

IoT Anomaly Traffic Detection Method Based on Transformer Encoder Architecture

Yuxuan Jiang

Qingdao University, Qingdao 266071, China

Email: jiangyuxuan@qdu.edu.cn

How to cite this paper: Jiang, Y. X. (2026). IoT anomaly traffic detection method based on Transformer encoder architecture. Journal of Computer Science and Frontier Technologies, 3(2), 177–186. ISSN Print: 3104-4204, ISSN Online: 3104-4212.

<https://doi.org/10.63313/JCSFT.9081>

Published: 2026-05-30

Copyright © 2026 by author(s) and Erytis Publishing Limited.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Abstract

The widespread deployment of IoT devices has made heterogeneous network traffic highly complex, making abnormal traffic detection vital for IoT security. This paper presents a Transformer-based IoT abnormal traffic detection method tailored to the long-range temporal correlation of IoT traffic and the drawbacks of existing approaches. It first builds traffic embeddings with learnable positional encoding to project raw traffic features into a high-dimensional space, then leverages Transformer multi-head self-attention to capture long-range dependencies by modeling temporal correlations in traffic sequences. Feature compression and residual enhancement modules are further adopted to lower model complexity and inference delay for resource-limited IoT terminals. Experiments verify that the scheme accurately identifies normal traffic and diverse anomalous attacks with outstanding detection accuracy and stability.

Keywords

IoT Security; Anomaly Network Traffic Detection; Transformer Encoder; Self-Attention Mechanism

1. Introduction

With the rapid development of Internet of Things (IoT) technology, various sensing devices, intelligent terminals and embedded systems have been widely deployed in multiple fields such as smart cities, industrial Internet and smart homes, leading to a sharp surge in the number of network-connected devices. IoT devices are generally characterized by large quantities, diverse types, and limited computing and storage resources. Their communication behaviors differ significantly from those of traditional Internet networks, which not only complicates the overall structure of network traffic but also raises higher requirements for traffic monitoring and security protection.

2. Related Works

Early studies predominantly employed rule-based or statistical feature approaches. For instance, Raza et al. [1] proposed a lightweight statistical feature-based method for IoT intrusion detection, which identifies anomalous communication behaviors by analyzing features such as packet frequency and session duration. Addressing the issue of IoT botnets, Doshi et al. [2] analyzed the traffic characteristics of Mirai attacks and proposed a detection method based on traffic rates and connection behaviors.

Subsequently, researchers began to introduce artificial intelligence (AI) techniques into the field of IoT traffic detection. Meidan et al. [3] proposed a Random Forest-based method for IoT device identification and anomaly detection, modeling different types of IoT devices by extracting features of their communication behaviors. Ammar et al. [4] utilized Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) algorithms to detect attack traffic in IoT networks. Alauthman et al. [5] proposed a Deep Neural Network (DNN)-based IoT intrusion detection method that enhances the recognition capability for various attack types through automatic feature learning. Shone et al. [6] introduced an anomaly detection model based on stacked autoencoders, utilizing unsupervised learning to identify abnormal traffic in IoT networks, thereby reducing the dependency on labeled data.

3. Preliminary

3.1. Network Traffic Detection in IoT Environments

The IoT typically adopts a multi-layered, distributed network architecture, wherein terminal devices access the core network via gateways or edge nodes. This heterogeneity results in significant diversity in data volume, communication frequency, and transmission modes [7]. The corresponding network architecture and traffic characteristics are illustrated in Figure 1.

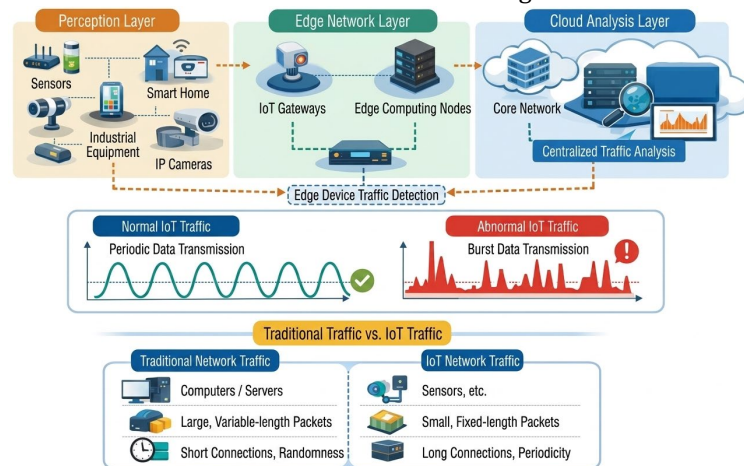


Figure 1. Comparison of IoT Network Architectures and Traffic Characteristics

3.2. Attention Mechanism and Transformer

The Attention Mechanism was introduced into the field of neural networks in 2017 [8] to simulate the human cognitive process of focusing on key information when processing data. Its core principle involves assigning varying weights to input features based on their importance, thereby enhancing the model's capacity to represent critical information. The working mechanism of attention is illustrated in Figure 2.

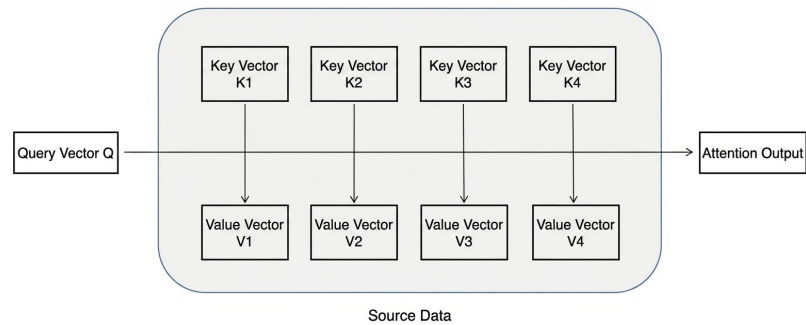


Figure 2. Working Mechanism of the Attention Mechanism

Input features are mapped into a Query vector (Q), a Key vector (K), and a Value vector (V). By calculating the similarity between the Query and Key vectors, the model obtains attention weights for different input positions. These weights are then used to perform a weighted sum of the Value vectors, ultimately generating an output representation fused with global information.

4. Methodology

4.1. Model Architecture

The proposed model consists of four main components: a traffic embedding module, a Transformer encoder module, a feature compression and residual enhancement module, and an anomaly discrimination module. The overall framework is illustrated in Figure 3.

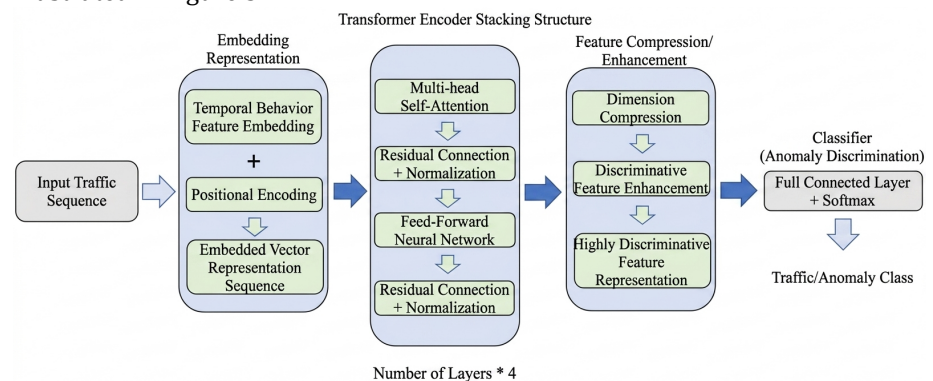


Figure 3. Framework of the IoT Anomaly Traffic Detection Model

The model serializes raw IoT packets into temporal sequences via time windows, leveraging the inherent periodicity of device communication. Traffic features are

mapped to a high-dimensional space by an embedding module, augmented with learnable positional encoding to capture temporal correlations for the Transformer encoder. Using multi-head self-attention, the encoder mines long-range dependencies in parallel, suiting long-sequence IoT scenarios. To accommodate resource-constrained terminals, integrated compression and residual modules reduce complexity and latency while preserving discriminative features. Finally, fused high-level features are classified via fully connected layers, enabling both binary detection and fine-grained identification of various attacks to meet diverse IoT security needs.

4.2. Traffic Embedding Representation

This section elaborates on the embedding representation method for IoT traffic features, providing unified and high-quality inputs for subsequent temporal modeling by the Transformer model. The integration process of IoT traffic embedding and positional encoding is illustrated in Figure 4.

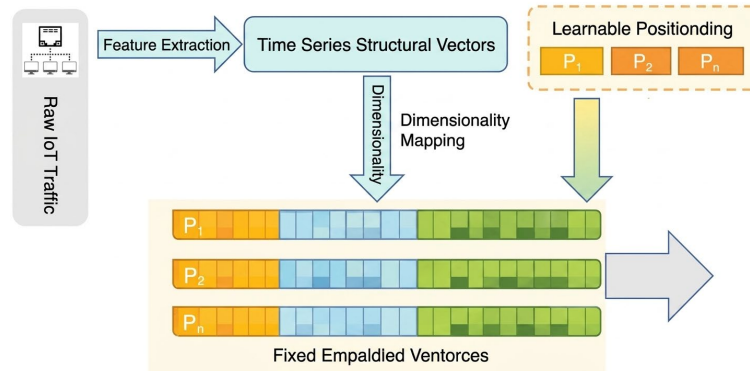


Figure 4. Integration Process of Traffic Embedding and Positional Encoding

Each preprocessed traffic unit feature vector is denoted as $x_t \in R^d$. It is mapped to a high-dimensional embedding vector $e_t \in R^D$ via a learnable embedding matrix W_e , where $D=128$ represents the embedding dimension. This scheme incorporates learnable positional encoding into the traffic embedding representation to explicitly model the temporal order of traffic sequences. The implementation details are as follows:

(1) Encoding Form: For a traffic sequence length T (set to $T=32$, representing 32 time steps per sequence in this work), a positional encoding matrix $P \in R^{T \times D}$ is constructed, matching the embedding dimension ($D=128$). Each position i ($0 \leq i < T$) corresponds to an encoding vector p_i , implemented as learnable parameters optimized jointly during model training.

(2) Fusion Method: The positional encoding vector is added element-wise to the traffic embedding vector, formulated as $e'_t = e_t + p_t$. This enables the model to utilize both traffic feature information and temporal positional cues during self-attention computation.

After constructing the complete traffic embedding sequence, the final embedding

representation $E' = \{e'_1, e'_2, \dots, e'_T\} \in \mathbb{R}^{T \times D}$ serves as the input to the Transformer encoder.

4.3. Transformer Encoder Model

The core component of the proposed model is composed of four identical encoder layers stacked sequentially, as illustrated in Figure 5.

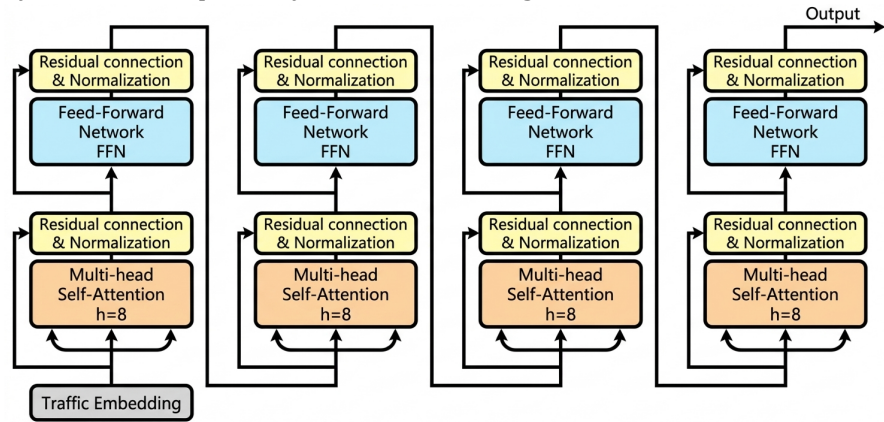


Figure 5. Structure of the Transformer Encoder Feature Extraction Model

The input sequence is represented as $X \in \mathbb{R}^{T \times D}$, where $T=32$ denotes the length of the time steps and $D=128$ represents the feature dimension. In the self-attention module, the input features are linearly mapped to generate the Query, Key, and Value vectors, as shown in Equation (1):

$$Q = XW^Q, K = XW^K, V = XW^V \quad (1)$$

where $W^Q, W^K, W^V \in \mathbb{R}^{D \times d_k}$ are learnable parameter matrices, and d_k denotes the dimension of the Key and Query vectors. Subsequently, scaled dot-product attention is applied to compute the correlations between different time steps, as expressed in Equation (2):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

The Transformer adopts a multi-head self-attention mechanism, where the computation of the i -th attention head is given by Equation (3):

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

The input features are divided into $h=8$ attention heads, each computing its own attention output independently. The outputs of all heads are concatenated and linearly projected back to the original dimension, as shown in Equation (4):

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)W^O \quad (4)$$

where W^O is the output projection matrix. Following the attention module, the FFN applies non-linear transformations to the features at each time step. It consists of two fully connected layers and a ReLU activation function, computed as in Equation (5):

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (5)$$

Both the attention module and the FFN are followed by residual connections and

layer normalization, calculated as in Equation (6):

$$y = \text{LayerNorm}(x + \text{Sublayer}(x)) \quad (6)$$

4.4. Feature Compression and Residual Enhancement

The feature compression and residual enhancement module is designed to reduce model complexity while preserving critical discriminative information. The computational procedure of this module is detailed in Algorithm 1.

Algorithm 1: Feature Compression and Residual Enhancement Algorithm

Input: $H \in \mathbb{R}^{T \times 128}$
Output: $Z_{\text{enhanced}} \in \mathbb{R}^{64}$
Initialization: $W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, W_{\text{proj}}$

- 1: $H_{\text{global}} \leftarrow \text{GlobalAvgPooling}(H)$
- 2: $z^{(0)} \leftarrow H_{\text{global}}$
- 3: $z^{(1)} \leftarrow \text{ReLU}(W^{(1)} z^{(0)} + b^{(1)})$
- 4: $Z_{\text{compressed}} \leftarrow \text{ReLU}(W^{(2)} z^{(1)} + b^{(2)})$
- 5: $Z_{\text{proj}} \leftarrow W_{\text{proj}} \cdot H_{\text{global}}$
- 6: $Z_{\text{enhanced}} \leftarrow Z_{\text{compressed}} + Z_{\text{proj}}$
- 7: $Z_{\text{enhanced}} \leftarrow \text{LayerNorm}(Z_{\text{enhanced}})$

The temporal features output by the Transformer encoder are first aggregated along the time dimension via global average pooling, yielding a global feature representation $H_{\text{global}} \in \mathbb{R}^{128}$ that characterizes the overall traffic behavior. Subsequently, a two-layer fully connected network maps the high-dimensional features into a compact 64-dimensional latent space: the first layer reduces the dimensionality from 128 to 96, followed by ReLU activation, and the second layer maps it further to 64 dimensions.

Taking the Transformer encoder output $H \in \mathbb{R}^{T \times 128}$ as input, the feature compression module projects high-dimensional features into a compact low-dimensional space through global average pooling and progressive dimensionality reduction. Each compression unit consists of a linear mapping followed by a non-linear activation function, as expressed in Equation (7):

$$z^{(l)} = \sigma(W^{(l)} z^{(l-1)} + b^{(l)}) \quad (7)$$

where $W^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias term of the l -th layer, respectively, and $\sigma(\cdot)$ represents the ReLU activation function.

The 128-dimensional features obtained after global average pooling are linearly projected into a 64-dimensional space, aligning them with the dimensionality of the compressed features. These are then combined via element-wise addition to form the enhanced feature representation, as described in Equation (8):

$$Z_{\text{enhanced}} = Z_{\text{compressed}} + W_{\text{proj}} H_{\text{global}} \quad (8)$$

where $W_{\text{proj}} \in \mathbb{R}^{64 \times 128}$ is the projection matrix. The output of the compression layer depends not only on the non-linear mapping results but also on the fusion with

the original input features via residual connections, followed by normalization to stabilize training, as shown in Equation (9):

$$Z_{\text{enhanced}} = \text{LayerNorm}(Z_{\text{compressed}} + W_{\text{proj}}H_{\text{global}}) \quad (9)$$

5. Experiment

5.1. Experimental Datasets

The Bot-IoT dataset is selected as the primary data source for experimental evaluation [9]. Constructed by the cybersecurity research team at the University of New South Wales (UNSW), Australia, it stands as one of the most widely adopted and representative large-scale public datasets in the field of IoT anomaly traffic detection.

5.2. Experimental Setup and Model Parameters

The core architecture of the proposed model consists of four stacked Transformer encoder layers. Each encoder layer incorporates an 8-head multi-head self-attention mechanism (with a single-head dimension of 16) and a feed-forward neural network (dimension transformation: $256 \rightarrow 128$, utilizing ReLU activation and layer normalization).

Following the Transformer encoders, a global average pooling layer and a two-layer linear compression structure (reducing dimensionality from 128 to 64 with ReLU activation) are introduced. Furthermore, a residual enhancement mechanism—comprising linear projection and residual connections—refines the compressed features, reinforcing the representation of critical anomaly-related characteristics while maintaining reduced complexity. The hyperparameter configuration for model training is summarized in Table 1.

Table 1. Hyperparameter Settings for Model Training

Hyperparameters	Settings Value
Training Epochs (Epoch)	50
Batch Size (Batch Size)	64
Loss Function	Cross-Entropy Loss (Cross-Entropy Loss)
Optimizer	AdamW
Initial Learning Rate	0.001
Learning Rate Decay Strategy	Cosine Annealing Learning Rate Scheduler Strategy

5.3. Experimental Results and Analysis

The variation of accuracy and loss during model training is illustrated in Figure 6.

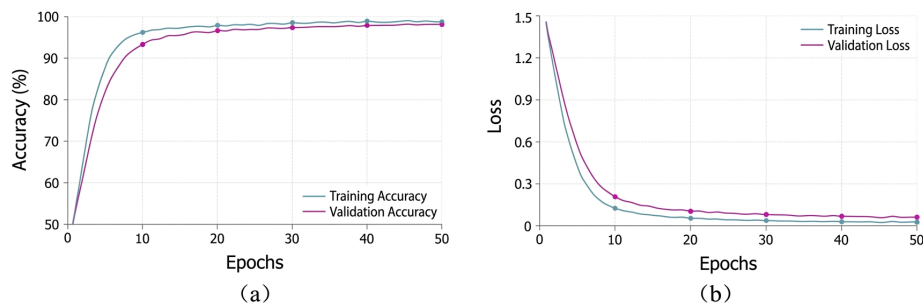


Figure 6. Accuracy and loss curves during model training: (a) Accuracy; (b) Loss

The model demonstrates a rapid learning rate during the early stages of training, with both training and validation accuracies increasing sharply and losses decreasing substantially within the first 10 epochs. Between epochs 10 and 40, the model enters a convergence phase, with accuracy gradually stabilizing. Full convergence is achieved by approximately epoch 50. Notably, no significant divergence is observed between the training and validation curves for either accuracy or loss, indicating sufficient training without underfitting or overfitting, and demonstrating a stable and efficient training process.

(I) Analysis of Detection Results

The detection results for various categories of IoT anomalous traffic in the dataset are presented in Table 2.

Table 2. Detection Results for Different Categories of IoT Traffic Anomalies

Traffic Category	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Normal Traffic	99.1	99.8	99.0	98.9
DDoS	99.7	99.8	99.5	99.6
DoS	99.6	99.4	99.2	99.3
OS Scan	98.4	98.3	98.1	98.2
Service Scan	98.5	98.6	98.3	98.4
Keylogging	98.9	98.9	98.3	98.5
Data Theft	98.8	98.5	98.2	98.3

Experimental results indicate that the model maintains high detection performance across all traffic categories, achieving an overall average accuracy of 98.8%, precision of 98.8%, recall of 98.6%, and F1-score of 98.7%, reflecting strong performance in the multi-class anomaly traffic detection task.

The model demonstrates exceptional performance against high-volume attacks like DDoS and DoS, achieving 99.6% accuracy for both (with a 99.3% F1-score for DoS) by effectively capturing bursts and high concurrency. It also excels at detecting covert threats such as Data Theft and Keylogging, maintaining F1-scores above 98.3% and 98.5%, respectively, by identifying subtle pattern changes rather than volume spikes. Furthermore, unlike traditional methods prone to misclassifying intermittent behaviors, the model stably detects OS and Service Scans with 98.4% and 98.5% accuracy (F1>98%), proving its robustness across diverse attack vectors.

(II) Comparative Experiment Analysis

To further verify the performance advantages of the proposed model, comparative evaluations were conducted against multiple baseline anomaly detection models, specifically SVM, Random Forest, CNN-LSTM, TS-IDS [10], and MTL-DAE [11]. The comparison results are presented in Table 3.

Table 3. Performance Comparison Between the Proposed Model and Other Models

Model Name	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
SVM	92.4	91.8	90.9	91.4
Random Forest	95.6	95.1	94.7	94.9
CNN-LSTM	97.9	97.8	97.5	97.6
TS-IDS	98.2	98.1	97.9	98.0
MTL-DAE	98.5	98.4	98.2	98.3
Our Model	98.8	98.8	98.6	98.7

Compared with traditional machine learning models (SVM and Random Forest), the proposed model achieves improvements of approximately 3%–7% in accuracy and F1-score. Relative to the CNN-LSTM model, it maintains stable gains of 1%–2% across all four evaluation metrics. TS-IDS, which incorporates a traffic-aware self-supervised learning mechanism to enhance feature representation, demonstrates strong generalization capability in IoT intrusion detection tasks, attaining an F1-score of 98.0%. However, as it relies primarily on packet-level feature learning and does not fully exploit deep temporal dependencies, its performance remains about 0.6% lower than that of the proposed model. The multitask-learning-based deep autoencoder model MTL-DAE leverages feature reconstruction and task-cooperative optimization mechanisms to achieve high detection accuracy (98.5% accuracy) on the Bot-IoT dataset. Nevertheless, due to its relatively limited capacity to characterize long-sequence dynamic variations in attacks with complex temporal dependencies, its overall detection performance is slightly inferior to that of the proposed model.

6. Conclusion

This paper focuses on the challenges posed by prominent long-range temporal dependencies in IoT network traffic and the resource-constrained nature of terminal devices. Addressing the limitations of traditional traffic detection methods in feature representation and temporal modeling, this study proposes an anomaly traffic detection scheme based on Transformer-based long-range dependency modeling. By treating the temporal behavioral characteristics of IoT traffic as the primary research subject, the proposed scheme effectively distinguishes between normal device communication behaviors and various types of attack traffic without relying on plaintext payload parsing.

References

- [1] Raza S, Wallgren L, Voigt T. SVELTE: Real-time intrusion detection in the Internet of Things[J]. *Ad hoc networks*, 2013, 11(8): 2661-2674.

-
- [2] Doshi R, Apthorpe N, Feamster N. Machine learning ddos detection for consumer internet of things devices[C]//2018 IEEE security and privacy workshops (SPW). IEEE, 2018: 29-35.
 - [3] Meidan Y, Bohadana M, Shabtai A, et al. ProfilloT: A machine learning approach for IoT device identification based on network traffic analysis[C]//Proceedings of the symposium on applied computing. 2017: 506-509.
 - [4] Ammar M, Russello G, Crispo B. Internet of Things: A survey on the security of IoT frameworks[J]. Journal of information security and Applications, 2018, 38: 8-27.
 - [5] Alauthman M. P2P bot detection using deep learning with traffic reduction schema[J]. Journal of Theoretical and Applied Information Technology, 2020, 98(15).
 - [6] Shone N, Ngoc T N, Phai V D, et al. A deep learning approach to network intrusion detection[J]. IEEE transactions on emerging topics in computational intelligence, 2018, 2(1): 41-50.
 - [7] Portela A L, Menezes R A, Costa W L, et al. Detection of iot devices and network anomalies based on anonymized network traffic[C]//NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2023: 1-6.
 - [8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
 - [9] Koroniotis N, Moustafa N, Sitnikova E, et al. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset[J]. Future Generation Computer Systems, 2019, 100: 779-796.
 - [10] Nguyen H, Kashef R. TS-IDS: Traffic-aware self-supervised learning for IoT Network Intrusion Detection[J]. Knowledge-Based Systems, 2023, 279: 110966.
 - [11] Dong H, Kotenko I. Multi-task learning for IoT traffic classification: a comparative analysis of deep autoencoders[J]. Future Generation Computer Systems, 2024, 158: 242-254.