Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060357

ARTICLE

Diff-IDS: A Network Intrusion Detection Model Based on Diffusion Model for Imbalanced Data Samples

Yue Yang[1,2], Xiangyan Tang[2,3,*], Zhaowu Liu[2,3,*], Jieren Cheng[2,3], Haozhe Fang[3] and Cunyi Zhang[3]

[1]School of Cyberspace Security, Hainan University, Haikou, 570228, China
[2]Hainan Province Blockchain Technology Engineering Research Center, Haikou, 570228, China
[3]School of Computer Science and Technology, Hainan University, Haikou, 570228, China

*Corresponding Authors: Xiangyan Tang. Email: tangxy36@163.com; Zhaowu Liu. Email: pigcrazy001@163.com

Received: 30 October 2024; Accepted: 23 December 2024; Published: 06 March 2025

ABSTRACT: With the rapid development of Internet of Things technology, the sharp increase in network devices and their inherent security vulnerabilities present a stark contrast, bringing unprecedented challenges to the field of network security, especially in identifying malicious attacks. However, due to the uneven distribution of network traffic data, particularly the imbalance between attack traffic and normal traffic, as well as the imbalance between minority class attacks and majority class attacks, traditional machine learning detection algorithms have significant limitations when dealing with sparse network traffic data. To effectively tackle this challenge, we have designed a lightweight intrusion detection model based on diffusion mechanisms, named Diff-IDS, with the core objective of enhancing the model's efficiency in parsing complex network traffic features, thereby significantly improving its detection speed and training efficiency. The model begins by finely filtering network traffic features and converting them into grayscale images, while also employing image-flipping techniques for data augmentation. Subsequently, these preprocessed images are fed into a diffusion model based on the Unet architecture for training. Once the model is trained, we fix the weights of the Unet network and propose a feature enhancement algorithm based on feature masking to further boost the model's expressiveness. Finally, we devise an end-to-end lightweight detection strategy to streamline the model, enabling efficient lightweight detection of imbalanced samples. Our method has been subjected to multiple experimental tests on renowned network intrusion detection benchmarks, including CICIDS 2017, KDD 99, and NSL-KDD. The experimental results indicate that Diff-IDS leads in terms of detection accuracy, training efficiency, and lightweight metrics compared to the current state-of-the-art models, demonstrating exceptional detection capabilities and robustness.

KEYWORDS: Network traffic; feature enhancement; diffusion model; multi-classification

1 Introduction

In today's rapidly evolving digital era, analyzing and processing network traffic is crucial for the stable operation of network systems, optimizing performance, and securing information [1]. The widespread adoption of Internet applications and the explosive growth in user numbers have led to an exponential increase in network traffic, along with its complexity and diversity. Consequently, accurately extracting valuable information from massive network traffic data and achieving precise classification have become urgent engineering challenges. Crucial to streamlining network resource management and boosting user satisfaction, this approach is equally indispensable for defending against network intrusions and upholding network integrity.

Recently, a plethora of cutting-edge techniques for intrusion detection have come to the fore [2]. Scholars frequently conceptualize intrusion detection as either a binary classification issue or a multi-class categorization challenge. Following this line of thinking, conventional machine learning algorithms including decision trees, support vector machines, neural networks with multiple layers, and random forests have seen extensive utilization in the realm of intrusion detection [3,4]. However, due to the inherent limitations of traditional machine learning methods, they often perform poorly when handling large-scale, high-dimensional, and complex network data. In contrast, deep learning methods have demonstrated superior performance, particularly in dealing with high-dimensional, complex, and noisy data, thanks to their exceptional representation learning capabilities. As a result, deep learning techniques have been extensively studied for network intrusion detection, with common models including convolutional neural networks (CNNs) [5], recurrent neural networks (RNNs) [6], and autoencoder (AE) [7].

Although deep learning-based methods can automatically extract features, they are limited by the inherent structure of neural networks. The detection performance of these methods heavily relies on a large amount of training data and extensive parameter tuning work [8]. However, information about how hidden layers specifically represent data is relatively scarce, leading to concerns about the quality of feature representations in these layers. In the case of medium to small problems, the learning process may fail or may not provide optimal feature representations [9].

As a novel deep generative technology, diffusion models have demonstrated impressive performance in tasks such as image generation and multimodal generation. We recognize their powerful denoising capabilities and are the first to apply diffusion models to network intrusion detection, proposing an innovative lightweight intrusion detection model. This paper is dedicated to crafting a cutting-edge and high-efficiency technique for the analysis of network traffic data. It proposes the development of a feature mask representation enhancement algorithm, inspired by the Unet framework, which is intended to isolate crucial features from the multifaceted nature of network traffic data to ensure precise classification outcomes. Our main contributions include enhancing sparse data through data augmentation techniques, designing a feature mask data augmentation algorithm, and building a lightweight model. Experimental results show that our model method effectively reduces noise interference, extracts critical structural information from network traffic data, and achieves more accurate classification.

The main academic contributions of this paper can be outlined as follows:

1. We propose an efficient lightweight intrusion detection model based on diffusion models. This model converts network flow data into grayscale images and applies data augmentation. It utilizes a feature mask representation enhancement algorithm constructed with a pre-trained frozen Unet model to improve the model's ability to extract features from complex network traffic.

2. We present a feature mask representation enhancement algorithm that increases the model's robustness when handling noisy and structurally complex feature data, enhancing its adaptability.

3. The model employs a lightweight architecture and utilizes an end-to-end training strategy to improve response speed and operational efficiency.

4. Utilizing NSL-KDD, CIC IDS2017 and KDD 99 as imbalanced datasets for intrusion detection, we carried out comprehensive experiments incorporating a range of assessment criteria. Our proposed Diff-IDS model was compared with several state-of-the-art models, demonstrating superior performance.

The structure of subsequent sections of this paper is organized as follows: Section 2 reviews related work, encompassing traditional intrusion detection techniques and deep learning-based intrusion detection methods. Section 3 provides a detailed description of the model's architecture, including the pre-training of the network traffic diffusion model, the design of the feature mask representation enhancement algorithm,

the end-to-end training process, and the introduction of the lightweight model. Section 4 validates our method through multiple sets of experiments. In conclusion, Section 5 encapsulates the findings of this paper and delineates potential avenues for subsequent research.

2 Related Work

2.1 Network Intrusion Detection Based on Machine Learning

Machine learning techniques are extensively employed in intrusion detection due to their capability to autonomously discern intricate patterns within data. These approaches have been demonstrated to augment the accuracy and rapidity of detection systems, showcase adaptability, and exhibit proficiency in adjusting to emerging attack strategies.

Farooq et al. [10] introduced an intrusion detection system enhanced by an integrated machine learning approach, termed IDS-FMLT. This IDS-FMLT model demonstrated a validation accuracy of 95.18% and a miss rate of 4.82% in its intrusion detection capabilities. Zhang et al. [11] proposed a privacy-preserving anomaly-based intrusion detection system for future IIoT networks, leveraging federated learning. This method tackles the critical challenge of training local models with non-independent and identically distributed (non-IID) data. Sezign et al. [12] crafted an automated machine learning (AutoML) protocol designed to fortify the efficacy of intrusion detection systems in identifying and neutralizing threats. This protocol streamlines the machine learning process, thereby ensuring the sanctity, security, and confidentiality of data traversing IIoT networks. Wang et al. [13] proposed an innovative intrusion detection methodology for IIoT, christened MTID, which is anchored in the temperature profiles of microcontroller units (MCU). This method employs an online incremental learning paradigm to ensure the model's pertinence across a spectrum of IIoT deployment contexts. Li et al. [14] proposed an intrusion detection system based on joint learning, called DAFL. This system has been lauded for its prowess in detecting intrusions, concurrent with a substantial reduction in communication expenditures. Thockchom et al. [15] proposed an intrusion detection model based on ensemble learning. The proposed ensemble model employs Gaussian Naive Bayes, logistic regression, and decision trees as its foundational classifiers, while stochastic gradient descent serves as the meta-classifier. The findings show that employing this ensemble classifier is highly effective in managing imbalanced datasets within intrusion detection systems. Wu et al. [16] have crafted a compact machine learning-based intrusion detection system that utilizes a feature selection technique to pinpoint the most effective features, thereby diminishing the computational burden of the IDS. This proposed approach is equipped to tackle intrusion detection challenges even in situations where there is incomplete data availability. Hu et al. [8] proposed an early and accurate network intrusion detection method called Graph2vec+RF, which is based on graph embedding technology. This approach autonomously extracts flow graph features by leveraging subgraph structures and operates with just a minimal number of initial interactive packets for each bidirectional network flow, thereby obviating the requirement for an extensive array of training samples to facilitate prompt and precise network intrusion detection.

However, the majority of machine learning algorithms are founded on shallow learning, which encounters difficulties when tasked with classifying vast and high-dimensional data sets [17]. As the volume and intricacy of data escalate, conventional machine learning tactics have unveiled their shortcomings in managing high-dimensional and non-linear data. For instance, methodologies predicated on the Incremental Support Vector Machine (ISVM) algorithm endeavor to retain non-support vectors during the incremental classification process through selective preservation [18]. While these methodologies have provided valuable insights for bolstering detection efficacy, they remain susceptible to the challenges inherent in data imbalance, frequently neglecting instances from minority classes, which consequently leads to reduced detection rates for these classes.

2.2 Network Intrusion Detection Based on Deep Learning

As computational power and computer hardware have evolved, deep learning methods have gained extensive application in the field of network intrusion detection. Recently, a growing number of scholars have turned to deep learning algorithms to tackle the challenges posed by intrusion detection. Sun et al. [19] introduced a Logic Understanding Intrusion Detection System, a rule-based IDS that deeply comprehends industrial control logic. This system employs custom deep learning models to autonomously extract features and categorize attacks. Wang et al. [20] Experiments on the CIC-IDS-2017 and UNSW-NB15 datasets indicate that K-GetNID performs comparably to deep learning methods in terms of adjustable early intrusion detection and transferability. Chen et al. [21] put forward an Information-Aware Adversarial Domain Adaptation approach, capable of training cross-domain industrial intrusion detection deep models in the face of imbalanced data, yet still preserving high detection precision. Gong et al. [22] introduced a two-stage deep learning model that utilizes a multi-agent reinforcement learning framework to decrease detection time and enhance accuracy. Li et. al [23] developed an adversarial environment learning-based model, AE-SAC, aimed at enhancing the detection rate of infrequent network attacks. When pitted against current sophisticated intrusion detection algorithms on the NSL-KDD dataset, AE-SAC garnered an accuracy of 84.15% and an Fl-measure of 83.97%. Wu et al. [24] introduced an innovative active learning framework based on Deep Q-Networks, highlighting its broad applicability and versatility. This framework plays a crucial role in accurately pinpointing and detecting network intrusion activities. Thakkar et al. [25] employed a Bagging classifier, using Deep Neural Networks as the base estimator, to tackle the issue of class imbalance in intrusion detection datasets. Ye et al. [26] put forth a deep reinforcement learning-based intrusion detection approach, capitalizing on the adaptive nature of reinforcement learning agents in interaction with their environment. This method has shown robust adaptability to environmental noise and effectively tackles intrusion detection challenges in IoT boundaries. Shahriar et al. [27] introduced CAN Shield, a deep learning framework designed for intrusion detection at the CAN bus signal level. Assessments conducted on two CAN attack datasets demonstrate that CANShield possesses a high level of accuracy and promptness in identifying sophisticated intrusion attacks.

However, current methods often lead to a high rate of missed detections and low recognition rates for minority class attacks when handling imbalanced data. In such cases, the performance of deep learning models is limited, preventing them from fully extracting critical information. In contrast, diffusion models demonstrate unique advantages in handling complex data structures and noise. By effectively modeling the distribution of data and the relationships between features, they can better capture the subtle differences of rare-class samples. Furthermore, diffusion models possess adaptive learning capabilities, allowing them to dynamically adjust parameters in imbalanced data environments to enhance the accuracy and robustness of detection.

Therefore, the introduction of diffusion models in the latest intrusion detection technologies provides new insights for handling imbalanced data. Greidanus et al. [28] proposed an intrusion detection strategy utilizing diffusion models that can quickly respond to attacks. Wang et al. [29] introduced a network intrusion detection method based on DDPM, which reduces the need for extensive annotations by using an unsupervised reconstruction error approach. Krishnasamy et al. [30] proposed a dual-interaction Wasserstein generative adversarial network that incorporates anisotropic diffusion Kuwahara filtering techniques in the preprocessing unit to consider time series. However, despite the potential of diffusion models to become the preferred choice in intrusion detection, existing methods still face some common issues. First, there is insufficient modeling capability for complex data structures, and second, learning efficiency is low in high noise and imbalanced data environments. These issues limit the effectiveness of diffusion models in practical intrusion detection scenarios. Based on this consideration, we propose the Diff-IDS model to address the

shortcomings of existing methods in extracting features from rare classes. This model introduces a Unet-based feature mask representation enhancement algorithm that effectively processes noise and complex feature structures in network traffic data. By efficiently leveraging the information provided by noise during the denoising process, our model can more accurately capture the features of rare-class samples, thereby significantly improving the overall performance of network intrusion detection.

3 Method

3.1 Enhanced Data Representation for Intrusion Detection

In real network environments, there is a significant imbalance between attack traffic and normal traffic, which can easily lead to model bias, poor generalization, and other issues. Therefore, before inputting data into the model for learning, we first perform data augmentation to reduce the distortion of evaluation metrics caused by extreme data imbalance. We use the CI CIDS2017 dataset for representation processing and provide examples. First, we calculated the correlation matrix between features to identify and remove highly correlated feature columns. Specifically, we used the Pearson correlation coefficient to quantify the correlation between features. For features with an absolute correlation coefficient of 1, we retained one and removed the others to reduce the negative impact of multicollinearity on model performance. By removing these highly correlated features, we selected a more independent feature set from the original features, reduced the dimensionality of the model input, and improved training efficiency. To use these datasets in image-based deep learning models, we converted the selected feature sets into grayscale image format. The values are arranged in a two-dimensional matrix to represent the grayscale image. An example of these images is shown in Fig. 1.

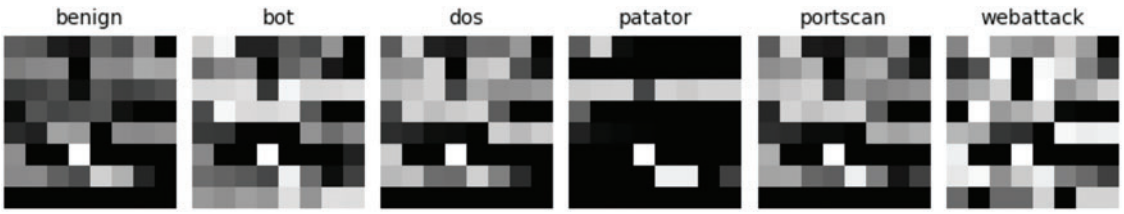

Figure 1: Grayscale representation of network traffic

Given that we have converted network traffic data into grayscale images, we further employed image-flipping augmentation techniques, especially for grayscale images of minority classes, including horizontal and vertical flipping. Flipping techniques have spatial significance in image processing, but in network traffic data, they simulate the reordering of features. This reordering enhances the model's robustness, allowing it to better handle and recognize features in the data. It not only improves the model's ability to recognize minority classes but also enhances its overall sensitivity to data features. Finally, we integrate these augmented data into the original dataset, providing richer and more balanced data support for subsequent model training.

3.2 Generative Diffusion Model for Unet Pre-Training

The diffusion model is a type of probabilistic generative model, primarily consisting of a forward diffusion phase and a reverse diffusion phase. During the forward diffusion phase, the model gradually introduces Gaussian noise, transforming the original data samples into a noise distribution. Specifically, the change in data at each time step is described by the following equation:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right), \tag{1}$$

In the reverse diffusion phase, the model learns how to reverse this process and recover the original data from the noise. The reverse diffusion process is described by the following equation:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{2}$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the mean and covariance functions learned by the model. During the sampling process, we use the U-Net network as a denoising model. The U-Net network, a deep learning architecture widely applied in diffusion models, has demonstrated exceptional performance in image segmentation and reconstruction tasks. This network is known for its unique symmetric U-shaped structure, where the contracting path is responsible for capturing contextual information, and the expansive path focuses on precise localization, enabling U-Net to effectively extract and fuse features at multiple scales. Additionally, U-Net combines low-level features from the encoder with high-level features from the decoder through skip connections, a design that enhances the network's ability to preserve details and textures during the denoising process, providing insights for us to extract structural information of network data from noisy data.

To achieve better training results, we utilized a U-Net architecture autoencoder to predict the noise at each time step during the diffusion process. The training objective of the model is to optimize parameters by minimizing the following loss function:

$$L_{\text{simple}} = E_{x_0,\varepsilon,t}\left[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\right], \tag{3}$$

where ε represents the noise added to the data, and $\varepsilon_\theta(x_t, t)$ is the noise predicted by the model. By minimizing this loss function, the model can effectively reverse the forward diffusion process and generate new samples that closely resemble the original data.

3.3 Feature Masking Enhancement Algorithm

The overall framework of Diff-IDS is shown in Fig. 2. Data is input into the model, and we consider the data as the noise data to be denoised at the t-th step (where t is a random time step from 0 to 20), denoted as x_t. x_t is input into the U-Net network with frozen parameters, and the predicted noise $\varepsilon_\theta(x_t, t)$ is output.

To effectively process the noise data output by U-Net and facilitate subsequent classification, we designed a feature mask data augmentation algorithm, with the specific details shown as Algorithm 1. First, we enhance the noise features using a masking method. Specifically, the sigmoid function is used to map the noise features to a probability distribution, making the differences in the data more distinct. The sigmoid function is defined as follows:

$$\sigma(x_t, t) = \frac{1}{1 + e^{-\varepsilon_\theta(x_t, t)}}. \tag{4}$$

By applying the sigmoid function, each noise feature value $\varepsilon_\theta(x_t, t)$ is converted into $\sigma(x_t, t)$, a probability value between 0 and 1. This process effectively transforms and highlights the complex, noisy features, making the data features more prominent.

After completing the masking process, we use the masked information to weight the original data. The specific steps are as follows:

The original noise data $\varepsilon_\theta(x_t, t)$ is element-wise multiplied by the mask feature obtained through the sigmoid function $\sigma(x_t, t)$, forming the weighted data:

$$x' = \varepsilon_\theta(x_t, t) \odot \sigma(x_t, t), \tag{5}$$

where x' represents the weighted data. This operation ensures that the original data corresponding to larger values in the masked features has a greater impact on the weighted result, while smaller values reduce their influence accordingly.

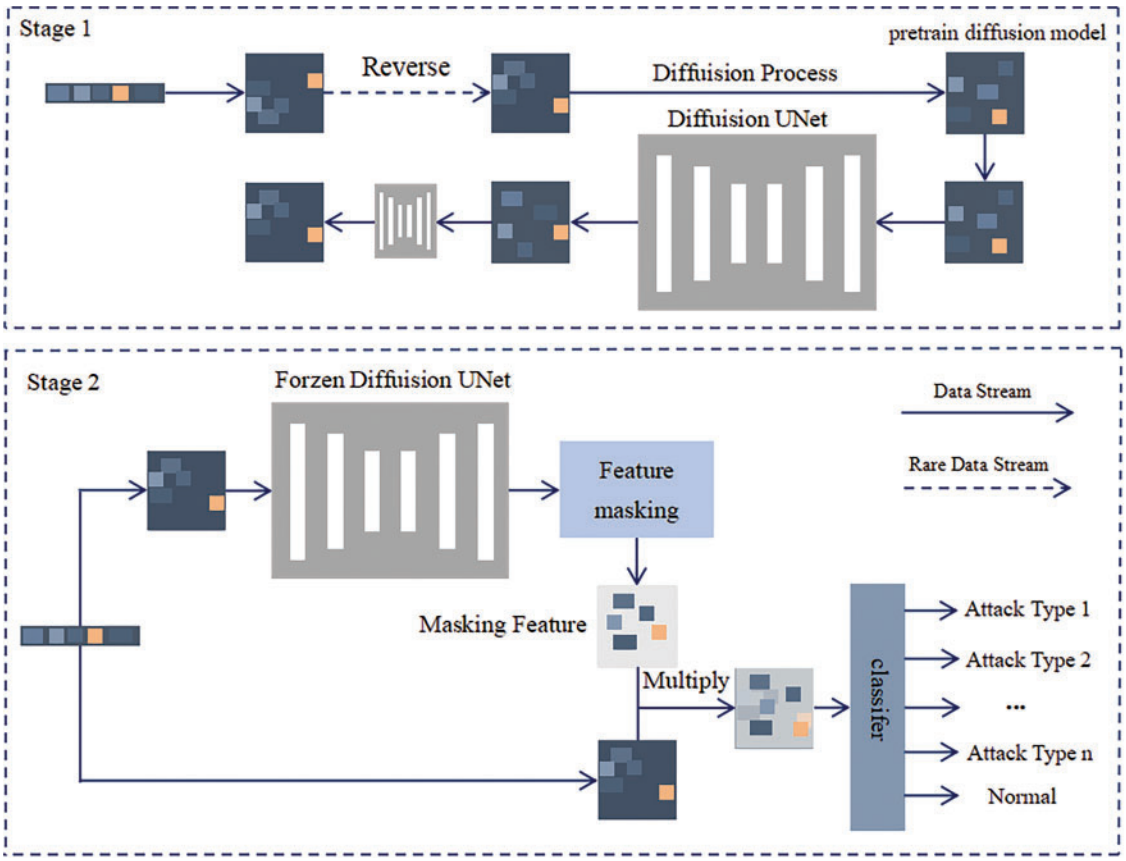

Figure 2: Diff-IDS model architecture diagram

Algorithm 1: Feature masking representation enhancement

Input: Training sample set X, number of classes N, number of training steps for the classification model Epoch, total training iterations T
Output: Enhanced features $\mathbf{x}''$
Initialize model parameters $\theta_{Unet}, \theta_{Mask}, \theta_{Cls}$
while iteration count $t < T$ **do**
  for $i = 1$ to t **do**
    Draw a batch of input samples from the training set: $X_i \sim P(X)$
    Randomly select a time step $ts \sim \{0, 1, \ldots, 20\}$
    Extract features using the frozen U-Net: $F = Unet(X_i, t)$
    Apply the sigmoid function to the features to generate a mask: $M = \text{Sigmoid}(F)$
    Multiply original input and masked correspondence: $\mathbf{x}'' = X_i \cdot M$
  end for
end while
Return $\mathbf{x}''$

Further enhancement of the features can be computed as follows:

$$x'' = \frac{x'}{\max(x')}. \tag{6}$$

Here x'' is the final enhanced feature, and $\max(x')$ denotes the maximum value in the weighted data, used for normalization to make the data features more distinct.

Ultimately, the features extracted are fed into the final one-dimensional convolutional network for classification, yielding the ultimate classification outcomes. The results demonstrate that our approach exhibits superior performance and accuracy.

3.4 End-to-End Optimized Model Training

To ensure the generalization capability of the model and to prevent overfitting, we adopted an end-to-end training strategy. This approach allows the model to automatically adjust its parameters during the training process to minimize prediction errors. During the classification training phase, as the goal is to create a multi-class classification model, we employ one-hot encoding for categorical labeling. This method of labeling signifies the most desirable output for the model. The loss function can be mathematically expressed as follows:

$$L = -\sum_{i=1}^{N}\sum_{c=1}^{K} y_{ic}\log(p_{ic}) \tag{7}$$

where L is the loss value, N is the number of samples, K is the number of categories, y_{ic} is the true label of whether the i-th sample belongs to category c (one-hot encoded), and p_{ic} is the probability that the model predicts the i-th sample belongs to category c. This loss function measures the difference between the model's predicted probability distribution and the true labels. The complete end-to-end optimized model training process is shown in Algorithm 2.

Algorithm 2: End-to-end optimized model training

Input: Training sample set X, Labels Y, number of classes N, number of training epochs E, batch size B, learning rate α, Feature Masking Representation Enhancement Algorithm M, One-dimensional convolutional classifier C, total training iterations T
Output: Predicted Labels y
Load Algorithm M
Initialize Classifier C and optimizer Adam with learning rate α
for epoch = 1 to E **do**
  for each batch (X_B, Y_B) in X **do**
    Output enhancement features (grayscale images): $\mathbf{x}'' = M(X_B, T)$
    Forward pass: Compute predictions $y_B = C(\mathbf{x}'')$
    Compute loss $\mathscr{L} = \text{Loss}(y_B, Y_B)$
    Backward pass: Compute gradients $\nabla_\theta \mathscr{L}$
    Update model parameters using the optimizer: $\theta = \theta - \alpha \cdot \nabla_\theta \mathscr{L}$
  end for

(Continued)

Algorithm 2 (continued)

13: **end for**
14: Return y

During the training phase, the U-Net model is designed to output grayscale images of network traffic, which capture the structural features of the network traffic. These features are significantly different from other types of network traffic images in high-dimensional space. To enhance the model's ability to recognize minority classes, we employed a feature masking enhancement algorithm, which combines blurred features with original features to increase the discriminability of minority class features. In the final stage, we abandoned traditional activation and classification functions, instead using one-dimensional convolution and a single linear layer to classify the enhanced features directly, thus forming an efficient end-to-end training method.

3.5 Lightweight Two-Stage Classification Model

In the field of cyberspace security, faced with the vast and complex network traffic data, the development of a lightweight detection model is particularly urgent. Such a model can quickly process large amounts of data, enabling efficient monitoring and analysis of network traffic. Therefore, we have designed a staged model architecture aimed at achieving efficient and accurate network traffic detection.

Our model development is divided into two stages. In the first stage, we perform only the pretraining of the diffusion model, which aims to initialize the U-Net network and freeze its parameters, laying the foundation for subsequent feature extraction and the construction of the lightweight model. The lightweight design is primarily focused on the second stage, with the goal of optimizing the model structure, reducing the number of parameters, and computational complexity.

During the data processing phase, we convert network traffic data into an $8 \times 8 \times 1$ grayscale image format. For data that is insufficient to fill this format, we process it with zero padding to ensure data consistency. In the second stage, we adopt an innovative training strategy: during training, the weights of the U-Net network remain unchanged, significantly shortening the training time. Moreover, the entire model contains only two components with parameters: the U-Net network and a one-dimensional convolutional neural network (1D CNN). This design not only diminishes the count of model parameters but also lessens the requirement for computational resources, thereby rendering the model more appropriate for implementation in practical network intrusion detection environments.

4 Experimental

4.1 Experimental Setup

4.1.1 Datasets

To address the issue of imbalanced data samples encountered during intrusion detection, which can lead to a decline in model performance across various metrics, this paper selects two datasets to validate the effectiveness and performance of our proposed method. The chosen datasets include real-time network traffic data and exhibit significant imbalances in data sample quantities across multiple aspects.

The CIC-IDS 2017 is a dataset developed in collaboration between the Canadian Communication Security Establishment and the Canadian Institute for Cybersecurity, primarily used for network traffic analysis and intrusion detection research. It encompasses five days of network traffic data, including normal traffic and various common network attacks such as brute force, DoS attacks, and Heartbleed vulnerability

exploitation, providing a rich and real-world experimental sample for the field of network security, aiding researchers and developers in better understanding and defending against network threats.

The KDD 99 dataset is a dataset used in the 1999 KDD Cup organized by DARPA of the United States. The dataset comprises approximately 4.9 million network connection records, including normal connections and various types of attacks. It features 41 attributes, which include basic features, traffic features, time-based features and host-based features. The KDD 99 dataset is a crucial benchmark in the field of network security for anomaly detection and intrusion detection, and it is widely used in machine learning and data mining research.

NSL-KDD is a dataset used for network security research and the evaluation of Intrusion Detection Systems (IDS). It is an improved version of the original KDD Cup 99 dataset. NSL-KDD addresses some issues present in the KDD Cup 99 dataset, such as data redundancy and uneven distribution of attack samples, by providing more balanced training and testing datasets. It includes normal network traffic as well as various types of network attacks, such as Denial of Service, Probe, and Privilege Escalation, allowing researchers to develop and evaluate IDS models in a more realistic environment. The distribution of training and test sets for the two representative datasets is shown in Tables 1–3.

Table 1: CIC IDS 2017 data distribution

Category	Training set	Test set
Benign	329,959	109,724
DoS hulk	172,318	57,806
DoS goldeneye	7739	2554
DoS slowloris	4358	1438
DoS slowhttptest	4170	1329
Heartbleed	10	1
Total	518,554	172,852

Table 2: KDD 99 data distribution

Category	Training set	Test set
Normal	97,277	60,592
DoS	391,438	229,825
Probe	4107	4166
R2L	1126	16,189
U2R	52	228
Total	494,000	311,000

Table 3: NSL-KDD data distribution

Category	Training set	Test set
Normal	67,343	9711
DoS	45,927	7460
Probe	11,656	2421

(Continued)

Table 3 (continued)

Category	Training set	Test set
R2L	995	2885
U2R	52	67
Total	125,973	22,544

4.1.2 SOTA Models

We comprehensively assess our proposed model by comparing it with various state-of-the-art models, each model utilizing different techniques such as feature selection, SVM, self-supervised learning, semi-supervised learning, neural networks, and knowledge distillation for efficient and effective intrusion detection with a focus on reducing complexity and enhancing performance in IoT networks. The aforementioned methods can be categorized into network intrusion detection models based on machine learning and deep learning. A comparison of the parameters of these models with those of the model proposed in this paper is shown in Table 4.

Table 4: Comparison of implementation parameters between deep learning models and the Diff-IDS

Model	Optimizer	Parameters	Batch size	Lr	Epoch
LOGNN [31]	Adam	12,333	512	1e-4	20
SDAE-ELM [32]	Adam	44,293	100	1e-4	100
LSTM-FCNN [33]	Adam	4624	10	1e-4	40
IBYOL-IDS [34]	Adam	1,578,145	512	1e-4	20
KD-TCNN [35]	Adam	12,333	512	1e-4	40
CL-SKD [17]	AdamW	14,494	1024	1e-3	20
SS-Deep-ID [36]	Adam	663,434	512	3e-4	20
Diff-IDS	Adam	778	128	2e-4	30

4.1.3 Evaluation Metrics

We used the following four metrics to evaluate model performance: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Using these four metrics, we derived four performance indicators for evaluating the model: Accuracy, Precision, Recall, F1-measure, FPR (False Positive Rate) and FNR (False Negative Rate). The calculation formulas are as follows:

$$\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FP} \tag{8}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{F1-measure} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \tag{11}$$

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

$$FNR = \frac{FN}{TP + FN} \tag{13}$$

These metrics help us comprehensively assess the model's performance in handling imbalanced datasets, ensuring that the model can effectively detect various types of network attacks in real-world applications.

4.2 Model Training

The changes in loss during each phase of model training are shown in Fig. 3. It can be observed that as the number of training epochs increases, the model's performance improves steadily without any signs of overfitting. Additionally, since we have completely separated the test set from the training set, there is no issue of data leakage.

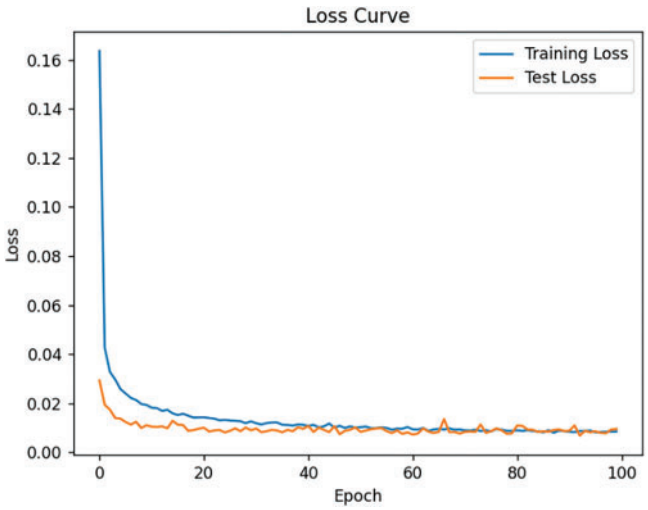

Figure 3: Training and testing loss performance

4.3 Effectiveness of Data Augmentation

In the methods section, we convert the data to grayscale images to enable the model to better extract key features from different types of data. Due to the imbalance of samples in network intrusion behaviors, it is necessary to perform data augmentation on the samples. In this section, we will compare the detection efficiency of our model in binary and multi-class classification before and after applying the data augmentation algorithm to validate the effectiveness of our data augmentation. The specific classification outcomes are illustrated in Fig. 4.

The comparison result diagram is shown in Fig. 4. The figure on the left illustrates the binary classification results of our model before and after enhancement, while the figure on the right displays the multi-class classification outcomes. By transforming the data into grayscale images and employing image data augmentation techniques to enrich the network traffic data, we have successfully mitigated the discrepancies in the number of different attack types. This approach has bolstered the model's learning capacity, leading to an improvement across all performance metrics.

It can be observed from Table 5 that Diff-IDS-E is a model trained with data augmentation. Due to the expansion of the training set through data augmentation, the training time of the model has increased. However, we have achieved better accuracy at the cost of a slight increase in training time, which is an

acceptable trade-off. Moving forward, we will use the Diff-IDS-E model trained with data augmentation as our benchmark model.

Figure 4: Comparison results of the model before and after data augmentation in the CIC IDS2017

Table 5: Detection metrics for different classification tasks before and after data augmentation

Model	Acc	Pre	Recall	F1	Training time
Diff-IDS_binary	99.93	99.93	99.93	99.93	15.93
Diff-IDS-E_binary	99.94	99.94	99.94	99.94	16.07
Diff-IDS_multi	99.89	99.89	99.89	99.89	15.96
Diff-IDS-E_multi	99.93	99.93	99.93	99.93	16.34

4.4 Comparison of Lightweight Models

To verify the lightweight effect of the Diff-IDS model, we have employed traditional deep learning models such as DNN, CNN, and RNN to perform multi-classification tasks on the CIC IDS2017 dataset. In addition to accuracy, F1-measure, FNR, FPR, the number of model parameters, model size, and flops are also used as indicators for model lightweightness.

The classification results are shown in Table 6. The Diff-IDS model excels in both classification precision and lightweight model metrics. At a similar level of accuracy, our model boasts a reduced number of parameters and floating-point calculations compared to DNN and RNN, resulting in a 76-fold decrease in model size. In the context of comparable F1-measure, our model outperforms CNN across all lightweight criteria. Although the distribution of the CIC IDS2017 dataset is imbalanced, the number of attack types is not extensive, which allows all methods to achieve relatively good detection effects. However, in terms of model size and other metrics, it is evident that our model, which employs one-dimensional convolution as a lightweight classifier, demonstrates superior detection efficiency and lighter metrics standards.

Table 6: Comparison of lightweight NID models based on different deep learning networks

Model	Acc	F1	FNR	FPR	Params	Size	Flops
DNN	99.83 (↓0.10)	90.99 (↓8.94)	0.17	0.03	4758	229 KB	150,528
CNN	99.42 (↓0.51)	99.41 (↓0.52)	0.58	0.20	349,229	1.33 MB	2,295,070,720
RNN	99.83 (↓0.10)	91.52 (↓8.41)	0.17	0.05	58,630	0.22 MB	3,743,744
Diff-IDS	99.93	99.93	0.07	0.01	778	3 KB	40,960

4.5 The Classification Effect of the Model

4.5.1 Binary Classification

We classify all attack traffic as the anomaly class, thereby transforming intrusion detection into a binary classification task. We utilize the KDD 99,NSL-KDD and CIC IDS2017 datasets to benchmark the Diff-IDS model against the most recent cutting-edge models, aiming to substantiate the superiority of the model proposed in our research.

Table 7 indicates that all models achieve an accuracy of over 96%, largely due to the CIC IDS 2017 dataset imbalanced distribution, which, although addressed, is not excessively skewed, enabling all models to achieve sound classification results. In precision, the Diff-IDS model leads with a top score of 99.94%, outperforming the IBYOL-IDS model by 4.70%. Furthermore, the Diff-IDS model also yields the best outcomes in terms of recall and F1-measure. From Table 8, we can observe that on the KDD 99 dataset, the metrics of our model approach a level close to 1, reaching 99.97%. This is because the KDD 99 dataset has a relatively sufficient sample size, allowing the model to be adequately trained. After in-depth feature mining, our model has been able to effectively distinguish between attack traffic and normal traffic. In Table 9, although the attack types are the same as KDD 99, the reduction in the amount of data in the NSL-KDD dataset has led to a decrease in the accuracy of all models. However, the model we proposed still maintains an exceptionally high detection capability of 99.84%. All other three metrics significantly outperform the comparative models, with the recall rate even 35.72% higher than that of the SDAE-ELM model. Due to our use of data augmentation to bridge the distribution gap between attack traffic and normal traffic in the training set, and the fact that our network is capable of effectively integrating feature information from different levels while maintaining precise representation of data details, our results are significantly superior to other methods.

Table 7: The binary classification performance of Diff-IDS and SOTA models on CIC IDS2017

Model	Acc	Pre	Recall	F1
IBYOL-IDS [34]	96.70 (↓3.24)	95.24 (↓4.70)	95.76 (↓4.18)	95.50 (↓4.44)
NB-SVM [37]	99.35 (↓0.59)	–	99.24 (↓0.70)	–
SS-Deep-ID [36]	99.33 (↓0.61)	99.47 (↓0.47)	99.23 (↓0.71)	99.35 (↓0.59)
CL-SKD [17]	99.80 (↓0.14)	99.81 (↓0.13)	99.80 (↓0.14)	99.80 (↓0.14)
Diff-IDS	99.94	99.94	99.94	99.94

Table 8: The binary classification performance of Diff-IDS and SOTA models on KDD 99

Model	Acc	Pre	Recall	F1
IBYOL-IDS [34]	99.25 (↓0.72)	99.74 (↓0.23)	99.33 (↓0.64)	99.53 (↓0.44)
SDAE-ELM [32]	93.57 (↓6.40)	98.69 (↓1.28)	93.19 (↓6.78)	95.86 (↓4.11)
CL-SKD [17]	99.95 (↓0.02)	99.95 (↓0.02)	99.95 (↓0.02)	99.95 (↓0.02)
Diff-IDS	99.97	99.97	99.97	99.97

Table 9: The binary classification performance of Diff-IDS and SOTA models on NSL-KDD

Model	Acc	Pre	Recall	F1
IBYOL-IDS [34]	92.67 (↓7.17)	92.48 (↓7.36)	92.06 (↓7.78)	92.27 (↓7.57)
NB-SVM [37]	92.35 (↓7.49)	–	99.24 (↓0.60)	–

(Continued)

Table 9 (continued)

Model	Acc	Pre	Recall	F1
SDAE-ELM [32]	78.04 (↓21.80)	95.99 (↓3.85)	64.12 (↓35.72)	76.87 (↓22.97)
CL-SKD [17]	99.43 (↓0.41)	99.43 (↓0.41)	99.43 (↓0.41)	99.43 (↓0.41)
Diff-IDS	99.84	99.84	99.84	99.84

The ROC curve is not affected by imbalanced class distribution and is a very useful evaluation tool when dealing with datasets with uneven positive and negative sample ratios. As shown in Fig. 5, our AUC value reaches 0.9997 on KDD 99. On the NSL-KDD and CIC IDS2017 datasets, due to the addition of some new attack types, the difficulty in identifying normal traffic and attack traffic increases. The performance metric of Diff-IDS on these datasets is slightly lower, with an AUC value of 0.9984 and 0.9994, respectively. Overall, the ROC curves from the three datasets indicate that our model exhibits excellent classification performance.

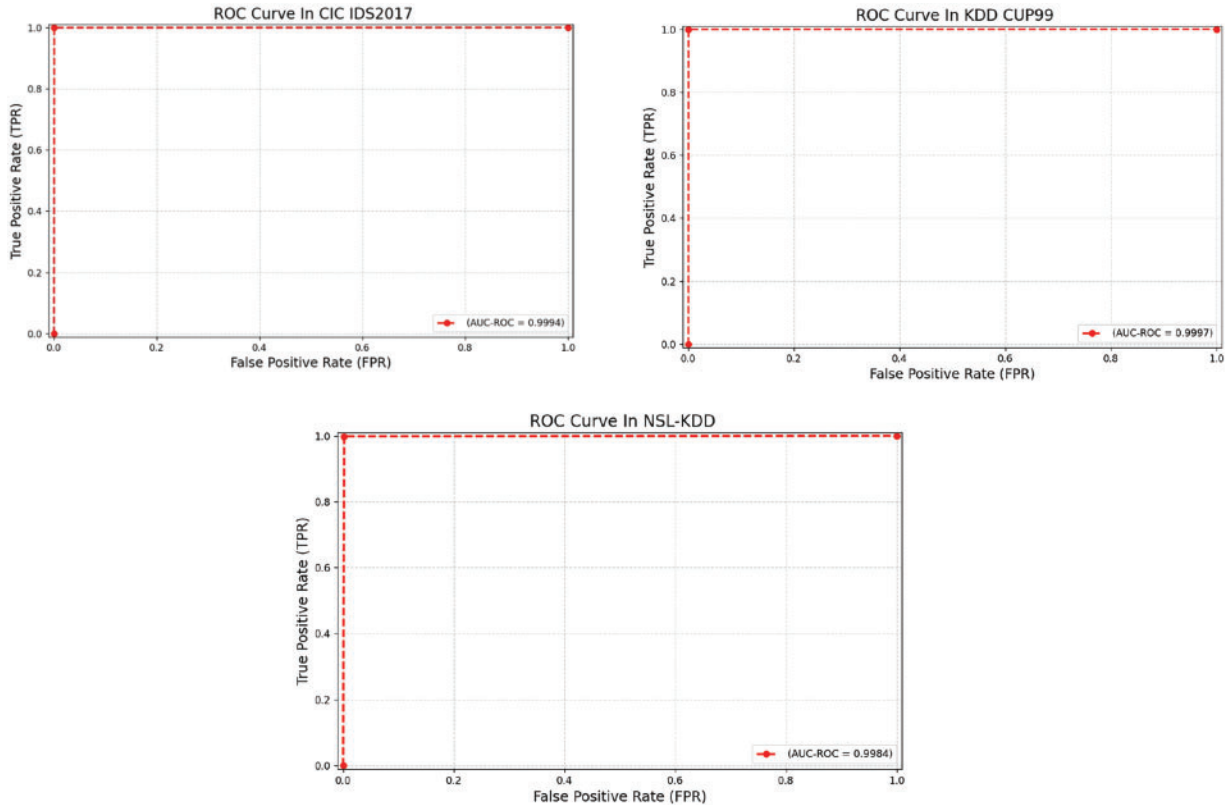

Figure 5: The ROC curve of Diff-IDS on CIC IDS 2017, KDD 99 and NSL-KDD

4.5.2 Multi Classification

The multi-class performance of the model on imbalanced samples is crucial for the ability to specifically identify various types of attacks. We will compare the proposed model with state-of-the-art models on each dataset in the coming year to demonstrate the effectiveness and robustness of our approach.

As shown in Table 10, even as the difficulty of the classification task increases from binary to multi-class classification, our method maintains an accuracy rate of 99.93% on the CIC IDS2017 dataset. The

experimental results show that, except for a 0.07% lower precision compared to LSTM-FCNN, Diff-IDS outperforms all other algorithms in the remaining metrics. Furthermore, in the KDD 99 and NSL-KDD datasets shown in Tables 11 and 12, our multi-classification metrics all achieve above 99%, far surpassing the performance of all comparative methods, demonstrating the model's efficient detection capability for various types of attacks in multiple imbalanced network intrusion scenarios. Due to our proposed feature mask representation enhancement algorithm, which effectively captures the intrinsic characteristics of anomalous traffic data, even underrepresented minority class attacks can have their internal features extracted, enhancing the model's representational capability. This has led to the achievement of such remarkable detection capabilities.

The confusion matrix depicted in Fig. 6 presents a heatmap that illustrates various types of attacks, utilizing color variations and brightness levels. This visualization aggregates the actual and predicted outcomes of the dataset, providing insight into the model's performance strengths. Due to our model's ability to effectively extract fine-grained features and deeply mine the underlying information in the data, we have improved the representation and recognition capabilities of minority class attacks in imbalanced multi-classification tasks. As can be seen from the confusion matrix, we demonstrate precise detection capabilities in identifying minority class attacks such as Probe and R2L.

Table 10: The multi-classification performance of Diff-IDS and SOTA models on CIC IDS2017

Model	Acc	Pre	Recall	F1
KD-TCNN [35]	99.44 (↓0.49)	99.48 (↓0.45)	99.47 (↓0.46)	99.46 (↓0.47)
LSTM-FCNN [33]	99.49 (↓0.44)	100 (↑0.07)	99.60 (↓0.33)	99.70 (↓0.23)
CCFS-DT [38]	97.91 (↓2.02)	99.42 (↓0.51)	99.07 (↓0.86)	99.24 (↓0.69)
CL-SKD [17]	99.84 (↓0.09)	99.84 (↓0.09)	99.84 (↓0.09)	99.84 (↓0.09)
Diff-IDS	99.93	99.93	99.93	99.93

Table 11: The multi-classification performance of Diff-IDS and SOTA models on KDD 99

Model	Acc	Pre	Recall	F1
SVM [39]	98.00 (↓1.09)	98.75 (↓0.32)	97.75 (↓1.34)	98.75 (↓0.28)
DT [39]	98.50 (↓0.59)	95.25 (↓3.82)	96.25 (↓2.84)	95.75 (↓3.28)
LSTM-FCNN [33]	98.52 (↓0.57)	98.90 (↓0.17)	98.70 (↓0.39)	98.70 (↓0.33)
Diff-IDS	99.09	99.07	99.09	99.03

Table 12: The multi-classification performance of Diff-IDS and SOTA models on NSL-KDD

Model	Acc	Pre	Recall	F1
LSTM-FCNN [33]	98.94 (↓0.08)	98.90 (↓0.11)	99.10 (↑0.08)	99.00 (↓0.01)
KD-TCNN [35]	98.44 (↓0.58)	98.60 (↓0.41)	98.47 (↓0.55)	98.51 (↓0.50)
LOGNN [31]	98.70 (↓0.32)	98.40 (↓0.61)	98.70 (↓0.32)	98.50 (↓0.51)
Diff-IDS	99.02	99.01	99.02	99.01

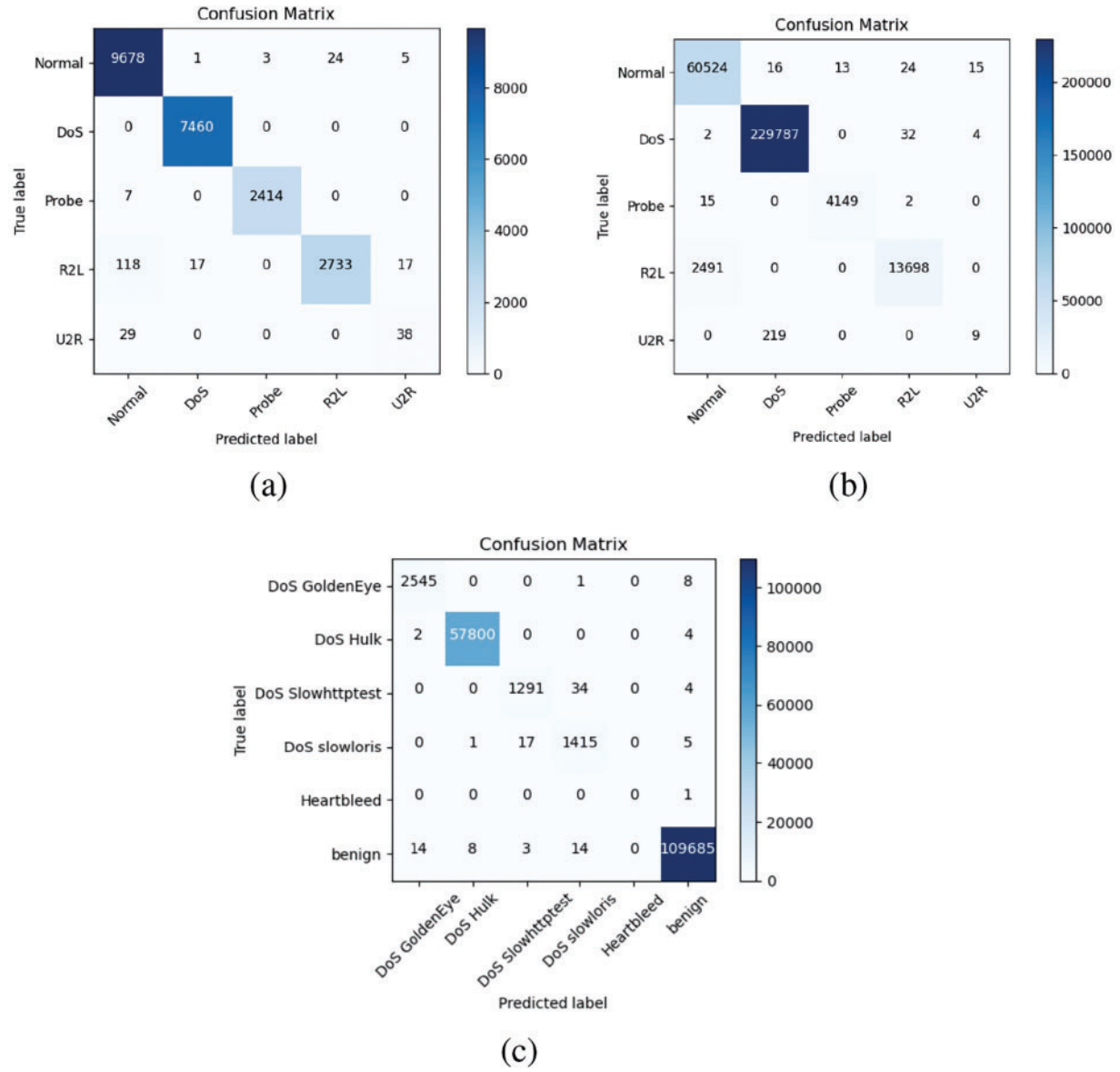

Figure 6: The confusion matrix of Diff-IDS on NSL-KDD (a), KDD 99 (b) and CIC IDS 2017 (c)

5 Summary and Outlook

This paper proposes a novel network intrusion detection model named Diff-IDS, which is based on the diffusion model. It encompasses the visualization and augmentation of network flow data, feature masking techniques based on Unet, and an end-to-end lightweight intrusion detection strategy, effectively addressing the issues of poor training and detection accuracy caused by imbalanced network intrusion datasets, while also achieving lightweight processing. Our model has been tested across multiple datasets including CIC IDS2017, KDD 99, and NSL-KDD, and its high detection efficiency and robustness have been verified through comparisons with state-of-the-art methods.

In future work, we plan to further research more advanced deep learning models and algorithms to optimize the structure and parameters of our diffusion model, thereby improving the accuracy and efficiency of detection. Additionally, we aim to explore the integration of network traffic data with other types of

security data, such as log files and user behavior data, to develop a more comprehensive and precise intrusion detection system. Although Diff-IDS has achieved outstanding results in network traffic classification, there is still room for improvement in terms of hyperparameter selection and model lightweight. We plan to refine the parameters using hyperparameter optimization techniques, aiming to decrease the model's complexity and computational expense through methods such as model quantization, network pruning, and feature selection. Moreover, we acknowledge the challenge in determining the statistical significance of the performance variations among classifiers, given the constraints of the current research environment and experimental design. This issue is on our agenda to be tackled in subsequent research. We firmly believe that with the continuous deepening of research and the expansion of exploration, network intrusion detection technology based on diffusion models will play an increasingly critical role in strengthening cybersecurity defenses, optimizing detection processes, and enhancing recognition accuracy.

Acknowledgement: The authors would like to thank the National Cybersecurity College Student Innovation Funding Program and Beijing Topsec Network Security Technology Co., Ltd., for funding this research. Also, the authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.

Funding Statement: This work was supported by the Key Research and Development Program of Hainan Province (Grant Nos. ZDYF2024GXJS014, ZDYF2023GXJS163), the National Natural Science Foundation of China (NSFC) (Grant Nos. 62162022, 62162024), Collaborative Innovation Project of Hainan University (XTCX2022XXB02).

Author Contributions: The authors confirm their contribution to the paper as follows: Yue Yang proposed the main ideas and principles of this research, designed and implemented part of the algorithms and experimental schemes, and wrote the paper. Xiangyan Tang guided the design of the algorithms and experiments. Zhaowu Liu verified the dataset for the model algorithm and visualized the results. Jieren Cheng controlled the overall design and quality of the paper. Haozhe Fang completed the comparative experiments related to the SOTA model. Cunyi Zhang drew the charts for the paper, adjusted the references, and formatted the paper. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at https://github.com/Jehuty4949/NSL_KDD (accessed on 22 December 2024), https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 22 December 2024), http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 22 December 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Qi L, Yang Y, Zhou X, Rafique W, Ma J. Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure Industry 4.0. IEEE Trans Ind Informat. 2022 Sep;18(9):6503–11.
2. Xu C, Shen J, Du X. A method of few-shot network intrusion detection based on meta-learning framework. IEEE Trans Inf Forensics Secur. 2020;15:3540–52.
3. Papamartzivanos D, Gómez Mármol F, Kambourakis G. *Dendron*: genetic trees driven rule induction for network intrusion detection systems. Future Gener Comput Syst. 2018 Feb;79:558–74. doi:10.1016/j.future.2017.09.056.
4. Abdulhammed R, Musafer H, Alessa A, Faezipour M, Abuzneid A. Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics. 2019 Mar;8(3):322.
5. Mendonça RV, Teodoro AAM, Rosa RL, Saadi M, Melgarejo DC, Nardelli PHJ, et al. Intrusion detection system based on fast hierarchical deep convolutional neural network. IEEE Access. 2021;9:61024–34.

6. Atefinia R, Ahmadi M. Network intrusion detection using multiarchitectural modular deep neural network. J Supercomput. 2021 Apr;77(4):3571–93. doi:10.1007/s11227-020-03410-y.
7. Yang J, Chen X, Chen S, Jiang X, Tan X. Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection. IEEE Trans Inf Forensic Secur. 2021;16:3538–53. doi:10.1109/TIFS.2021.3083422.
8. Hu X, Gao W, Cheng G, Li R, Zhou Y, Wu H. Towards early and accurate network intrusion detection using graph embedding. IEEE Trans Inf Forensic Secur. 2023;18:5817–31. doi:10.1109/TIFS.2023.3318960.
9. Ravi V, Chaganti R, Alazab M. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. Comput Electr Eng. 2022;102(3):108156. doi:10.1016/j.compeleceng.2022.108156.
10. Sajid Farooq M, Abbas S, Atta-ur-Rahman, Sultan K, Adnan Khan M, Mosavi A. A fused machine learning approach for intrusion detection system. Comput Mater Contin. 2022;74(2):2607–23. doi:10.32604/cmc.2023.032617.
11. Zhang J, Luo C, Carpenter M, Min G. Federated learning for distributed IIoT intrusion detection using transfer approaches. IEEE Trans Ind Informat. 2022;19(7):8159–69.
12. Sezgin A, Boyaį A. AID4I: an intrusion detection framework for industrial internet of things using automated machine learning. Comput Mater Contin. 2023;76(2):2121–43. doi:10.32604/cmc.2023.040287.
13. Wang T, Fang K, Wei W, Tian J, Pan Y, Li J. Microcontroller unit chip temperature fingerprint informed machine learning for IIoT intrusion detection. IEEE Trans Ind Informat. 2022;19(2):2219–27.
14. Li J, Tong X, Liu J, Cheng L. An efficient federated learning system for network intrusion detection. IEEE Syst J. 2023;17(2):2455–64.
15. Thockchom N, Singh MM, Nandi U. A novel ensemble learning-based model for network intrusion detection. Complex Intell Syst. 2023;9(5):5693–714. doi:10.1007/s40747-023-01013-7.
16. Wu Y, Yang L, Zhang L, Nie L, Zheng L. Intrusion detection for unmanned aerial vehicles security: a tiny machine learning model. IEEE Internet Things J. 2024 Jun 15;11(12):20970–82.
17. Li Z, Yao W. A two stage lightweight approach for intrusion detection in Internet of Things. Expert Syst Appl. 2024;257:124965.
18. Chitrakar R, Huang C. Selection of candidate support vectors in incremental SVM for network intrusion detection. Comput Secur. 2014;45:231–41. doi:10.1016/j.cose.2014.06.006.
19. Sun M, Lai Y, Wang Y, Liu J, Mao B, Gu H. Intrusion detection system based on in-depth understandings of industrial control logic. IEEE Trans Ind Informat. 2022;19(3):2295–306.
20. Wang M, Yang N, Weng N. K-GetNID: knowledge-guided graphs for early and transferable network intrusion detection. IEEE Trans Inf Forensic Secur. 2024;19:7147–60.
21. Chen Y, Su S, Yu D, He H, Wang X, Ma Y, Guo H. Cross-domain industrial intrusion detection deep model trained with imbalanced data. IEEE Internet Things J. 2022;10(1):584–96. doi:10.1109/JIOT.2022.3201888.
22. Gong T, Zhu L, Yu FR, Tang T. Train-to-edge cooperative intelligence for obstacle intrusion detection in rail transit. IEEE Trans Veh Technol. 2024 Jun;73(6):7669–80.
23. Li Z, Huang H, Deng S, Qiu W, Gao X. A soft actor-critic reinforcement learning algorithm for network intrusion detection. Comput Secur. 2023;135(3):103502. doi:10.1016/j.cose.2023.103502.
24. Wu Y, Hu Y, Wang J, Feng M, Dong A, Yang Y. An active learning framework using deep Q-network for zero-day attack detection. Comput Secur. 2024;139(4):103713. doi:10.1016/j.cose.2024.103713.
25. Thakkar A, Lohiya R. Attack classification of imbalanced intrusion data for IoT network using ensemble-learning-based deep neural network. IEEE Internet Things J. 2023;10(13):11888–95. doi:10.1109/JIOT.2023.3244810.
26. Ye J, Lv J, Xu G, Liu T. Leaky cable perimeter intrusion detection based on deep reinforcement learning. IEEE Internet Things J. 2024 Jun 15;11(12):22616–27.
27. Shahriar MH, Xiao Y, Moriano P, Lou W, Hou YT. CANShield: deep learning-based intrusion detection framework for controller area networks at the signal-level. IEEE Internet Things J. 2023 Dec 15;10(24):22111–27. doi:10.1109/JIOT.2023.3303271.

28. Greidanus MDR, Gupta S, Sur D, Mazumder SK. Diffusion of radiated side-channel noise intrusion in PLL-dependent cascaded solid-state transformers. IEEE Trans Power Electron. 2024 Nov;39(11):14148–54. doi:10.1109/TPEL.2024.3437233.
29. Wang Y, Ding J, He X, Wei Q, Yuan S, Zhang J. Intrusion detection method based on denoising diffusion probabilistic models for UAV networks. Mobile Networks Appl. 2023;1–10. doi:10.1007/s11036-023-02222-7.
30. Krishnasamy B, Muthaiah L, Kamali Pushparaj JE, Pandey PS. DIWGAN optimized with Namib beetle optimization algorithm for intrusion detection in mobile ad hoc networks. IETE J Res. 2024;70(5):4422–41. doi:10.1080/03772063.2023.2223181.
31. Wang Z, Xu Z, He D, Chan S. Deep logarithmic neural network for Internet intrusion detection. Soft Comput. 2021;25(15):10129–52. doi:10.1007/s00500-021-05987-9.
32. Wang Z, Liu Y, He D, Chan S. Intrusion detection methods based on integrated deep learning model. Comput Secur. 2021;103(6):102177. doi:10.1016/j.cose.2021.102177.
33. Sahu SK, Mohapatra DP, Rout JK, Sahoo KS, Pham QV, Dao NN. A LSTM-FCNN based multi-class intrusion detection using scalable framework. Comput Electr Eng. 2022;99:107720. doi:10.1016/j.compeleceng.2022.107720.
34. Wang Z, Li Z, Wang J, Li D. Network intrusion detection model based on improved BYOL self-supervised learning. Secur Commun Netw. 2021;2021:1–23. doi:10.1155/2021/8690662.
35. Wang Z, Li Z, He D, Chan S. A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning. Expert Syst Appl. 2022;206(4):117671. doi:10.1016/j.eswa.2022.117671.
36. Abdel-Basset M, Hawash H, Chakrabortty RK, Ryan MJ. Semisupervised spatiotemporal deep learning for intrusions detection in IoT networks. IEEE Internet Things J. 2021;8(15):12251–65. doi:10.1109/JIOT.2021.3060878.
37. Gu J, Lu S. An effective intrusion detection approach using SVM with naïve Bayes feature embedding. Comput Secur. 2021;103:102158.
38. Farahani G. Feature selection based on cross-correlation for the intrusion detection system. Secur Commun Netw. 2020;2020(6):1–17. doi:10.1155/2020/8875404.
39. Sharma NV, Yadav NS. An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers. Microprocess Microsyst. 2021;85(3):104293. doi:10.1016/j.micpro.2021.104293.