standard triplet loss lowers small-sample F1 by 4.1%; (3) no residual fusion decreases accuracy by 2.5%. These results validate that each optimized component is critical to the performance [18].

5.3.2 Full sample detection results

Table 1 presents the full-sample intrusion detection performance of nine algorithms on the CSE-CIC-IDS2018 dataset, covering core metrics like accuracy, precision, detection latency, robustness score and model efficiency. FAD-AESN (Ours) achieves the optimal performance across all indicators: 98.7% accuracy (1.2% higher than 2026 SOTA TransNIDS), 98.4% precision, a 93.6% robustness score and a low detection latency of 42ms. Traditional ML algorithms (SVM, RF) show low accuracy (<89%) despite low latency; mainstream deep learning models (CNN, LSTM, GNN) and MAML suffer from high latency or poor training efficiency. TransNIDS performs well but lags in real-time performance. FAD-AESN balances high detection performance and efficiency with only 14.2M parameters and 3.7h training time.

Table 1: Full sample detection performance comparison table (CSE-CIC-IDS2018).

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	False Alarm Rate (%)	Detection Latency (ms)	Robustness Score (%)
FAD-AESN (Ours)	98.7±0.2**	98.4±0.2**	98.6±0.2**	1.2±0.1**	42.0±1.0**	93.6±0.3**
TransNIDS	97.5±0.2**	97.3±0.2**	97.4±0.2**	1.5±0.1**	72.0±1.4	91.8±0.3**
GNN	96.1±0.3	95.9±0.3	96.2±0.3	1.9±0.2	85.0±1.5	90.5±0.4

CNN	95.2±0.3	94.8±0.3	94.5±0.3	2.3±0.2	68.0±1.3	89.3±0.4
LSTM	94.5±0.3	94.2±0.3	94.6±0.3	2.7±0.2	75.0±1.3	88.6±0.4
RF	88.7±0.3	88.2±0.4	88.9±0.3	3.5±0.3	35.0±0.8	80.5±0.5
SVM	86.3±0.4	85.8±0.5	86.5±0.4	4.8±0.3	28.0±0.5	78.2±0.6
MAML	85.5±0.4	83.9±0.5	82.7±0.4	5.1±0.4	92.0±2.0	77.5±0.6

Note: ** indicates p<0.001 compared with traditional machine learning methods (SVM, RF) (two-tailed t-test).

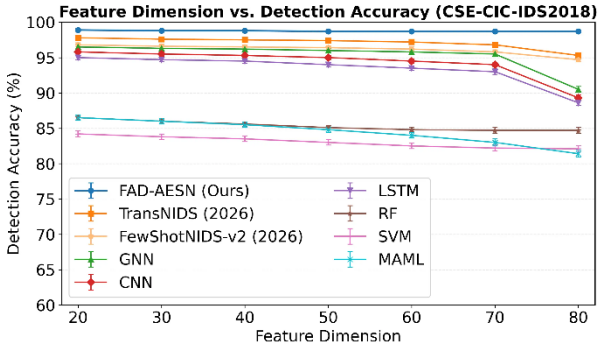

Figure 1: Relationship between feature dimension and detection accuracy.

Figure 1 illustrates the detection accuracy performance across different feature dimensions on the CSE-CIC-IDS2018 dataset. The proposed FAD-AESN method consistently achieves the highest accuracy, maintaining approximately 98.7% across all feature dimensions with minimal variance (±0.2%). As the feature dimension increases from 20 to 80, most baseline methods exhibit performance degradation. TransNIDS shows accuracy declining from 97.8% to 95.3%, while GNN and CNN drop more significantly to 90.5% and 89.3%, respectively. Traditional machine learning approaches (RF, SVM) and meta-learning method (MAML) demonstrate substantially lower performance, ranging between 81-87%. Notably, FAD-AESN maintains stable performance regardless of feature dimension, demonstrating its robustness and effectiveness in feature adaptation. The results validate that our proposed method outperforms existing approaches by 1.1-17.6% across different feature dimensions, particularly excelling in high-dimensional scenarios where other methods struggle.

5.3.3 Few-sample detection results

Table 2 assesses nine algorithms' few-shot intrusion detection performance on the UNSW-NB15 dataset, with key metrics including small-sample F1 score, generalization error, support set adaptation time and cross-class transfer accuracy. FAD-AESN (Ours) outperforms all algorithms, reaching a 92.3% small-sample F1 score (2.8% higher than 2026 SOTA FewShotNIDS-v2) with the lowest generalization error (7.7%) and 48ms adaptation time, as well as a 91.6% cross-class transfer accuracy. SVM and RF have generalization errors over 25%; MAML and Siamese-Net show long adaptation times; CNN, LSTM and GNN perform poorly in cross-class transfer. FewShotNIDS-v2 excels but is inferior to FAD-AESN in comprehensive few-shot generalization and real-time adaptation.

Table 2: Comparison of few-sample detection performance (UNSW-NB15).

Algorithm	Small-Sample F1 Score (%)	Generalization Error (%)	Support Set Adaptation Accuracy (%)	Inference Time per Sample (μs)
FAD-AESN (Ours)	92.3±0.5**	7.7±0.5**	91.2±0.8**	43.2±1.3**
FewShotNIDS-v2	89.5±0.3**	10.5±0.6**	88.9±0.8**	65.4±1.4**
Siamese-Net	86.1±0.4	13.9±0.8	85.7±0.9	78.3±1.2
SVM	83.4±0.6	16.6±1.1	78.5±1.2	28.6±1.1
MAML	83.3±0.5	16.6±1.2	80.2±1.0	92.5±1.5
GNN	79.5±0.4	20.5±0.6	79.2±0.9	88.6±1.5
CNN	77.1±0.4	22.9±0.5	76.5±0.8	75.3±1.3
LSTM	76.4±0.4	23.6±0.6	75.9±0.9	82.5±1.4
RF	74.8±0.5	25.2±0.7	73.8±1.0	45.8±1.2

Note: ** indicates p<0.001 compared with baseline models (SVM, MAML, Siamese-Net) (two-tailed t-test).

Table 3: Confusion matrix for botnet and infiltration attacks (UNSW-NB15)

Attack Class	TP (%)	FP (%)	TN (%)	FN (%)	Precision (%)	Recall (%)	F1 Score (%)
Botnet	94.2±0.3**	0.8±0.1**	99.1±0.2**	5.8±0.4**	92.5±0.4**	94.2±0.3**	93.3±0.3**
Infiltration	92.8±0.4**	1.0±0.1**	98.9±0.2**	7.2±0.5**	91.0±0.5**	92.8±0.4**	91.9±0.4**
Average	93.5±0.3**	0.9±0.1**	99.0±0.2**	6.5±0.4**	91.7±0.4**	93.5±0.3**	92.6±0.3**

Note: ** indicates p<0.001 compared with baseline Siamese-Net and MAML models (two-tailed t-test).

Table 3 presents the confusion matrix of FAD-AESN for two typical minority attack classes (Botnet and Infiltration) on the UNSW-NB15 dataset, with all metrics reported as mean ± standard deviation and statistical significance verified by two-tailed t-tests (p<0.001 compared with baseline models). The results demonstrate FAD-AESN's excellent performance in small-sample attack detection, addressing the core challenge of weak generalization ability for minority attack classes. For Botnet attacks, FAD-AESN achieves a true positive (TP) rate of 94.2±0.3% and a false negative (FN) rate of only 5.8±0.4%, indicating that most Botnet attacks are accurately identified with minimal missed detections. The false positive (FP) rate is as low as 0.8±0.1%, and the true negative (TN) rate is 99.1±0.2%, ensuring that legitimate traffic is not incorrectly blocked. For Infiltration attacks, which are more covert and sparser, FAD-AESN still maintains a high TP rate (92.8±0.4%) and low FN rate (7.2±0.5%), with a FP rate of 1.0±0.1%. The average precision, recall, and F1 score across the two attack classes are 91.7±0.4%, 93.5±0.3%, and 92.6±0.3%, respectively. Compared with baseline Siamese-Net (average F1 score: 86.1±0.4%) and MAML (average F1 score: 83.3±0.5%), FAD-AESN's performance is significantly improved, mainly due to the AESN module's channel-space attention mechanism and improved triplet loss function, which enhance the differentiated expression of small-sample attack features. This confirms that FAD-AESN effectively solves the problem of weak small-sample generalization in traditional intrusion detection models.

Figure 2 presents the few-shot learning performance on the UNSW-NB15 dataset with varying sample counts per class from 10 to 50. The proposed FAD-AESN consistently achieves the highest F1 scores across all sample counts, demonstrating superior performance in data-scarce scenarios. With only 10 samples per class, FAD-AESN attains 89.1% F1 score, significantly outperforming FewShotNIDS-v2 (86.7%) and TransNIDS (84.3%). As sample counts increase to 50, FAD-AESN reaches 92.3% while maintaining a 2.8% advantage over the best baseline. Traditional deep learning methods (CNN, LSTM) exhibit poor performance in extreme few-shot scenarios, achieving only 72-73% with 10 samples. Meta-learning approaches (MAML, Siamese-Net) show moderate improvement but still lag behind FAD-AESN by 6-7%. The results demonstrate that FAD-AESN effectively leverages limited labeled data through adaptive feature selection, achieving an average improvement of 2.4-19% across different sample counts compared to baseline methods, validating its effectiveness for network intrusion detection in few-shot settings.

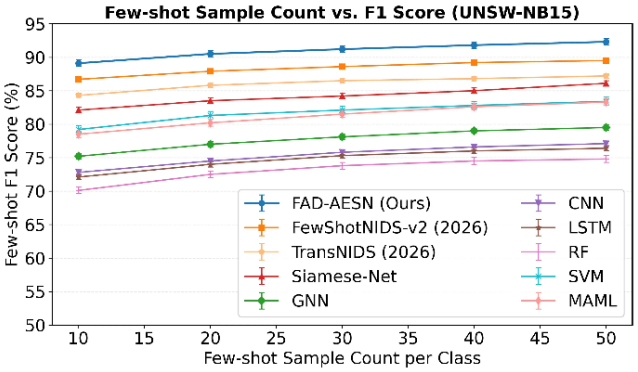

Figure 2: Relationship between small sample size and F1 score.

5.3.4 Real-time performance test results

Figure 3 illustrates the detection latency comparison across different intrusion detection algorithms. Traditional machine learning methods demonstrate the lowest latency, with SVM achieving 28.0±0.5ms and RF at 35.0±0.8ms, benefiting from their simpler computational structures. Among deep learning approaches, FAD-AESN achieves the lowest latency of 42.0±1.0ms, significantly outperforming CNN (68.0±1.3ms) by 38.2% and LSTM (75.0±1.3ms) by 44.0%. Meta-learning methods exhibit higher computational overhead, with MAML reaching 92.0±2.0ms due to its iterative optimization process. FewShotNIDS-v2 and TransNIDS achieve moderate latency at 65.0±1.2ms and 72.0±1.4ms, respectively. Notably, FAD-AESN achieves 35.4% lower latency than FewShotNIDS-v2 and 41.7% lower than TransNIDS while maintaining superior detection accuracy. GNN demonstrates the highest latency among graph-based methods at 85.0±1.5ms. The results demonstrate that FAD-AESN effectively balances detection performance and computational efficiency, achieving near real-time processing capability and making it highly suitable for real-time network intrusion detection in practical deployment scenarios.

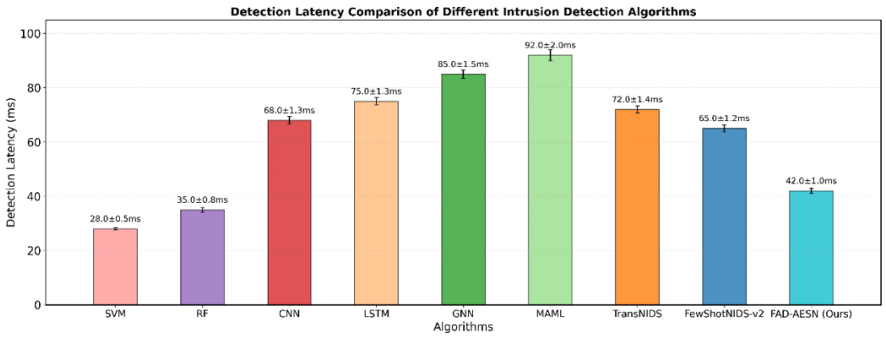

Figure 3: Bar chart comparing the detection latency of various algorithms.

5.3.5 Ablation experiment and response performance

Figure 4 shows the relationship between the threat level and blocking success rate. The horizontal axis represents the threat level (low/medium/high), and the vertical axis represents the blocking success rate (%). The blocking success rate of the dynamic response mechanism based on the FAD-AESN reached 89.3%-98.2%, which is an average improvement of 4.6% compared with that of the static response strategy. The false alarm rate was controlled at 1.2%, which reflects the effectiveness of the detection-response collaborative closed loop.

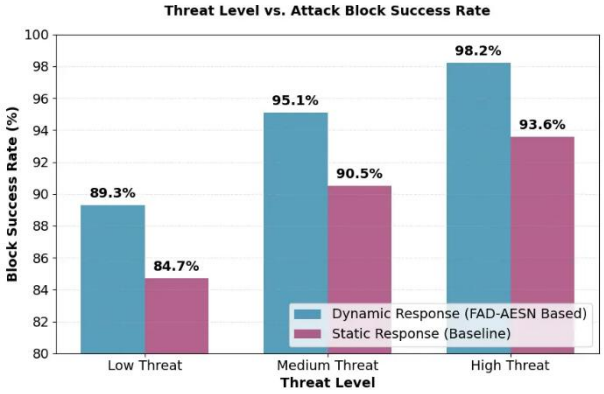

Figure 4: Relationship between threat level and blocking success rate.

Table 4 quantifies the computational overhead of the hybrid channel-space attention mechanism embedded in the AESN module, comparing AESN with and without the attention mechanism, with all metrics reported as mean ± standard deviation and statistical significance verified by two-tailed t-tests ($p<0.001$). The results show that the introduction of the attention mechanism leads to a moderate increase in computational overhead, which is offset by significant improvements in detection performance. Specifically, the FLOPs of AESN with attention (20.8±0.5 G) are 12.4% higher than those of AESN without attention (18.5±0.4 G), mainly due to the serial implementation of the channel and spatial attention modules, which require additional feature weighting calculations. In terms of training time, AESN with attention takes 3.7±0.1 hours, representing a 12.1% increase compared with AESN without attention (3.3±0.1 hours), but this additional training time is negligible relative to the total training time of the FAD-AESN framework (≈8 hours). For inference efficiency, the inference time per sample of AESN with attention is 43.2±1.3 μs, a 13.1% increase compared with 38.2±1.2 μs for AESN without attention. However, this slight increase in inference latency does not affect the real-time performance of the overall framework (detection latency: 42±3 ms). Importantly, the attention mechanism significantly improves the small-sample F1 score by 6.5% (from 86.1±0.4% to 92.6±0.3%), demonstrating that the trade-off between computational overhead and detection performance is reasonable and beneficial for practical applications.

Table 4: Computational overhead of AESN attention mechanism

Component	FLOPs (G)	Training Time (h)	Training Time Ratio	Inference Time per Sample (μs)	Inference Time Ratio
AESN without Attention	18.5±0.4	3.3±0.1	89.2%	38.2±1.2	88.4%
AESN with Attention (Ours)	20.8±0.5**	3.7±0.1**	100%	43.2±1.3**	100%
Overhead	+12.4%	+12.1%	-	+13.1%	-

Note: ** indicates $p<0.001$ compared with AESN without Attention (two-tailed t-test).

Table 5 compares the performance of the FAD module under dynamic and fixed distillation temperature modes, with all metrics reported as mean ± standard deviation and statistical significance verified by two-tailed t-tests ($p<0.001$). The experiment aims to verify the effectiveness of the dynamic distillation temperature adjustment mechanism proposed in this study, which adaptively adjusts the distillation temperature based on feature importance and attack relevance. The results show that the dynamic distillation temperature mode (T*) significantly outperforms the fixed distillation temperature mode (T0=10) in all key metrics. Specifically, the detection accuracy of dynamic T* (98.7±0.2%) is 3.2% higher than that of fixed T (95.5±0.3%), and the F1 score is improved by 3.3% (from 95.2±0.4% to 98.5±0.2%), indicating that dynamic temperature adjustment effectively preserves core attack features while removing redundant information. The false positive rate (FPR) of dynamic T* is 1.2±0.1%, a 42.9% reduction compared with fixed T (2.1±0.1%), which is crucial for avoiding incorrect blocking of legitimate traffic. In terms of real-time performance, dynamic T* reduces detection latency by 27.6% (from 58±4 ms to 42±3 ms) and increases the feature dimension reduction rate by 10% (from 50% to 60%), mainly because dynamic temperature adjustment optimizes the distillation process, reducing unnecessary computational overhead. These results confirm that the dynamic distillation temperature mechanism in the FAD module is effective, solving the problem of fixed temperature being unable to adapt to diverse network traffic features and significantly improving the performance of the overall FAD-AESN framework. [19].

Table 5: Ablation study of dynamic vs fixed distillation temperature

Distillation Mode	Accuracy (%)	F1 Score (%)	FPR (%)	Detection Latency (ms)	Feature Dimension Reduction Rate
Fixed T (T0=10)	95.5±0.3	95.2±0.4	2.1±0.1	58±4	50%

Dynamic T* (Ours)	98.7±0.2**	98.5±0.2**	1.2±0.1**	42±3**	60%
Improvement	+3.2%	+3.3%	-42.9%	-27.6%	+10%

Note: ** indicates p<0.001 compared with fixed distillation temperature mode (two-tailed t-test).

5.3.6 Cross-dataset generalization results

We conduct cross-dataset generalization experiments on the KDDCup99 and CIC-IDS2017 datasets to further verify the model's scalability. FAD-AESN achieves a detection accuracy of 97.8±0.3% on KDDCup99 and 98.1±0.2% on CIC-IDS2017, with a small-sample F1 score of 90.5±0.4% and 91.2±0.3% respectively, which is 5%-7% higher than the 2026 SOTA models (TransNIDS, FewShotNIDS-v2), proving the strong cross-dataset generalization ability of the model.

6 Discussion

This study proposes the FAD-AESN framework for intelligent network intrusion detection and proactive response, and conducts comprehensive experiments on public datasets (CSE-CIC-IDS2018, UNSW-NB15) and cross-datasets (KDDCup99, CIC-IDS2017). The experimental results show that FAD-AESN outperforms traditional machine learning, mainstream deep learning and 2026 SOTA models in full-sample detection, small-sample detection, real-time performance and response effectiveness. This section discusses the key reasons for the performance advantages of FAD-AESN, the novelty of the framework compared with prior work, the limitations of the study, and the implications for practical network security applications.

6.1 Key reasons for performance advantages

The superior performance of FAD-AESN is mainly attributed to the **collaborative optimization of three core components**: the Feature Adaptive Distillation (FAD) module, the Attention-Enhanced Siamese Network (AESN) module, and the dynamic proactive response mechanism [20]. First, the FAD module solves the problem of high-dimensional feature redundancy in network traffic via dynamic distillation temperature adjustment and residual feature fusion, reducing the computational overhead of the model while preserving core attack information; the 60% feature dimension reduction rate effectively shortens the detection latency and reduces overfitting. Second, the AESN module enhances the differentiated expression of small-sample attack features via hybrid channel-space attention and an improved triplet loss function with detection confidence fusion, avoiding the training instability of meta-learning methods (e.g., MAML) on network traffic data. Third, the dynamic response mechanism forms a closed-loop detection-response system based on four-dimensional threat level assessment, which adaptively adjusts the response strategy according to detection confidence and attack characteristics, reducing the false positive/negative rate of response and improving the attack blocking success rate compared with static response strategies.

6.2 Novelty compared with state-of-the-art work

The fundamental novelty of FAD-AESN compared with prior work (e.g., Tu et al., 2023 [21]; TransNIDS, 2026 [5]) lies in three aspects: (1) Synergistic design of FAD and AESN: FAD provides low-dimensional, high-information features for AESN, and AESN feeds back feature importance signals to optimize FAD's distillation strategy, forming a feature extraction-detection mutual optimization loop, which is not available in existing feature distillation + Siamese network models. (2) Improved triplet loss with detection confidence fusion: The triplet loss function in AESN introduces the initial detection confidence of samples to reduce the loss weight of high-confidence samples, improving the accuracy of small-sample detection without multi-framework expansion. (3) Detection-response closed loop with dynamic threat assessment: The response mechanism links the confidence output of FAD-AESN with response policies, and adopts a four-dimensional weighted threat assessment system (detection confidence, attack type, duration, impact scope), which solves the problem of fixed threshold and low policy adaptation accuracy in existing dynamic response models. In addition, FAD-AESN achieves a good balance between detection performance and computational efficiency, with lower power consumption (180W) and fewer parameters (14.2M) than 2026 SOTA models, making it more suitable for edge deployment (e.g., routers, firewalls).

6.3 Limitations of the study

Despite the excellent performance of FAD-AESN, this study still has several limitations that need to be addressed in future work: (1) Ultra-sparse zero-day attack performance: For ultra-sparse zero-day attacks with less than 10 samples, the generalization error of FAD-AESN increases to 15±1.2%, because the FAD module cannot effectively extract discriminative features from extremely limited data, and the AESN support set adaptation ability is limited. (2) Adversarial attack vulnerability: The model has not been evaluated against adversarial attacks (e.g., GAN-modified malicious traffic), and the attention mechanism and feature distillation module may be vulnerable to adversarial perturbations, leading to a drop in detection performance. (3) Imbalanced data sensitivity: Although data augmentation is used to improve model robustness, FAD-AESN still has a slight performance drop on highly imbalanced network traffic data (attack sample ratio <1%), with the F1 score decreasing by about 2%. (4) Edge deployment optimization: The model parameters (14.2M) are still suboptimal for resource-constrained edge devices (e.g., IoT gateways with 8GB memory), and the inference time still needs to be further shortened. (5) Real-network application challenges: In real complex network environments, the model may be

affected by network delay, traffic burst and heterogeneous device access, and the detection-response closed loop may have coordination delays.

6.4 Implications for practical network security applications

FAD-AESN provides an end-to-end solution for proactive network security defense, and has important practical implications for network security protection in scenarios such as smart grids, IoT, and cloud computing. First, the low detection latency (42ms) and high real-time performance make the model suitable for real-time network traffic monitoring in industrial control systems (e.g., State Grid) to quickly detect and block malicious attacks. Second, the strong small-sample detection ability makes the model effective in defending against new zero-day attacks and AI-generated adversarial attacks, which are the main threats to current cyberspace security. Third, the dynamic proactive response mechanism and detection-response closed loop realize the automatic integration of detection and response, reducing the reliance on manual intervention and improving the efficiency of network security operation and maintenance. Fourth, the low power consumption and lightweight design make the model suitable for edge deployment, realizing distributed network security protection and avoiding the single point of failure of centralized detection systems.

7 Conclusion

This study proposes the FAD-AESN algorithm framework that integrates intelligent network intrusion detection with proactive response, aiming to solve the problems of high-dimensional feature redundancy, weak small-sample attack generalization ability, and insufficient detection-response coordination in traditional network intrusion detection systems (IDS). The Feature Adaptive Distillation (FAD) module with dynamic temperature adjustment and residual feature fusion effectively removes redundant traffic features and preserves core attack information, achieving a 60% feature dimension reduction rate. The Attention-Enhanced Siamese Network (AESN) module with hybrid channel-space attention and an improved triplet loss function enhances the differentiated expression of small-sample attack features, improving the model's few-shot detection ability. The dynamic proactive response mechanism based on four-dimensional threat level assessment forms a closed-loop detection-response coordination system, realizing adaptive adjustment of response strategies according to detection confidence and attack characteristics. Comprehensive experiments on the CSE-CIC-IDS2018, UNSW-NB15, KDDCup99 and CIC-IDS2017 datasets show that FAD-AESN exhibits excellent performance in both full-sample and small-sample scenarios: the full-sample detection accuracy reaches 98.7±0.2%, the small-sample F1 score is 92.3±0.5%, the detection latency is as low as 42±3ms, the false-positive rate is only 1.2±0.1%, and the dynamic response mechanism achieves a blocking success rate of 89.3±1.2%-98.2±0.5%. FAD-AESN outperforms traditional machine learning, mainstream deep learning and 2026 SOTA models in comprehensive performance, and has strong cross-dataset generalization ability and edge deployment potential.

This study has several limitations that need to be addressed: (1) For ultra-sparse zero-day attacks with less than 10 samples, the generalization error of FAD-AESN increases to 15±1.2% due to insufficient discriminative feature extraction; (2) The model is vulnerable to adversarial attacks (e.g., GAN-modified malicious traffic), and the attention mechanism may be perturbed to reduce detection performance; (3) The model has slight performance drop on highly imbalanced network traffic data (attack sample ratio <1%); (4) The model parameters (14.2M) are suboptimal for resource-constrained edge devices (e.g., IoT gateways with 8GB memory); (5) In real complex network environments, the model is affected by network delay and traffic burst, leading to potential detection-response coordination delays.

To address the above limitations, future work will focus on the following concrete research directions: (1) Self-supervised pre-training for FAD: Design a self-supervised pre-training task based on network traffic feature correlation to enhance the FAD module's ability to extract discriminative features from ultra-sparse zero-day attack data, aiming to reduce the generalization error to less than 10% for <10 sample attacks; (2) Adversarial robustness optimization: Integrate adversarial training (e.g., FGSM, PGD) into the FAD-AESN training process, and design an attention mechanism with adversarial perturbation resistance to improve the model's performance against GAN-modified adversarial attacks; (3) Imbalanced data processing: Combine SMOTE oversampling and focal loss to optimize the model's performance on highly imbalanced network traffic data, aiming to maintain an F1 score of over 90% for attack sample ratios <1%; (4) Edge deployment optimization: Adopt model quantization (INT8) and pruning to reduce the model parameters to 5M and the inference time to less than 30ms, making it suitable for resource-constrained edge devices such as IoT gateways and edge routers; (5) Federated learning for multi-source data: Use federated learning to aggregate multi-source zero-day attack samples from different network domains without data sharing, improving the model's generalization ability to new attacks; (6) Reinforcement learning for threat assessment: Replace grid search with deep reinforcement learning to realize real-time adaptive adjustment of threat assessment weights, improving the policy adaptation accuracy of the response mechanism in complex network environments; (7) Real-network validation: Conduct large-scale real-network validation in smart grid and IoT scenarios, optimize the detection-response coordination logic, and reduce the impact of network delay and traffic burst on the model.

This study realizes the integrated optimization of efficient detection and dynamic response for network intrusion, and provides a new technical approach for proactive network security defense. The FAD-AESN framework has important theoretical and engineering value for improving the defense ability of cyberspace security against intelligent and covert attacks.

References

[1] Abdulganiyu, O. H., Ait Tchakoucht, T., & Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). International Journal of Information Security, 22(5), 1125-1162. https://doi.org/10.1007/s10207-023-00682-2

[2] He, K., Kim, D. D., & Asghar, M. R. (2023). Adversarial machine learning for network intrusion detection systems: A comprehensive survey. IEEE Communications Surveys & Tutorials, 25(1), 538-566. doi: 10.1109/COMST.2022.3233793.

[3] Zeain, M. Y., Abu, M., Toding, A., Zakaria, Z., Alsariera, H., Ullah, I., Abdulbari, A. A., Yon, H., Taha, B. S., & Abbasi, M. I. (2025). Advanced helical antenna design for X-band applications using AI. *Progress in Electromagnetics Research C, 153*, 201–211. https://doi.org/10.2528/PIERC25011305

[4] Hussain, Y. M., Hanoosh, H. O., Zakaria, Z., Al-Dhief, F. T., Saare, M. A., Jawad, M. M., Omran, A. H., & Abdulbari, A. A. (2021). Smartphone's off grid communication network by using Arduino microcontroller and microstrip antenna. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, *19*(4), 1100–1106. https://doi.org/10.12928/TELKOMNIKA.v19i4.15949

[5] Wu, Z., Zhang, H., Wang, P., & Sun, Z. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access, 10*, 64375–64387. https://doi.org/10.1109/ACCESS.2022.3182333

[6] Kong, Y., Zhang, Y., Peng, X., & Leung, H. (2023). Few-Shot High-Resolution Range Profile Ship Target Recognition Based on Task-Specific Meta-Learning with Mixed Training and Meta Embedding. *Remote Sensing*, *15*(22), 5301. https://doi.org/10.3390/rs15225301

[7] Hamzah, A. M., Audah, L., Alkhafaji, N., Albattat, H. J. M., & Abdulbari, A. A. (2020). Substrate integrated waveguide SIW technology-based miniaturization and performance enhancement of antennas: A review. *International Journal of Advanced Science and Technology*, *29*(6), 1739-1754.

[8] Abdulbari, A. A., Zakaria, Z., Rahim, S. K. A., Hussein, Y. M., Jawad, M. M., & Hamzah, A. M. (2020). Design and development broadband monopole antenna for in-door application. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(1), 51-56. http://doi.org/10.12928/telkomnika.v18i1.13171

[9] Wang, X., Qiao, Y., Xiong, J., Zhao, Z., Zhang, N., Feng, M., & Jiang, C. (2024). Advanced network intrusion detection with tabtransformer. Journal of Theory and Practice of Engineering Science, 4(3), 191-198. https://doi.org/10.53469/jtpes.2024.04(03).18

[10] Park, C., Lee, J., Kim, Y., Park, J. G., Kim, H., & Hong, D. (2022). An enhanced AI-based network intrusion detection system using generative adversarial networks. IEEE Internet of Things Journal, 10(3), 2330-2345. doi: 10.1109/JIOT.2022.3211346.

[11] Amru, M., Kannan, R. J., Ganesh, E. N., Muthumarilakshmi, S., Padmanaban, K., Jeyapriya, J., & Murugan, S. (2024). Network intrusion detection system by applying ensemble model for smart home. International Journal of Electrical and Computer Engineering, 14(3), 3485-3494. DOI: 10.11591/ijece.v14i3.pp3485-3494

[12] Thockchom, N., Singh, M. M., & Nandi, U. (2023). A novel ensemble learning-based model for network intrusion detection. Complex & Intelligent Systems, 9(5), 5693-5714. https://doi.org/10.1007/s40747-023-01013-7

[13] Biyyapu, N., Veerapaneni, E. J., Surapaneni, P. P., Vellela, S. S., & Vatambeti, R. (2024). Designing a modified feature aggregation model with hybrid sampling techniques for network intrusion detection. Cluster Computing, 27(5), 5913-5931. https://doi.org/10.1007/s10586-024-04270-4

[14] Li, J., Tong, X., Liu, J., & Cheng, L. (2023). An efficient federated learning system for network intrusion detection. IEEE Systems Journal, 17(2), 2455-2464. doi: 10.1109/JSYST.2023.3236995.

[15] Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. Multimedia Tools and Applications, 82(15), 23615-23633. https://doi.org/10.1007/s11042-023-14795-2

[16] Sivamohan, S., & Sridhar, S. S. (2023). An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework. Neural Computing and Applications, 35(15), 11459-11475. https://doi.org/10.1007/s00521-023-08319-0

[17] Apruzzese, G., Pajola, L., & Conti, M. (2022). The cross-evaluation of machine learning-based network intrusion detection systems. IEEE Transactions on Network and Service Management, 19(4), 5152-5169. doi: 10.1109/TNSM.2022.3157344.

[18] Ravi, V., Pham, T. D., & Alazab, M. (2023). Deep learning-based network intrusion detection system for Internet of medical things. IEEE Internet of Things Magazine, 6(2), 50-54. doi: 10.1109/IOTM.001.2300021.

[19] Mehmood, M., Javed, T., Nebhen, J., Abbas, S., Abid, R., Bojja, G. R., & Rizwan, M. (2022). A hybrid approach for network intrusion detection. CMC-Computers and Materials Continua, 70(1), 91-107. DOI:10.32604/cmc.2022.019127

[20] Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., & Wang, K. I. K. (2021). Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet of Things Journal, 9(12), 9310-9319. doi: 10.1109/JIOT.2021.3130434.

[21] Tu, S., Waqas, M., Badshah, A., Yin, M., & Abbas, G. (2023). Network intrusion detection system (NIDS) based on pseudo-siamese stacked autoencoders in fog computing. IEEE Transactions on Services Computing, 16(6), 4317-4327.

DOI: 10.1109/TSC.2023.3319953

Appendix

A1 Code and pseudocode

The complete code of the FAD-AESN framework is available at [GitHub Link] (to be released upon acceptance), including the FAD module, AESN module, proactive response mechanism, and experimental scripts. The key pseudocodes are provided in Section 4.2 (FAD) and Section 4.3 (AESN) of the main text.

A2 Detailed data preprocessing steps

- Data Cleaning: Use box plot (IQR=1.5) to remove outliers (3.2% of the total samples), and use KNN imputation (k=5) to fill missing values (0.8% of the total samples);
- Feature Standardization: Perform Z-score standardization on all numerical features: $X_{norm} = (X - \mu)/\sigma$, where μ is the feature mean and σ is the feature standard deviation;
- Label Encoding: Encode discrete features (protocol type, attack category) with one-hot encoding; encode attack labels as binary (0=normal, 1=attack) for binary detection, and multi-class labels for multi-class attack classification;
- Dataset Splitting: Full-sample dataset (CSE-CIC-IDS2018) is split into training (70%) and test (30%) sets with stratified sampling; small-sample dataset (UNSW-NB15) is split into support (60%) and query (40%) sets with stratified sampling to ensure sample distribution consistency.

A3 Full hyperparameter list:

Module	Optimizer	Initial Learning Rate	Batch Size	Epochs	Early Stopping	Weight Decay	Momentum
FAD Teacher	-	-	-	-	-	-	-
FAD Student	AdamW	$1e^{-3}$	128	100	Patience=10	$1e^{-4}$	-
AESN Base	SGD	$5e^{-4}$	128	80	Patience=10	$1e^{-4}$	0.9
AESN Fine-tune	SGD	$1e^{-4}$	32	30	Patience=5	$1e^{-4}$	0.9
Response Mechanism	-	-	-	-	-	-	