

Article

AI-Based Malicious Encrypted Traffic Detection in 5G Data Collection and Secure Sharing

Gang Han ^{1,2,3}, Haohe Zhang ^{2,3,*}, Zhongliang Zhang ¹, Yan Ma ³ and Tiantian Yang ³

¹ School of Management, Hangzhou Dianzi University, Hangzhou 310005, China; hangang668866@xupt.edu.cn (G.H.); zlzhang@hdu.edu.cn (Z.Z.)

² The State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China

³ Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; mayan@stu.xupt.edu.cn (Y.M.); ttyang6120@stu.xupt.edu.cn (T.Y.)

* Correspondence: zhh1008@stu.xupt.edu.cn; Tel.: +86-18165116501

Abstract: With the development and widespread application of network information, new technologies led by 5G are emerging, resulting in an increasingly complex network security environment and more diverse attack methods. Unlike traditional networks, 5G networks feature higher connection density, faster data transmission speeds, and lower latency, which are widely applied in scenarios such as smart cities, the Internet of Things, and autonomous driving. The vast amounts of sensitive data generated by these applications become primary targets during the processes of collection and secure sharing, and unauthorized access or tampering could lead to severe data breaches and integrity issues. However, as 5G networks extensively employ encryption technologies to protect data transmission, attackers can hide malicious content within encrypted communication, rendering traditional content-based traffic detection methods ineffective for identifying malicious encrypted traffic. To address this challenge, this paper proposes a malicious encrypted traffic detection method based on reconstructive domain adaptation and adversarial hybrid neural networks. The proposed method integrates generative adversarial networks with ResNet, ResNeXt, and DenseNet to construct an adversarial hybrid neural network, aiming to tackle the challenges of encrypted traffic detection. On this basis, a reconstructive domain adaptation module is introduced to reduce the distribution discrepancy between the source domain and the target domain, thereby enhancing cross-domain detection capabilities. By preprocessing traffic data from public datasets, the proposed method is capable of extracting deep features from encrypted traffic without the need for decryption. The generator utilizes the adversarial hybrid neural network module to generate realistic malicious encrypted traffic samples, while the discriminator achieves sample classification through high-dimensional feature extraction. Additionally, the domain classifier within the reconstructive domain adaptation module further improves the model's stability and generalization across different network environments and time periods. Experimental results demonstrate that the proposed method significantly improves the accuracy and efficiency of malicious encrypted traffic detection in 5G network environments, effectively enhancing the detection performance of malicious traffic in 5G networks.



Academic Editor: Christos J. Bouras

Received: 28 October 2024

Revised: 14 December 2024

Accepted: 15 December 2024

Published: 26 December 2024

Citation: Han, G.; Zhang, H.; Zhang, Z.; Ma, Y.; Yang, T. AI-Based Malicious Encrypted Traffic Detection in 5G Data Collection and Secure Sharing. *Electronics* **2025**, *14*, 51. <https://doi.org/10.3390/electronics14010051>

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: deep learning; adversarial generative networks; hybrid neural networks; encrypt malicious traffic; domain adaptation

1. Introduction

With the rapid development of 5G technology, data collection and sharing have been widely applied across various fields, including smart cities, autonomous driving, and the industrial Internet of Things (IIoT). These 5G networks provide higher bandwidth, lower latency, and broader connectivity, significantly facilitating real-time data collection and fast transmission. However, alongside the widespread adoption of 5G networks, data security issues have become increasingly pressing. A vast amount of sensitive data collected and shared within 5G networks faces potential cyber threats, especially with the growing prevalence of malicious encrypted traffic, which traditional security measures struggle to address effectively. To ensure data security in 5G networks and maintain the integrity, confidentiality, and availability of collected and shared data, the introduction of more intelligent detection mechanisms is crucial. AI-based malicious encrypted traffic detection technologies offer a key solution to this challenge, effectively addressing the security risks posed by encrypted communications and providing stronger protection for data collection and sharing in 5G networks.

Malicious traffic detection is a key research area in 5G data collection and sharing, widely applied to enhance the quality of data traffic. Traditional malicious traffic detection mainly relies on conventional machine-learning techniques. Khan, S.H. et al. [1] proposed a malicious traffic detection framework based on ensemble learning, which combines statistical features and sequential features to describe network traffic, enhancing representation capabilities by expanding feature dimensions. However, the performance of MalFinder heavily depends on the selected statistical and sequential features. Compared to traditional machine learning techniques, deep learning can automatically extract features, reducing the subjectivity involved in feature selection. Yuan, J et al. [2] proposed an image post-processing method that combines Convolutional Neural Networks (CNN) and Transformer structures to restore high-quality images from low-dose computed tomography (LDCT) scans. To achieve this, a hybrid CNN–Transformer encoder–decoder network (HCformer) is designed specifically for LDCT image denoising. The method introduces a Neighborhood Feature Enhancement (NEF) module, which integrates local information into the Transformer, improving the representation of neighboring pixel information during the denoising process. Additionally, a shifted window technique is employed to reduce the computational complexity of the model and address issues related to the Multi-Head Self-Attention (MSA) process in traditional fixed windows. Moreover, the W/SW-MSA (Window/Shifted Window Multi-Head Self-Attention) mechanism is alternately used in two layers of the Transformer, facilitating information exchange between different layers. Chapaneri, R et al. [3] proposed a multi-stage classification method to accurately classify network traffic and identify unknown attacks, addressing the significant increase in malicious network traffic. This method employs unsupervised Gaussian mixture modeling to learn the statistical features of each traffic class and uses a quartile-based adaptive threshold technique to identify outliers. Zihao Wang et al. [4] conducted an in-depth analysis of traffic features and proposed a novel feature concept specifically designed for analyzing encrypted malicious traffic. They designed a two-layer detection framework that combines deep learning and traditional machine learning. Experimental results showed that this framework outperforms classical algorithms such as ResNet and Random Forest in terms of performance. Furthermore, multiple public datasets were integrated to ensure sufficient training for the deep learning model. Bingfeng Xu et al. [5] proposed a novel framework and system called ME-Box (Machine Learning and Evidence Verification) for reliably detecting malicious encrypted traffic. ME-Box operates by deploying middleboxes within the network and installing agents on the sending hosts. First, ME-Box evaluates the trustworthiness of encrypted flows using machine learning methods. For suspicious

traffic, it provides evidence and requests session keys for verification. If the evidence is validated, the middlebox decrypts the suspicious traffic and performs regular Deep Packet Inspection (DPI). Experimental results from the prototype system demonstrate that ME-Box protects user privacy without modifying current encryption protocols and is practically deployable. Therefore, this paper proposes an encrypted malicious traffic detection method based on domain adaptation and adversarial hybrid neural networks. As the scale of data increases, encrypted malicious traffic detection is viewed as a multimodal task involving various types of features. Compared to single-modal deep learning methods that rely on a single information source, multimodal approaches generally demonstrate better performance.

This paper addresses the challenge of detecting malicious encrypted traffic in 5G network environments by proposing an innovative detection method. Unlike traditional traffic detection techniques, the proposed method analyzes encrypted traffic without the need for decryption, primarily utilizing an adversarial hybrid neural network that combines generative adversarial networks with multiple deep convolutional networks (ResNet, ResNeXt, and DenseNet) to construct the detection model. By leveraging deep feature extraction and adaptive domain transfer techniques, a reconstructive domain adaptation module is designed to reduce the data distribution discrepancy between the source domain and the target domain, enhancing the cross-domain adaptability of the model. Furthermore, the generator module utilizes GAN to generate realistic malicious encrypted traffic samples, while the discriminator achieves precise classification through high-dimensional feature extraction. Through this design, the proposed method can achieve deep feature extraction and detection of encrypted traffic without decryption while ensuring model robustness and stability in different network environments. This approach provides a novel solution for encrypted traffic detection, demonstrating significant advantages in both accuracy and efficiency.

The main contributions of this paper are as follows:

- (1) This paper proposes a malicious encrypted traffic detection method based on reconstructive domain adaptation and adversarial hybrid neural networks, which can detect malicious encrypted traffic without decryption, effectively addressing the limitations of traditional methods in encrypted traffic detection.
- (2) By integrating generative adversarial networks with ResNet, ResNeXt, and DenseNet, an adversarial hybrid neural network is designed to achieve deep extraction of diverse features in encrypted traffic, thereby enhancing detection accuracy.
- (3) A reconstructive domain adaptation module is introduced to reduce the data distribution discrepancy between the source and target domains, enhancing the model's cross-domain adaptability and generalization, enabling stable detection performance in different network environments.

2. Preliminaries

2.1. Transfer Learning

Transfer learning [6] is a machine learning technique that leverages the similarities between data, tasks, or models to apply the knowledge learned in the source domain to the target domain. Unlike traditional machine learning approaches, which aim to gather information from each task independently, transfer learning focuses on transferring knowledge from previous tasks when the target task lacks high-quality training data, helping to address the issue of missing labels in the target domain. The key advantage of transfer learning lies in its ability to utilize existing knowledge and resources, reducing the need for large amounts of labeled data, speeding up the learning process, and

achieving better performance on new tasks. Transfer learning can be classified based on the relationship between the source and target tasks, including the following types:

- (1) Inductive Transfer Learning: The source and target tasks are different, though the source and target domains can be the same or different.
- (2) Transductive Transfer Learning: The source and target domains are different, but the tasks are the same.
- (3) Cross-Domain Transfer Learning: Both the domains and tasks are different between the source and target.

In the field of encrypted traffic detection, reference [1] applies the concept of transfer learning by proposing the EMTD-SSC model, which uses dual residual convolutional autoencoders for robust feature extraction. By minimizing the multilayer multi-kernel maximum mean discrepancy (MLMK-MMD) between network layers, the model improves transmission efficiency and enhances effective domain adaptation. Through unsupervised training and fine-tuning on target domain data, the model significantly enhances anomaly detection. In experiments, the EMTD-SSC model achieved an impressive accuracy of 94.8% in binary classification tasks.

It is evident that transfer learning, through the use of pre-trained models, can reduce training time, leading to faster convergence of the model. Additionally, adversarial generative networks can further reduce the reliance on large-scale labeled datasets.

2.2. Generative Adversarial Networks

Generative adversarial networks are a deep learning approach that achieves data generation and feature extraction through adversarial training between the generator and discriminator. The main feature of GAN is its dual-network structure, consisting of a generator and a discriminator. The generator aims to produce synthetic samples that closely approximate the real distribution, while the discriminator is responsible for classifying the samples to determine whether they are real or fake. The generator and discriminator engage in a continuous adversarial process, optimizing each other and thereby improving the quality of the generated data. The advantage of GAN lies in its ability to generate high-quality samples in situations where data are limited or lack labels, effectively enhancing the diversity of the dataset. This is particularly suitable for feature learning and generation tasks involving complex data such as malicious encrypted traffic.

In the field of encrypted traffic detection, GANs are widely used to generate malicious traffic samples, thereby improving the performance of detection models in feature extraction and classification tasks. The literature [7] has proposed a method that combines GAN with LSTM to generate high-quality data samples and detect the data through feature extraction and classification modules. Experimental results demonstrate that this method exhibits excellent detection performance in complex network environments. The generator of GAN uses random noise to generate synthetic data samples and further enhances the realism of the generated data through a multilayer network structure. Moreover, during the training process, GANs optimize the adversarial relationship between the generator and discriminator using cross-entropy loss to improve the authenticity of the generated samples and classification performance. For malicious encrypted traffic detection, GANs generate realistic samples without decryption to compensate for data scarcity, thereby enhancing the robustness of detection models.

3. System Architecture

The proposed system architecture is specifically designed for encrypted malicious traffic detection, integrating a preprocessing module, a feature extraction module, a generative adversarial network, and a cross-domain reconstruction-based domain adaptation module

to improve detection accuracy and generalization capability. The system architecture is primarily divided into the following four parts in Figure 1.

- (1) The preprocessing module processes the raw traffic data. First, redundant information is removed from the traffic data to ensure that the model focuses on the core features of malicious traffic. Through normalization and truncation, the traffic data are converted to a uniform length and further transformed into two-dimensional grayscale images, leveraging the advantages of deep learning models in image feature extraction. Meanwhile, considering the dynamic characteristics of 5G traffic, normalization enhances data consistency and the generalization capability of the model. In feature analysis, this paper primarily selects non-encrypted features that are independent of the encrypted content to capture traffic behavior patterns. By generating diverse training samples through appropriate augmentation, the model is ensured to adapt to variations in different network environments. Finally, after format conversion, the preprocessed image data are fed into the generative adversarial network to detect malicious traffic effectively.
- (2) Feature Processing. The generative adversarial network [8] is pre-trained to generate input data with enhanced features, increasing the diversity and coverage of the dataset. The pre-trained model is then fine-tuned on the encrypted traffic dataset to adapt to specific features. The generator utilizes ResNet [9], DenseNet [10], and ResNeXt [11] modules to generate realistic malicious traffic samples, thereby improving the ability to recognize malicious encrypted traffic. The discriminator also employs hybrid neural network modules to extract high-dimensional features, performing binary classification through fully connected layers and the Sigmoid activation function. Both the generator and discriminator use cross-entropy as the loss function. The generator improves the realism of the generated samples by deceiving the discriminator, while the discriminator enhances detection performance by continuously distinguishing between real and fake samples.
- (3) Domain Adaptation Processing [12]. The cross-domain reconstruction-based domain adaptation module reduces distribution discrepancies between the source and target domains through the collaborative work of feature extraction, reconstruction, adversarial networks, and classifiers, enhancing the accuracy and robustness of cross-domain detection. First, the feature extraction stage uses hybrid neural networks to extract high-dimensional features from both the source and target domains. Then, the reconstruction network learns the mapping between the source and target domain features, reconstructing the source domain features to align more closely with the target domain features, thereby minimizing feature distribution differences. Subsequently, the adversarial network further optimizes feature consistency through adversarial training, where the generator produces more deceptive target domain features while the discriminator improves its ability to distinguish between source and target domain features. Finally, the classifier categorizes the reconstructed features, optimizing the model's classification performance.
- (4) Analysis and Evaluation. During the in-depth implementation of the malicious encrypted traffic detection task, after rigorous training and testing, the performance of the adversarial hybrid neural network was evaluated using a confusion matrix.

The system architecture is shown in the following figure.

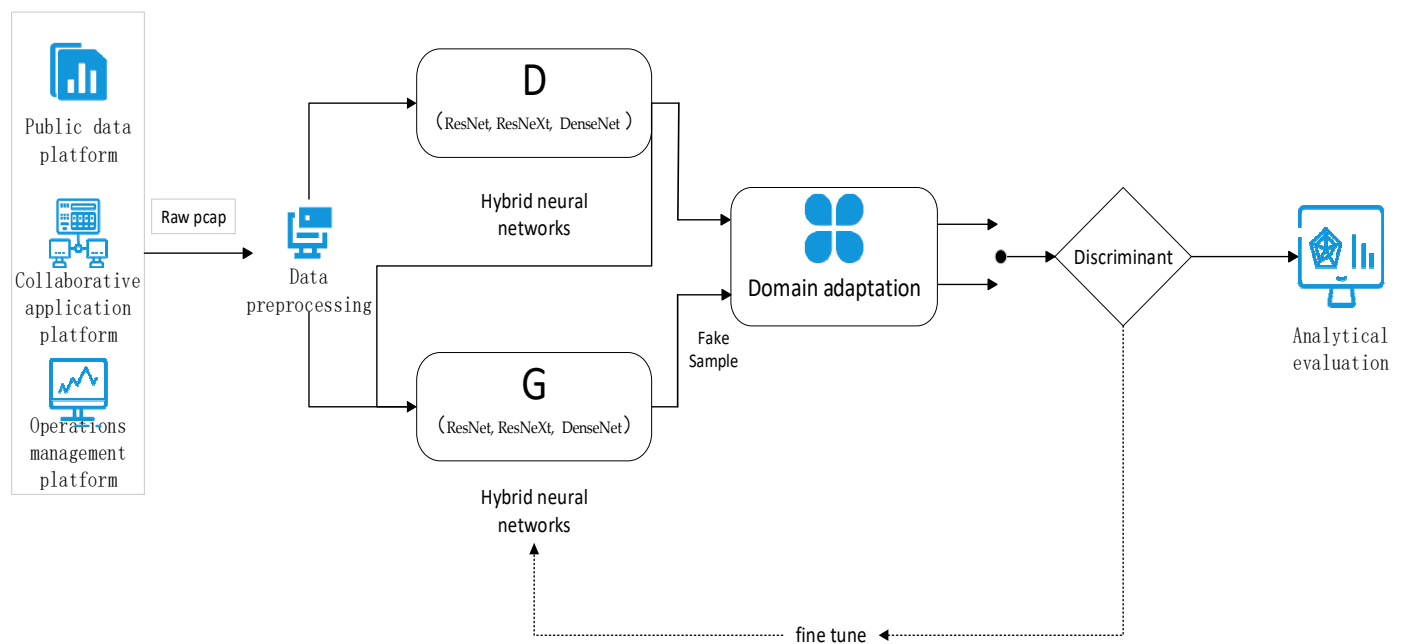


Figure 1. System architecture.

3.1. Stream-Based Preprocessing

Step 1: Removal of Redundant Data. Redundant data refers to parts of the original traffic that do not directly contribute to malicious traffic detection or might lead to overfitting, including network layer and transport layer headers (such as IP addresses, port numbers, TTL, checksums, etc.), duplicate data (multiple records of the same traffic), empty flows, and payloadless packets, as well as specific identifiers from certain higher-layer protocols (e.g., timestamps, device identifiers). While these pieces of information are essential for communication, they introduce noise or bias during model training, adversely affecting generalization performance. Therefore, the raw PCAP files are processed to remove these redundant components, retaining only the payload of the traffic, which serves as the starting point for input data. The payload contains the core features of the communication content, which are crucial for traffic feature extraction. By removing redundant data, the generator and discriminator can focus more effectively on extracting the features of malicious traffic. Additionally, all IP information is removed during preprocessing to avoid overfitting, ensuring that identifying information is not fed into the model, thereby improving its generalizability and applicability.

Step 2: Flow Deduplication and Cleaning. Flow deduplication is performed to prevent the introduction of unnecessary biases caused by duplicate traffic samples during training. By comparing unique characteristics of the traffic (such as the five-tuple: source IP, destination IP, source port, destination port, protocol type), duplicate flows are removed, retaining only one copy. Additionally, empty flows (flows without a valid payload) are removed to ensure that the model only processes meaningful traffic data. After anonymization, only the traffic features are retained to ensure that the data contain no direct identifiers of devices or users.

Step 3: Normalization and Truncation. To meet the input requirements of the hybrid neural network, all traffic sessions are normalized to a fixed length of 50,176 bytes. This is a typical input size used by classical convolutional networks such as ResNet and DenseNet, allowing the preservation of significant traffic features while being convertible to a 224×224 -pixel two-dimensional grayscale image for subsequent processing. If a traffic session exceeds 50,176 bytes, it is truncated; if it is shorter, it is padded with zeros to

reach 50,176 bytes. This ensures uniform input dimensions, avoiding issues arising from differences in truncation or padding, thereby enabling the model to focus on the features of encrypted traffic rather than variations in data length.

Converting the one-dimensional raw traffic data into two-dimensional image data is motivated by the rich research foundation and mature architectures in deep learning for image processing and feature extraction. The spatial arrangement of pixels forms an image pattern, which helps deep models recognize complex traffic feature patterns. Furthermore, image-based processing directly leverages the feature extraction capabilities of these models, particularly in capturing local features and spatial patterns, which are suitable for identifying complex feature patterns. This approach can even capture potential relationships and features between packets in encrypted traffic. Regarding data augmentation, a cautious approach is adopted. Common image augmentation techniques, such as rotation and scaling, are not entirely applicable to traffic data, as the direction and spatial relationship of traffic data do not have practical meanings for these transformations. Therefore, data augmentation is limited to slight perturbations and minor modifications of the traffic images (such as random cropping or adding small amounts of noise) to ensure that the logical structure and feature patterns of the data are not compromised.

Step 4: Feature Analysis and Selection. When dealing with encrypted traffic, we focused on analyzing non-encrypted features to achieve more accurate classification. In addition to protocol header information (such as source/destination IP, ports, and transport protocol), features unrelated to encrypted content, such as packet size, packet inter-arrival time, and traffic direction, are also considered. These features can reflect traffic behavior without involving encrypted content. Furthermore, relying solely on plaintext features from protocol headers may not be sufficient for accurate classification. Thus, we also focus on traffic pattern features, including packet flow sequences, size distributions, and statistical patterns. By preprocessing and denoising, the model is ensured to concentrate on traffic behavior characteristics rather than the encrypted content itself, enhancing its classification capability.

Step 5: Two-Dimensional Grayscale Image Conversion. The truncated traffic data are converted into a 224×224 two-dimensional grayscale image. Each byte value (0–255) corresponds to the grayscale value of a pixel, converting the traffic data from byte representation to image representation. This image-based processing fully utilizes the hybrid neural network's ability to extract features from images, thereby uncovering latent patterns and characteristics in encrypted traffic data.

Step 6: Enhancing Data Diversity. To facilitate the generative tasks of the adversarial network, an additional data augmentation step is introduced. Diverse training samples are generated through image processing techniques such as random rotation, scaling, and translation. Data augmentation helps the generator produce more varied samples, thereby improving the generalization capability of the model and making it more stable across different network environments and time periods.

Step 7: Data Format Conversion. After grayscale image processing, the traffic data are converted into a format suitable for model input. Similar to the processing of the MNIST dataset, these grayscale images are converted into batch training data. Using the PyTorch tensor format, the data are prepared for input into the generative adversarial network for training.

Step 8: Sample Generation and Classification. After the aforementioned preprocessing, the input to the generator consists of latent space vectors generated from random noise. These vectors are transformed by the generator into realistic 224×224 grayscale images representing malicious traffic samples. The discriminator's input includes both real encrypted malicious traffic samples (preprocessed) and generated traffic samples.

3.2. Feature Processing Model

In practical applications, data scarcity and the classification of data feature categories are common challenges in detecting malicious encrypted traffic. This paper proposes a model that combines a generative adversarial network with a hybrid neural network. The adversarial hybrid neural network is pre-trained on the original dataset to generate input data with enhanced features. The pre-trained model and its parameters are then transferred to the encrypted traffic dataset for further fine-tuning and retraining to adapt to the specific characteristics of encrypted traffic. The pre-training phase with the adversarial hybrid neural network not only generates data for the scarce portions, increasing the diversity and coverage of the dataset, but also leverages the advantages of the ResNe, ResNeXt, and DenseNet architectures. ResNet effectively alleviates the vanishing gradient problem in deep networks through residual connections, enabling deeper feature learning. DenseNet ensures efficient information and feature transmission and reuse through dense connections between layers, improving the learning efficiency of the model. ResNeXt enhances the model's representational capacity by introducing diverse feature extraction paths through its multi-path parallel structure. Compared to traditional models, this approach has the advantage of accelerating the training process by using pre-trained models and improving the detection performance of encrypted traffic by leveraging the feature extraction capabilities of multiple networks.

3.2.1. Generators

In malicious encrypted traffic detection, the primary task of the generator is to generate realistic malicious encrypted traffic samples from random noise vectors. By simulating and producing highly similar traffic data, the generator can provide more training data for the detection model, enhancing its ability to recognize malicious encrypted traffic.

(1) Input Layer

The generator first accepts a low-dimensional random noise vector z as input, which is randomly sampled from a standard normal distribution.

$$z \sim N(0, 1)$$

The noise vector is expanded through a fully connected layer, generating an initial high-dimensional representation that serves as the foundation for the subsequent malicious encrypted traffic samples. This process is expressed by the following equation:

$$f = \sigma(W_1 z + b_1)$$

In this formula, W_1 and b_1 represent the weights and biases of the fully connected layer, and is the ReLU activation function, which ensures that the generator can construct a non-linear initial representation from the random input.

(2) Hybrid Feature Extraction Module

In the hybrid feature extraction module, the generator replaces traditional convolutional layers with the feature extraction modules from ResNet, DenseNet, and ResNeXt to more comprehensively simulate the diversity and complexity of malicious encrypted traffic.

ResNet Residual Module: With residual connections, the generator maintains training stability when dealing with complex, high-dimensional encrypted traffic data, alleviating the vanishing gradient problem and deeply extracting malicious features within encrypted traffic.

$$f_{res} = F(x, \{W_i\}) + x$$

$F(x, \{W_i\})$ represents the non-linear transformation through convolutional layers and batch normalization; x is the input feature. The residual connection mechanism enables the direct passing of input features to the output, ensuring that the model learns deep information more stably.

DenseNet Module: The dense connections in DenseNet allow every layer in the network to share and reuse the features from all previous layers, ensuring that the generator can effectively capture both global and local features of malicious encrypted traffic, enabling efficient information transfer.

$$f_{dense} = [f_1, f_2, \dots, f_{i-1}]$$

In this formula, $[\cdot]$ Denotes the feature concatenation operation, which combines the output of each layer.

ResNeXt Multi-Path Module: The multi-path parallel convolution operations in ResNeXt allow the generator to extract different types of malicious traffic features from various perspectives. This structure greatly enriches the generator's feature representation capabilities and enhances the diversity of the generated samples.

$$f_{resnext} = \sum_{i=1}^k F_i(x, \{W_i\})$$

In this formula, k represents the number of parallel paths; F_i denotes the convolution operation of the i -th path.

(3) Feature Fusion

To further enhance the effectiveness of feature representation, a similarity detection module is introduced. The similarity detection module takes into account the covariance between features, enabling more accurate measurement of distances between high-dimensional feature vectors. During training, it is necessary to compute the covariance matrix S of the fused feature vectors [13]. Typically, the covariance matrix is calculated from the feature vectors of the entire training set as follows:

$$s = \frac{1}{N-1} \sum_{i=1}^N (f_i - \mu)(f_i - \mu)^T$$

$$\mu = \frac{1}{N} \sum_{i=1}^N f_i$$

In this formula, N is the number of training samples, f_i is the high-order feature vector of the i -th sample, and μ is the mean of the feature vectors.

Through similarity detection, the generator can effectively integrate the feature representations from different modules, generating malicious encrypted traffic samples with a more accurate and similar statistical distribution.

(4) Upsampling Layer

The upsampling layer is responsible for enlarging the fused feature maps and restoring them to the spatial resolution consistent with the target malicious traffic samples. This is achieved through the deconvolution operation:

$$h_{up} = ConvTranspose(h_{sim})$$

In the generative adversarial network, the upsampling layers not only increase the resolution of the generator's output images but also enhance the detail representation of the feature maps, making the generated samples more realistic. Additionally, the upsampling layers support the integration of multi-scale features during the gradual enlargement of the images, which helps the model capture both global patterns and local details in traffic data, particularly the latent feature patterns in encrypted traffic. By generating traffic images of different resolutions, the generator can produce more diverse samples, thereby improving the model's generalization ability and enabling better performance in different network environments and traffic patterns [14].

(5) Output Layer

In the output layer, the generator uses the Tanh activation function to constrain the generated samples within the range of $(-1, 1)$, ultimately producing realistic malicious encrypted traffic samples. The formula is as follows:

$$G(z) = h_{out} = \tanh(h_{up})$$

The final generated samples $G(z) = h_{out}$ will serve as input to the discriminator, which will determine the authenticity of the generated samples. In the task of malicious encrypted traffic detection, the generator, optimized by the similarity detection module, is capable of producing more realistic traffic samples, thereby enhancing the detection model's ability to identify malicious traffic.

3.2.2. Discriminators

The discriminator is responsible for distinguishing between the fake samples generated by the generator and the real malicious encrypted traffic samples. In this model, the discriminator also uses hybrid neural network modules to replace traditional convolutional layers, enhancing its ability to discriminate high-dimensional and complex data.

(1) Input Layer

The discriminator receives either real malicious encrypted traffic data samples or the fake samples generated by the generator as input. These input samples are then passed to the hybrid neural network modules for feature extraction and analysis, with the network architecture being the same as that of the generator.

(2) Fully Connected Layer

The high-dimensional features extracted by the hybrid feature extraction module are passed to the fully connected layer [15], which maps these high-dimensional feature vectors into lower-dimensional representations for the final classification task of malicious encrypted traffic. When processing malicious encrypted traffic, the feature dimensions of the data are often highly complex, containing numerous hidden patterns and behavioral characteristics. The deep features extracted by the hybrid neural network through the ResNet, DenseNet, and ResNeXt modules encapsulate multi-dimensional and multi-level feature representations of malicious traffic.

The role of the fully connected layer is to further compress and aggregate these high-dimensional features, thereby reducing redundant information while retaining the most distinctive feature vectors. This process is crucial because the data characteristics of malicious encrypted traffic are often highly complex and redundant, and uncompressed high-dimensional features may lead to model overfitting or low processing efficiency. By mapping the high-dimensional feature vectors into lower-dimensional representations, the discriminator can more efficiently classify malicious encrypted traffic, improving the accuracy of determining whether the traffic is a malicious sample. Moreover, during the

compression of these features, the fully connected layer effectively extracts the most critical feature points for detecting malicious traffic, laying the foundation for the subsequent binary classification operation.

(3) Output Layer

In the output layer, the discriminator uses the Sigmoid activation function for binary classification, producing a value $D(x)$ to determine whether the input sample is real data. The output range of the Sigmoid [16] activation function is between (0, 1), where values closer to 1 indicate that the input sample is more likely to be real, while values closer to 0 suggest that the input is a fake sample generated by the generator.

$$D(x) = \sigma(W \cdot f + b)$$

In this formula, W is the weight matrix of the fully connected layer, f is the feature vector obtained from the feature extraction module, b is the bias, and σ is the Sigmoid activation function.

3.2.3. Loss Function

In the training of the model, the loss functions of the generator and the discriminator are key components, reflecting the adversarial relationship between them.

(1) Generator Loss

The goal of the generator is to produce fake samples that are realistic enough to deceive the discriminator into classifying them as real. The generator's loss function is defined using cross-entropy loss as follows [17]:

$$L_G = -\log(D(G(z)))$$

In this formula, $G(z)$ Represents the samples generated by the generator, and $D(G(z))$ is the probability assigned by the discriminator to classify the generated samples. The generator minimizes this loss to deceive the discriminator as effectively as possible.

(2) Discriminator Loss

The task of the discriminator is to distinguish between real data and generated data. The discriminator's loss function also uses cross-entropy loss to maximize its ability to differentiate between real and fake data. It is defined as follows:

$$L_D = -[\log(D(x)) + \log(1 - D(G(z)))]$$

$D(x)$ Represents the probability assigned by the discriminator to classify real data. The discriminator minimizes this loss to learn how to better distinguish between real samples and the fake samples generated by the generator.

Through this adversarial process, the generator continuously improves the quality of the generated samples while the discriminator enhances its ability to distinguish them, ultimately making the generated malicious encrypted traffic samples more realistic. As shown in Figure 2.

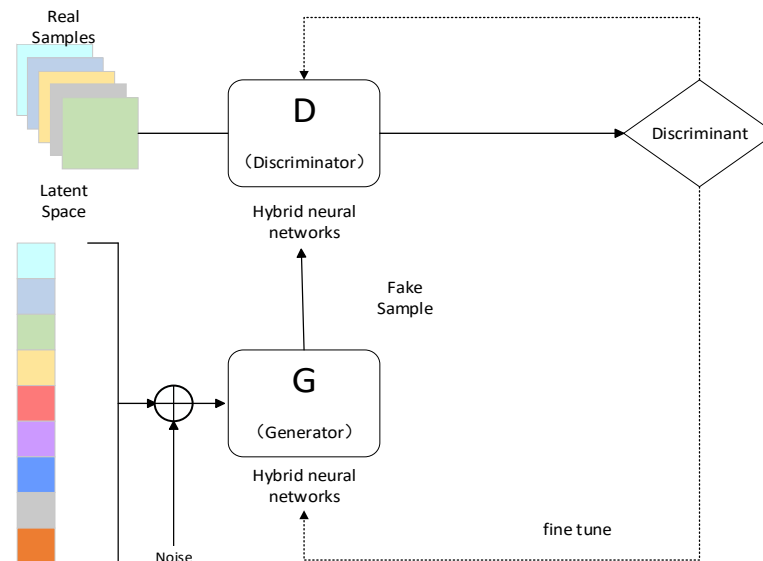


Figure 2. Adversarially generative hybrid neural networks.

3.3. Domain Adaptation Module Based on Cross-Domain Reconstruction

However, we have found that in practical applications, encrypted malicious traffic detection faces the challenge of data distribution discrepancies. To improve the model's performance across different domains, the reconstruction-based domain adaptation method effectively reduces inter-domain differences by reconstructing features between the source and target domains. This paper proposes an improved reconstruction-based domain adaptation module specifically designed for encrypted malicious traffic detection.

3.3.1. Cross-Domain Feature Extraction

In the reconstruction-based domain adaptation adversarial generative hybrid neural network, the source domain features [18] are extracted from real data, which are the actual source domain data received by the discriminator; the target domain features are generated by the generator. Feature extraction relies on the ResNet, DenseNet, and ResNeXt modules within the hybrid neural network [19].

Through the collaborative work of these modules, both the generator and the discriminator are able to extract high-dimensional, deep features from both the source and target domains. These features provide effective input for subsequent cross-domain feature reconstruction and adversarial training, helping to reduce the disparity between the source and target domains and enhance the model's generalization capability.

- (1) Output of the Generator: The generator processes random noise z to generate target domain features, expressed as:

$$X_{\text{target}} = G(z)$$

In this formula, G represents the generator, z is the random noise, and X_{target} is the generated target domain feature.

- (2) Output of the Discriminator: The discriminator processes the real data from the source domain and extracts the source domain features, expressed as:

$$X_{\text{source}} = D(X_{\text{source}})$$

In this formula, D represents the discriminator, and X_{source} is the real source domain feature.

3.3.2. Cross-Domain Reconstruction Module

The primary task of the cross-domain reconstruction module is to reconstruct target domain features from source domain features, thereby reducing the distribution discrepancies between the source and target domains. This module learns the mapping relationship between source and target domain features to minimize inter-domain differences, enabling the model to better adapt to cross-domain tasks. As shown in Figure 3.

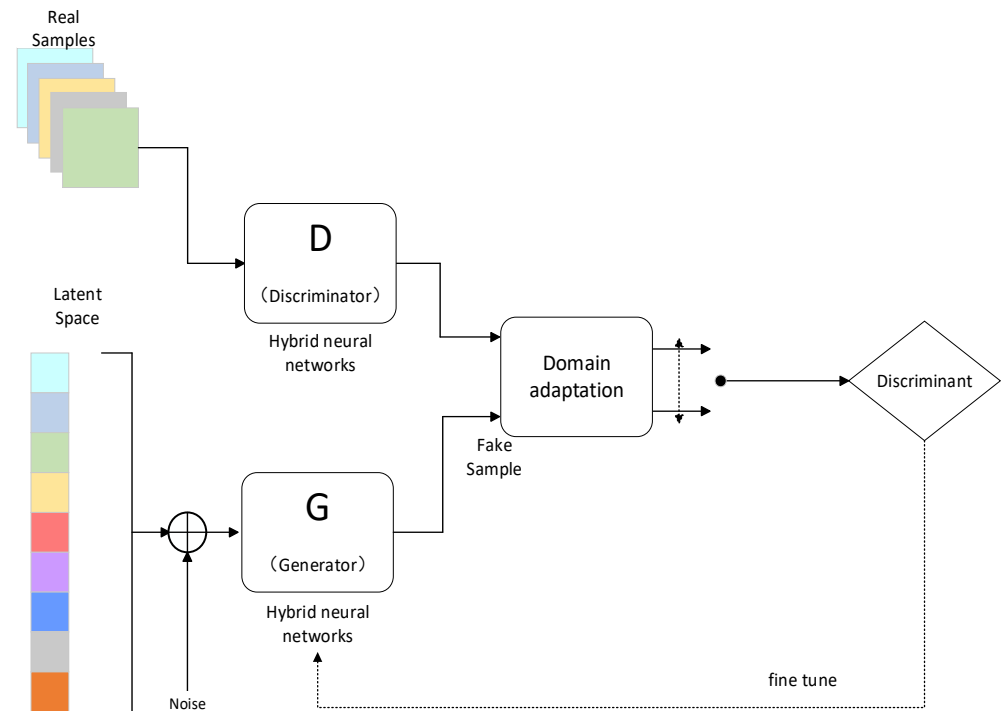


Figure 3. Adversarially generative hybrid neural networks.

- (1) **Feature Mapping:** Source domain features are received as input, containing complex patterns and structural information from the source domain data, which serve as the foundational input for cross-domain reconstruction.

The source domain features are mapped to the target domain feature space through multiple layers of non-linear transformations. In this process, the mapping relationship between the source domain features X_{source} and the target domain features X_{target} can be modeled using a deep neural network. Let the source domain features $X_{\text{source}} \in \mathbb{R}^{n \times d}$ and the target domain features $X_{\text{target}} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the dimension of the features. The feature mapping from the source domain to the target domain is performed through a multilayer perceptron network, producing a non-linear mapping.

We define the reconstruction network as a mapping function $f_{\theta}(\cdot)$, where f_{θ} is the mapping function composed of multiple layers of neural networks and non-linear activation functions. The reconstructed target domain features X'_{target} are obtained by applying the mapping function f_{θ} to the source domain features X_{source} , represented as:

$$X'_{\text{target}} = f_{\theta}(X_{\text{source}})$$

In this formula, $f_{\theta}(\cdot)$ is the mapping function defined by network parameters θ , designed to learn the mapping from the source domain to the target domain through training.

- (2) **Reconstruction Process:** In the reconstruction process, each layer of the network learns features using fully connected layers and convolutional layers. Let $X_{source}^{(i)} \in \mathbb{R}^d$ represent the source domain feature for the i -th sample. Each layer's output $X_{target}^{(l)}$ contains the information from the previous layer's input and performs feature extraction through a hierarchical structure. Non-linear activation functions are used to perform non-linear mappings to improve the model's expressiveness. The output for each layer can be represented as:

$$X_{target}^{(l)} = \sigma(W^{(l)}X_{target}^{(l-1)} + b^{(l)})$$

In this formula, $W^{(l)}$ is the weight matrix of the l -th layer, $b^{(l)}$ is the bias term, and $\sigma(\cdot)$ is the nonlinear activation function.

- (3) **Reconstruction Loss:** To optimize the quality of cross-domain reconstruction, mean squared error is used as the reconstruction loss function, calculating the difference between source domain features and target domain features. The reconstruction network is optimized by minimizing the difference between the reconstructed source domain features and the target domain features. The loss function is computed as follows:

$$L_{rec} = \frac{1}{N} \sum_{i=1}^N ||X_{source}^{(i)} - X_{target}^{(i)}||^2$$

In this formula, N is the number of samples, $X_{source}^{(i)}$ is the source domain feature for the i -th sample, and $X_{target}^{(i)}$ is the corresponding target domain feature. This measures the mean squared error between the source domain features and target domain features.

- (4) **Output Reconstructed Features:** The features generated by the reconstruction network are reconstructions of the source domain features. These reconstructed features will serve as input to subsequent modules, continuing to participate in the adversarial training and feature classification tasks. Through this reconstruction mechanism, the model can establish more consistent feature representations between the source and target domains, providing more reliable input data for loss calculation and classification training [20].

4. Simulation and Performance Analysis

4.1. Experimental Environment

This study establishes the experimental environment using Python 3.8, PyCharm 2023.1.1 as the development environment, and PyTorch 1.10.0 as the deep learning framework. The hardware setup includes a 64-bit Windows 10 operating system, an i7-14900K processor, and an RTX 4070Ti graphics card with 32 GB of GPU memory.

4.2. Dataset

In the experiments, the datasets used in this approach include USTC-TFC2016 and CTU-13. The USTC-TFC2016 dataset contains a large number of normal and malicious encrypted traffic samples, divided into two main parts. The first part includes 10 types of malware traffic samples collected by CTU researchers from real network environments between 2011 and 2015, all sourced from public websites. These malware samples cover various types of malicious activities, ensuring the diversity and representativeness of the dataset. The second part consists of 10 types of normal traffic data collected using IXIA BPS, a professional network traffic simulation device. As shown in Tables 1 and 2.

Table 1. Part 1 Dataset.

Name	CTU	Binary MD5	Procedure
Cridex	108-1	25b8631afeea279ac00b2da70fffe18a	original
Geodo	119-2	306573e52008779a0801a25fafb18101	part
Htbot	110-1	e515267ba19417974a63b51e4f7dd9e9	original
Miuref	127-1	a41d395286deb113e17bd3f4b69ec182	original
Neris	42,43	bf08e6b02e00d2bc6dd493e93e69872f	merge
Nsis-ay	53	eaf85db9898d3c9101fd5fcfa4ac80e4	original
Shifu	142-1	b9bc3f1b2aace824482c10ffa422f78b	part
Tinba	150-1	e9718e38e35ca31c6bc0281cb4ecfae8	part
Virut	54	85f9a5247afbe51e64794193f1dd72eb	original
Zeus	116-2	8df6603d7cbc2fd5862b14377582d46	original

Table 2. Part 2 Dataset.

Name	Type	Name	Type
BitTorrent	P2P	Outlook	Email/WebMail
Facetime	Voice/Video	Skype	Chat/IM
FTP	Data Transfer	SMB	Data Transfer
Gmail	Email/WebMail	Weibo	Social NetWork
MySQL	Database	WorldOfWarcraft	Game

IXIA BPS is an advanced network testing tool capable of generating precise and reliable network traffic simulation data. This part of the dataset covers a variety of normal network applications and service traffic, providing a solid foundation for model training and testing.

CTU-13 is a network traffic dataset used for analyzing malware behavior, collected by the Czech Technical University. This dataset aims to provide researchers with a standardized and realistic network traffic dataset for studying and analyzing the behavioral characteristics of malware. The CTU-13 dataset consists of traffic from 13 different scenarios, each containing normal traffic, malware traffic, and background traffic.network applications and service traffic, providing a solid foundation for model training and testing.

4.3. Evaluation Metrics

In the final part of the model, a classifier module [14] is added to classify the generated features, enhancing the model's classification performance on the target domain. By processing the input features, the classifier generates a probability distribution over the categories. The output corresponding to each feature is the probability for different categories, and the category with the highest probability is the model's prediction.

This prediction is critical for evaluating key performance metrics such as accuracy, precision, recall, and F1 score. These metrics comprehensively showcase the model's performance in detecting malicious encrypted traffic, providing an important basis for further improving network security defenses.

- (1) Accuracy: Accuracy represents the proportion of correctly classified instances among all predictions and is the most direct metric to measure classification performance. The formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP (True Positive): The number of samples that are actually positive and predicted as positive.

TN (True Negative): The number of samples that are actually negative and predicted as negative.

FP (False Positive): The number of samples that are actually negative but predicted as positive.

FN (False Negative): The number of samples that are actually positive but predicted as negative.

- (2) Precision: Precision measures the proportion of true positives among the samples predicted as positive. The formula is:

$$Precision = \frac{TP}{TP + FP}$$

- (3) Recall: Also known as the true positive rate, recall refers to the proportion of actual positive cases that are correctly identified as positive. The formula is:

$$Recall = \frac{TP}{TP + FN}$$

- (4) F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced and comprehensive metric. The formula is:

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.4. Experimental Results and Analysis

This study conducts multiple experiments to evaluate the performance of the proposed model. In these experiments, several critical parameters are configured to optimize the training process and ensure the reproducibility of the results. The batch size is set to 16 to enable effective gradient updates under limited GPU memory, avoiding the gradient noise caused by excessively large batch sizes. The Adam optimizer is employed for training due to its adaptive adjustment of gradients and fast convergence, making it well-suited for deep-learning tasks. The initial learning rate is set to 0.0001 and is dynamically adjusted during training using a decay factor of 0.001. This gradual reduction in the learning rate prevents oscillations near the optimal solution and ensures the model's convergence and accuracy. Moreover, to avoid local optima and overfitting, a dynamic adjustment strategy is adopted, which updates the learning rate based on feedback from the loss function.

To further mitigate overfitting during training, L2 regularization is introduced with a regularization parameter of 0.0001. This effectively constrains excessively large weights and reduces noise interference.

In terms of experimental setup, the initial focus is on assessing the model's loss values to provide a reference for subsequent experiments. The model is then tested on a binary classification task for all traffic data to evaluate its performance in detecting encrypted malicious traffic. On this basis, domain adaptation experiments are conducted using encrypted malicious traffic datasets across different times and locations. These experiments aim to analyze the model's generalization ability and stability in cross-domain data environments. Additionally, experiments on different network architectures are carried out to further test the model's performance under various network conditions and to analyze its detection effectiveness in different scenarios. These experimental designs and parameter configurations contribute to a comprehensive evaluation of the model's performance and adaptability.

4.4.1. Experiment 1: Impact of Removing Low-Contribution Features on the Model

In encrypted traffic detection, certain features were evaluated and found to contribute minimally to the model's performance or even introduce biases related to environmental or network configurations, thus proving to be invalid features. For instance, source and

destination IP addresses are primarily used for localization and routing, and they do not reflect the content characteristics of packets. Particularly in encrypted traffic, these features lack direct relevance to the communication content and are prone to causing the model to overfit specific network environments, thereby reducing generalization capability. Similarly, source and destination ports, which mainly indicate the type of service (e.g., HTTP, HTTPS), provide insufficient distinction in encrypted traffic detection. Port information has minimal impact on content characteristics and is ineffective in distinguishing malicious behaviors. Additionally, the Time-To-Live (TTL) field, designed to control the transmission scope of packets in the network, reflects certain network transmission properties but does not directly correlate with behavioral patterns or content characteristics of the traffic. Timestamp information, used to record the sending or arrival time of packets, can indicate network behavior patterns; however, it is sensitive to variations in different network environments, potentially leading to overfitting in specific environments and offering limited contributions to malicious behavior detection. The identification field in the IP protocol, used for the fragmentation and reassembly of packets, is unrelated to data content and offers negligible assistance in identifying traffic pattern features. Furthermore, traffic type or service type fields (ToS/DSCP) are employed to define the priority and service type of network transmission, but they generally lack effective content descriptors and provide little value in detecting malicious behavior within encrypted traffic. Eliminating these invalid features enables the model to focus more effectively on key features that reflect traffic behavior, thereby enhancing its generalization capability and detection performance. As shown in Figure 4.

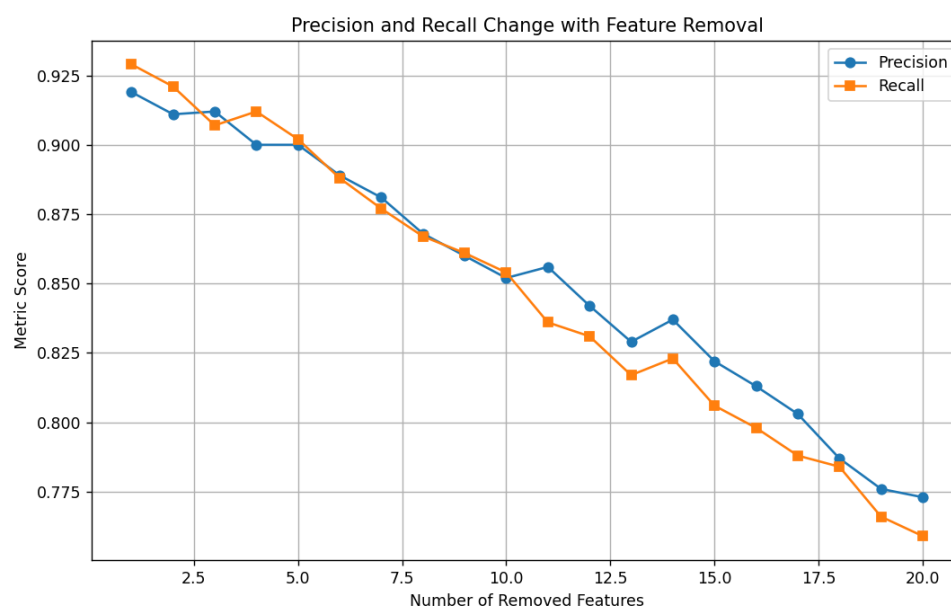


Figure 4. Remove the effect of low contribution features on precision and recall.

4.4.2. Experiment 2: Impact of Loss Values on Detection Results

In this experiment, using the USTC-TFC2016 and CTU-13 datasets, the impact of the loss values from each module on the detection results of malicious encrypted traffic is studied through experimentation.

As shown in Table 3, by reasonably adjusting the weights of the generator loss, discriminator loss, classification loss, and regularization loss, the overall training process of the model can be effectively optimized. This ensures that the various losses remain stable throughout the training process. This optimization strategy not only improves the convergence speed of the model but also effectively prevents overfitting. Specifically, the

adversarial generator loss and adversarial discriminator loss of the reconstruction-based domain adaptation adversarial hybrid neural network are 4.73/4.58 and 0.029/0.032, respectively, indicating that the generator and discriminator have achieved a good balance between generating realistic samples and distinguishing between real and fake samples. This further validates the applicability and robustness of this domain adaptation method in cross-domain scenarios.

Table 3. Module Loss Values.

Name	Loss
Counter generator losses	4.73/4.58
Counter discriminator loss	0.029/0.032

4.4.3. Experiment 3: Encrypted Malicious Traffic Detection Based on the Proposed System

In this experiment, using the encrypted malicious traffic dataset, the raw traffic data were preprocessed into grayscale images suitable for model input and then trained over several iterations. The study investigates whether the model demonstrates good performance in detection. Figure 5 shows the F1 score of the model after training.

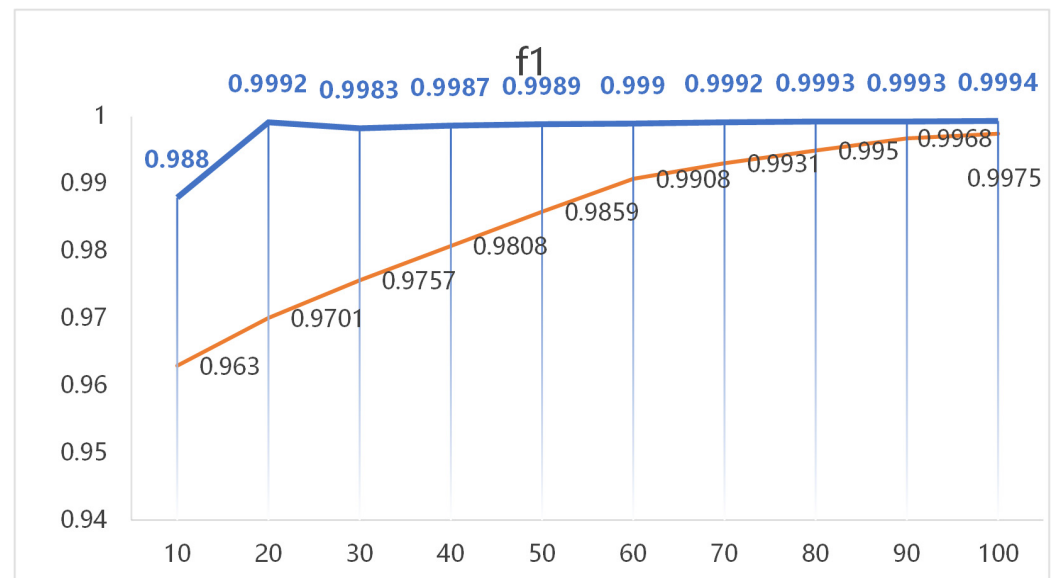


Figure 5. The F1 scores of the two datasets after training.

As seen in Figure 5, the USTC-TFC2016 and CTU-12 encrypted malicious traffic dataset was preprocessed and input into the model, and the model's performance was evaluated. To comprehensively assess the model's effectiveness in detecting malicious encrypted traffic, the F1 score was selected as the primary evaluation metric, with precision and recall also analyzed. The F1 score was used in this experiment to balance the model's precision and recall, ensuring that the model not only accurately detects malicious traffic but also minimizes the false-negative rate as much as possible. After 30 rounds of training, the model achieved an F1 score of 0.9983 and 0.9701, approaching the ideal value, indicating that the model can accurately identify most malicious traffic samples. In the early stages of the experiment, the F1 scores were 0.988 and 0.963, suggesting that the model was already capable of capturing some malicious traffic characteristics at an early stage, and its detection performance improved as training progressed. These results demonstrate the strong capability of the proposed adversarial hybrid neural network in handling complex malicious encrypted traffic.

4.4.4. Experiment 4: Impact of Different Network Environments, Time Periods, and Datasets on Detection Results

In practical applications, the detection model must not only accurately identify encrypted malicious traffic from large-scale and complex network traffic but also demonstrate stable detection performance across different time periods, network environments, and datasets. Given the massive and frequently changing nature of network traffic, the model must be capable of efficiently handling large-scale data, processing vast amounts of traffic quickly, and accurately identifying malicious traffic amidst substantial normal traffic, all while maintaining a low false-positive rate.

As shown in Figure 6, over the course of 10 training waves, the model's precision gradually improved with the increasing number of training rounds. Notably, after the 5th wave, the precision in all three different test environments stabilized above 0.98, indicating that the false-positive rate was effectively controlled, and the model was able to accurately identify malicious traffic. This demonstrates that the model can effectively distinguish between real malicious samples and normal traffic, with a low number of false positives.

As shown in the recall results in Figure 7, the recall rate gradually increased with the number of training rounds, eventually approaching 0.999. Recall represents the proportion of actual positive instances that are successfully identified by the model. A high recall rate indicates that the model is effectively detecting malicious traffic samples with a very low missed detection rate. Notably, after the 3rd wave of training, the recall rate remained at a high level, further enhancing the model's ability to detect malicious traffic in real-world scenarios.

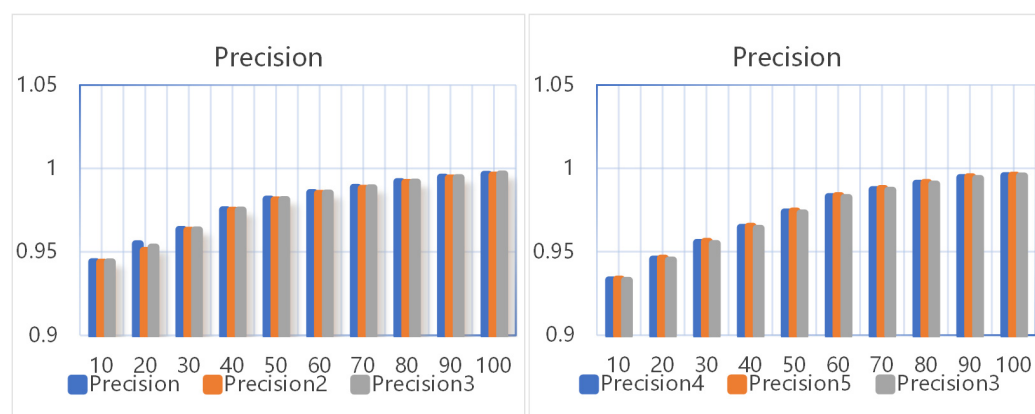


Figure 6. The precision of the two datasets in different network environments.

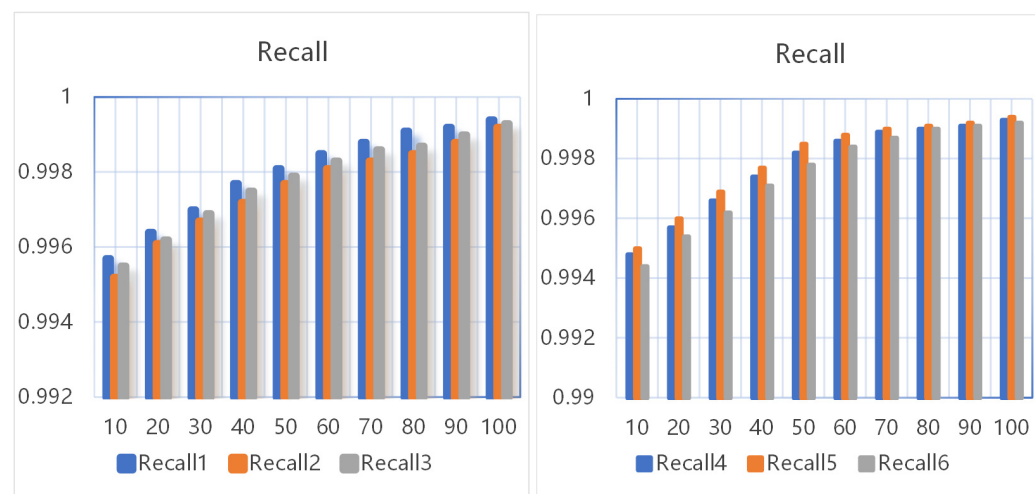


Figure 7. The recall of two datasets in different network environments.

The F1 score is the harmonic mean of precision and recall, providing a comprehensive measure of the model's detection performance. As shown in Figure 8, the F1 score gradually increases as the training progresses, eventually reaching nearly 1.0 during the 9th and 10th epochs. This indicates that the model achieves a good balance between precision and recall, effectively and accurately identifying malicious encrypted traffic in large-scale data flows.

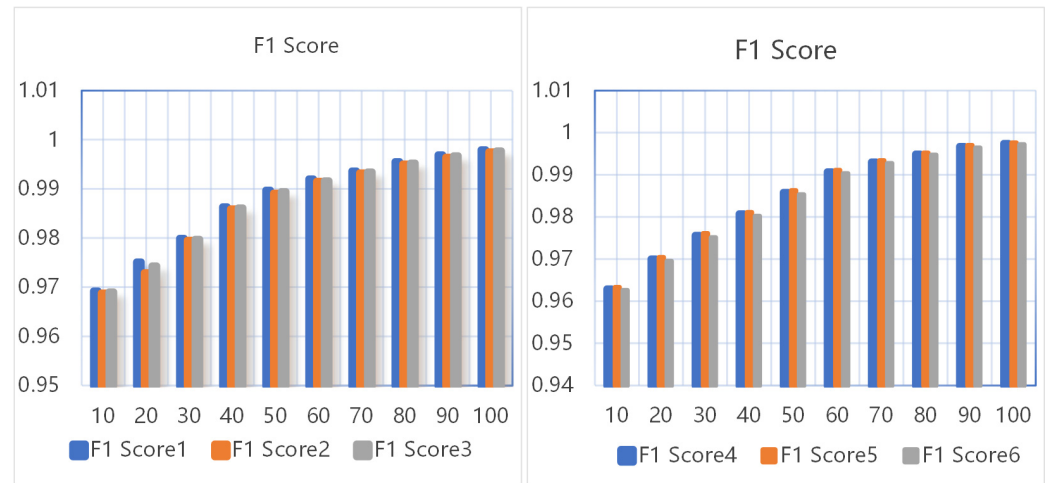


Figure 8. The F1 score of two datasets in different network environments.

4.4.5. Experiment 5: Comparative Experiment

This experiment begins with a detailed comparative analysis, contrasting the model presented in this paper with models from several published studies regarding the accuracy, recall, and F1 scores in detecting encrypted malicious traffic within 5G data. Through this comparative analysis, we provide a thorough evaluation of the model's advantages, revealing significant improvements in handling complex data and enhancing detection accuracy, especially when working with larger datasets. Furthermore, we explore the applicability of this model in comparison to traditional single models for such tasks. As shown in Table 4.

Table 4. Performance Comparison.

Method	Precision%	Recall%	F1 Score%
literature [21]	97.37	×	96.73
literature [22]	98.90	99.00	98.90
literature [23]	95.25	99.90	96.00
literature [24]	98.40	98.90	98.40
literature [25]	98.45	98.11	98.25
this scenario	99.68	99.93	99.94

The literature [20] proposes an ensemble model that combines deep learning, machine learning, and self-attention mechanisms. Additionally, novel TLS features were designed for traffic feature extraction and experimentally validated. Experimental results demonstrate that the ML-based ensemble model achieved an accuracy of 94.85%, while the ensemble model integrating RF, LSTM, Bi-LSTM, and self-attention techniques improved the precision to 96.71%.

The literature [21] proposes an improved system for detecting malicious traffic in multi-environment (M-En) networks, including IoT, SDN, and traditional IP traffic. To address the limitations of existing technologies in leveraging traffic diversity, the system introduces a Synthetic Data Augmentation Technique (S-DATE) and a Particle Swarm

Optimization-based Diverse Self-Ensemble Model (PSO-D-SEM). By integrating the InSDN, UNSWNB-15, and IoTID-20 datasets, an M-En dataset is constructed. S-DATE resolves data imbalance issues, enhancing model convergence and traffic detection capability. PSO-D-SEM significantly improves performance by optimizing the handling of complex scenarios and integrating the results of individual models. Experimental results demonstrate that this method achieves an accuracy of 0.989.

The literature [22] presents a distributed framework for detecting malicious traffic at IoT access points. The framework utilizes a transformer-based neural network to achieve distributed detection across multiple detection points. By employing a federated learning-based personalized collaboration algorithm, neural network parameters are shared to enhance detection performance. Experimental results demonstrate that this method achieves an accuracy of 99.2%.

Additionally, a machine learning-driven binary classification intrusion detection system is employed, combining eight different supervised machine learning techniques with an ensemble classification method, achieving an accuracy of 98.40% [23].

The literature [24] proposes a lightweight IoT malicious traffic detection model named LightGuard, constructed based on lightweight residual block modules. The LRB module draws inspiration from the design principles of ShuffleNetV2 and adopts a unique residual structure, effectively reducing model parameters, computational cost, and inference time while maintaining high detection performance. Additionally, the LRB module replaces traditional convolutions with lightweight ghost convolution modules, generating feature maps at lower cost without compromising detection accuracy. Experimental results show that LightGuard achieves an accuracy exceeding 99.6% across four widely used datasets.

To comprehensively demonstrate the advantages of the proposed method, we conducted comparison experiments, particularly focusing on performance comparisons across different datasets and network environments. Two widely used malicious traffic datasets were selected for the experiment, and detailed comparisons were made with existing methods, especially those that integrate deep learning, machine learning, and self-attention mechanisms [20], methods based on Synthetic Data Augmentation Technique (S-DATE) and Particle Swarm Optimization-based Diverse Self-Ensemble Model (PSO-D-SEM) [21], as well as the lightweight IoT malicious traffic detection model LightGuard [24]. The experimental results show that the proposed method achieves significant performance improvements on all datasets, outperforming existing methods in terms of accuracy, F1 score, and recall. Particularly in multi-environment (M-En) network scenarios, our method optimizes the handling of complex traffic patterns and integrates model results, enhancing overall model performance and demonstrating robustness and adaptability in diverse scenarios. Ultimately, the proposed method shows significant advantages in handling malicious traffic in multi-environment networks.

Although the methods proposed offer significant advantages in improving the model's adaptability, generalization, and ability to handle unknown malicious traffic, there are still several potential drawbacks and challenges. While optimizing computational complexity has effectively improved the model's resource utilization, simplifying the model architecture (such as pruning and lightweight techniques) may lead to information loss, which could, in turn, affect overall performance. This is particularly evident in cases requiring high-dimensional features and complex computations, where excessive simplification may undermine the model's expressive power, impacting its effectiveness in certain specialized network environments. In high-traffic or high-concurrency network scenarios, further research and exploration are necessary to enhance the model's real-time responsiveness and processing capabilities.

5. Conclusions

In a 5G network environment, the characteristics and behavior patterns of malicious traffic continuously evolve, imposing higher requirements on the adaptability and generalization capability of models. Therefore, enhancing adaptability to potential future types of malicious traffic is of critical importance. Specifically, models must not only identify known attack types but also possess the capability to detect and classify entirely unknown malicious behaviors. To this end, future research will focus on improving the model's generalization in unknown environments to ensure high detection accuracy when faced with novel attack techniques. At the same time, simplifying model structures and optimizing computational complexity are key directions for enhancing practical feasibility. Methods such as pruning, dimensionality reduction, and model lightweighting will be employed to reduce computational costs.

In future research, enhancing the model's adaptability and generalization in complex network environments will become a critical focus. Initially, future efforts will prioritize incorporating more diverse anomaly detection mechanisms, such as Isolation Forest and Local Outlier Factor, to strengthen the model's ability to identify unknown malicious traffic, particularly its capacity to detect novel and zero-day attacks. By refining these detection modules, future models will be increasingly sensitive to anomalous behaviors, thereby further enhancing their ability to adapt to emerging unknown threats. Furthermore, data augmentation and adversarial training will play a significant role, as they will simulate a wide array of network environments, traffic types, and attack patterns. By generating training data with diverse feature distributions, the model will learn to recognize more robust features, improving its resilience against perturbations in malicious samples and boosting its generalization capabilities while maintaining stability across various network configurations. Another key area of future research will be cross-domain data integration. By merging traffic data from different network environments, models can be trained to handle a broad spectrum of network security scenarios, ensuring greater adaptability and preserving high accuracy. Future studies will also place greater emphasis on feature selection and dimensionality reduction. Optimizing these processes will help eliminate noise and redundant features, reducing the model's computational complexity while enhancing its sensitivity to critical attack features and ensuring stable performance across varied settings. Lastly, reducing computational complexity will remain a central research focus. Techniques such as pruning, model lightweighting, quantization, and dimensionality reduction can significantly enhance computational efficiency and minimize resource consumption, allowing the model to perform effectively in resource-constrained environments without compromising accuracy. Through in-depth exploration and the implementation of these strategies, future research will drive breakthroughs in malicious traffic detection models, advancing their ability to counter unknown attacks, improve generalization, adapt to diverse network environments, and optimize computational performance.

Author Contributions: Conceptualization, Z.Z.; Validation, Y.M.; Writing—original draft, G.H.; Writing—review & editing, H.Z.; Visualization, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62102312), Key Research and Development Program of Shaanxi (Program No. 2024GX-YBXM-079), the fund of the State Key Laboratory of Integrated Service Networks (Grant No. ISN24-13), the Young Talent fund of University Association for Science and Technology in Shaanxi, China (Grant No. 20210119), The Youth Innovation Team of Shaanxi Universities (Program No. 23JJP160), and Shaanxi Key Laboratory of Information Communication Network and Security Open Project (Grant No. ICNS202402).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, S.H.; Alahmadi, T.J.; Ullah, W.; Iqbal, J.; Rahim, A.; Alkahtani, H.K.; Alghamdi, W.; Almagrabi, A.O. A new deep boosted CNN and ensemble learning based IoT malware detection. *Comput. Secur.* **2023**, *133*, 103385. [\[CrossRef\]](#)
2. Yuan, J.; Zhou, F.; Guo, Z.; Li, X.; Yu, H. HCformer: Hybrid CNN-transformer for LDCT image denoising. *J. Digit. Imaging* **2023**, *36*, 2290–2305. [\[CrossRef\]](#)
3. Chapaneri, R.; Shah, S. Multi-level Gaussian mixture modeling for detection of malicious network traffic. *J. Supercomput.* **2021**, *77*, 4618–4638. [\[CrossRef\]](#)
4. Wang, Z.; Thing, V.L. Feature mining for encrypted malicious traffic detection with deep learning and other machine learning algorithms. *Comput. Secur.* **2023**, *128*, 103143. [\[CrossRef\]](#)
5. Xu, B.; He, G.; Zhu, H. ME-Box: A reliable method to detect malicious encrypted traffic. *J. Inf. Secur. Appl.* **2021**, *59*, 102823. [\[CrossRef\]](#)
6. Ge, Y.; Gao, Y.; Li, X.; Cai, B.; Xi, J.; Yu, S. EMTD-SSC: An Enhanced Malicious Traffic Detection Model Using Transfer Learning Under Small Sample Conditions In IoT. *IEEE Internet Things J.* **2024**, *11*, 30725–30741. [\[CrossRef\]](#)
7. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Deep Learn.* **2015**, *521*, 436–444. [\[CrossRef\]](#)
8. Qu, A.; Shen, Q.; Ahmadi, G. Towards intrusion detection in fog environments using generative adversarial network and long short-term memory network. *Comput. Secur.* **2024**, *145*, 104004. [\[CrossRef\]](#)
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [\[CrossRef\]](#)
10. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500. [\[CrossRef\]](#)
11. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708. [\[CrossRef\]](#)
12. Li, J.; Yu, Z.; Du, Z.; Zhu, L.; Shen, H.T. A comprehensive survey on source-free domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5743–5762. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Scabini, L.F.; Bruno, O.M. Structure and performance of fully connected neural networks: Emerging complex network properties. *Phys. A Stat. Mech. Its Appl.* **2023**, *615*, 128585. [\[CrossRef\]](#)
14. Fang, Y.; Yap, P.-T.; Lin, W.; Zhu, H.; Liu, M. Source-free unsupervised domain adaptation: A survey. *Neural Netw.* **2024**, *174*, 106230. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Ferriyan, A.; Thamrin, A.H.; Takeda, K.; Murai, J. Encrypted malicious traffic detection based on Word2Vec. *Electronics* **2022**, *11*, 679. [\[CrossRef\]](#)
16. Langer, S. Approximating smooth functions by deep neural networks with sigmoid activation function. *J. Multivar. Anal.* **2021**, *182*, 104696. [\[CrossRef\]](#)
17. Zhang, X.; Zhao, M.; Wang, J.; Li, S.; Zhou, Y.; Zhu, S. Deep-forest-based encrypted malicious traffic detection. *Electronics* **2022**, *11*, 977. [\[CrossRef\]](#)
18. Yuan, S.; He, W.; Zhu, Z.; Liu, F.; Song, Z.; Dai, G.; He, G.; Sun, Y. HyCTor: A Hybrid CNN-Transformer Network Accelerator With Flexible Weight/Output Stationary Dataflow and Multi-Core Extension. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**. [\[CrossRef\]](#)
19. Zhang, X.; Wang, Q.; Qin, M.; Wang, Y.; Ohtsuki, T.; Adebisi, B.; Sari, H.; Gui, G. Enhanced few-shot malware traffic classification via integrating knowledge transfer with neural architecture search. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 5245–5256. [\[CrossRef\]](#)
20. Torabi, H.; Mirtaheri, S.L.; Greco, S. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity* **2023**, *6*, 1. [\[CrossRef\]](#)
21. Kondaiah, C.; Pais, A.R.; Rao, R.S. Enhanced malicious traffic detection in encrypted communication using TLS features and a multi-class classifier ensemble. *J. Netw. Syst. Manag.* **2024**, *32*, 76. [\[CrossRef\]](#)
22. Rustam, F.; Jurcut, A.D. Malicious traffic detection in multi-environment networks using novel S-DATE and PSO-D-SEM approaches. *Comput. Secur.* **2024**, *136*, 103564. [\[CrossRef\]](#)
23. Luo, Y.; Chen, X.; Sun, H.; Li, X.; Ge, N.; Feng, W.; Lu, J. Securing 5G/6G IoT Using Transformer and Personalized Federated Learning: An Access-Side Distributed Malicious Traffic Detection Framework. *IEEE Open J. Commun. Soc.* **2024**, *5*, 1325–1339. [\[CrossRef\]](#)

24. Alalwany, E.; Mahgoub, I. Classification of normal and malicious traffic based on an ensemble of machine learning for a vehicle can-network. *Sensors* **2022**, *22*, 9195. [\[CrossRef\]](#)
25. Huo, Y.; Liang, W.; Chen, J.; Zhuang, S.; Sun, J. LightGuard: A Lightweight Malicious Traffic Detection Method for Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 28566–28577. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.