Mathematical Biosciences and Engineering

MBE, 19(11): 11512–11532.
DOI: 10.3934/mbe.2022536
Received: 28 June 2022
Revised: 02 August 2022
Accepted: 05 August 2022
Published: 11 August 2022

http://www.aimspress.com/journal/MBE

Research article

Research on mobile traffic data augmentation methods based on SA-ACGAN-GN

Xingyu Gong[1], Ling Jia[1,2,*] and Na Li[1]

[1] College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China
[2] Hanzhong Vocational School of Science and Technology, Hanzhong 723200, China

* **Correspondence:** Email: jialing@stu.xust.edu.cn.

Abstract: With the rapid development and application of the mobile Internet, it is necessary to analyze and classify mobile traffic to meet the needs of users. Due to the difficulty in collecting some application data, the mobile traffic data presents a long-tailed distribution, resulting in a decrease in classification accuracy. In addition, the original GAN is difficult to train, and it is prone to "mode collapse". Therefore, this paper introduces the self-attention mechanism and gradient normalization into the auxiliary classifier generative adversarial network to form SA-ACGAN-GN model to solve the long-tailed distribution and training stability problems of mobile traffic data. This method firstly converts the traffic into images; secondly, to improve the quality of the generated images, the self-attention mechanism is introduced into the ACGAN model to obtain the global geometric features of the images; finally, the gradient normalization strategy is added to SA-ACGAN to further improve the data augmentation effect and improve the training stability. It can be seen from the cross-validation experimental data that, on the basis of using the same classifier, the SA-ACGAN-GN algorithm proposed in this paper, compared with other comparison algorithms, has the best precision reaching 93.8%; after adding gradient normalization, during the training process of the model, the classification loss decreases rapidly and the loss curve fluctuates less, indicating that the method proposed in this paper can not only effectively improve the long-tail problem of the dataset, but also enhance the stability of the model training.

Keywords: mobile traffic classification; long-tailed distribution; generative adversarial network; self-attention mechanism; gradient normalization

1. Introduction

With the rapid development of mobile Internet technology and the widespread popularization of smart devices, various mobile applications have not only changed people's work and lifestyle, but also brought problems such as user data security and privacy leakage. The research has received extensive attention from scholars. In order to help operators achieve more targeted network management and control, better user experience, and to help security regulators improve the security level of networks and applications, it is necessary to effectively analyze and accurately identify mobile traffic.

Compared with the PC-side traffic classification task, the random port number technology [1], encrypted traffic [2], cloud services and Content Distribution Network [3] widely used by mobile clients all make the traditional methods based on transport layer protocols, port numbers and deep packet inspection no longer suitable for mobile traffic identification. Therefore, the mobile traffic identification method based on Deep Learning (DL) came into being. Li et al. [4] uses convolutional autoencoders to learn a large number of unlabeled samples, and establishes a mapping from hidden layer features to application types through multi-type regression. Aceto et al. [5] lists some research works of DL technology in the field of mobile traffic identification, and tests and analyzes these related works. After that, the above-mentioned researchers conducted research on multimodal multitask deep learning methods, and successively proposed the multimodal deep learning architecture MIMETIC [6] and the DISTILLER classifier [7] to overcome the traffic classification based on single-modality deep learning performance limitations. Liu et al. [8] proposed a deep learning-based Flow Sequence Network (FS-Net) to learn representative features from the raw flows, and then classifies them in a unified framework. Nascita et al. [9] improves the multimodal deep learning architecture MIMETIC [6] by combining explainable artificial intelligence (XAI) technology, which better achieves the classification of mobile traffic.

However, in the process of mobile traffic classification, due to the large quantitative difference between various types of application traffic data, there is a long-tailed distribution with extremely unbalanced proportions of samples from the majority class and minority class, which leads to the final classification result being biased towards the majority class traffic and ignoring the minority class traffic, so that the classification model loses its ability to identify the minority class traffic, so even if a higher accuracy rate is obtained, it will mislead the researchers and even cause a higher cost [10,11]. In addition, there is a common problem in multi-layer deep neural networks [12], the error gradient is continuously accumulated or exponentially decreasing in the update, resulting in the problem of gradient explosion or gradient disappearance, and thus the network becomes unstable, which may completely stop further training of the neural network, and cannot continue to learn the target result. Therefore, researching data augmentation methods in mobile traffic classification to overcome the limitations brought by long-tail data and enhance the stability of model training has become a research hotspot in this field.

In view of the above problems, this paper proposes a mobile traffic data enhancement algorithm based on SA-ACGAN-GN. First, in order to give full play to the excellent performance of the Generative Adversarial Network (GAN) in image data enhancement, the traffic is converted into a single-channel grayscale image; secondly, in order to solve the problem of low quality of some generated images, the self-attention mechanism is introduced in both the generator and the discriminator of ACGAN, so that the network can quickly and accurately locate the key attention regions in the image, improving the richness of the details of the generated image, thereby better

balancing the dataset to improve the classification performance; finally, by introducing gradient normalization [13] strategy, accelerate the convergence of the model, stabilize the training process, and continuously improve the classification effect. The experimental results show that using the SA-ACGAN-GN algorithm proposed in this paper to enhance the data can improve the long-tailed distribution problem in the original dataset and significantly improve the accuracy of mobile traffic classification. At the same time, adding the gradient normalization strategy improves the algorithm training stability.

The rest of this paper is organized as follows. Section 2 introduces related works addressing the problem of long-tailed distributions. Section 3 describes the data augmentation algorithm based on SA-ACGAN-GN proposed in the paper. The experimental implementation details and results are provided and discussed in Section 4. Section 5 concludes the research work of the paper and illustrates possible future developments.

2. Related work

2.1. Long tail classification technology

Most of the traditional solutions to deal with long-tailed data are by introducing some imbalanced data learning techniques to resample the data before the learning reaches the equilibrium condition. The sampling method includes two different ways of under-sampling and over-sampling, that is, under-sampling the majority class samples and over-sampling the minority class samples. Random under-sampling (RUS) and random over-sampling (ROS) are the two main methods of under-sampling and over-sampling [14]. However, RUS may discard potentially useful data, and ROS may overfit the model by repeatedly learning the same sampled data. To limit overfitting, various data augmentation methods such as SMOTE [15], ADASYN [16], RFMSE [17] are formulated to generate artificial, similar samples. However, these methods usually ignore the distribution of samples of the majority class, resulting in overlapping data.

In recent years, the rapid development of deep learning technology has made it widely used in the field of data balancing. Hasibi et al. [18] proposed a novel data augmentation method based on using Long Short Term Memory networks to learn network traffic patterns for replicating minority class traffic packet sequences. The results show that the dataset enhanced by this method achieves better classification results. Arefeen et al. [19] used Artificial Neural Network to fit the minority class data, predicted the majority class data, and selected the data with larger prediction error for training.

In addition, many researchers have used advanced boosting technology to process long-tail data recently, and combined the Deep Neural Network (DNN) architecture with it to improve the classification effect by taking advantage of the good generalization performance of advanced boosting technology. Folino et al. [20] adopted an ad-hoc shared DNN architecture, featuring a combination of dropout capabilities, skip connections, along with a cost-sensitive loss to efficiently learn deep base classifiers from minority class samples，which is also the first attempt to combine a block-based learning scheme with DNN ensemble techniques to handle long-tail classification tasks, and the experimental results also confirm the feasibility and application prospects of the method. Bedi et al.[21] proposed an improved Siam-IDS (I-SiamIDS) algorithm-level method, using a collection of binary eXtreme Gradient Boosting, Siamese Neural Network and Deep Neural Network to improve the system's effectiveness in detecting intrusion attacks in an unbalanced network environment, but its

computational time overhead is very large. Gupta [22] et al. proposed a three-layer CSE-IDS based on cost-sensitive deep learning and ensemble algorithms to solve the long-tail problem in intrusion detection systems.

As Generative adversarial network [23] have achieved good results in data generation fields such as audio generation [24], image generation [25], style transfer [26], this DL technique has also begun to be applied to address the data augmentation problem in network traffic classification, researchers use a GAN-based method to generate traffic samples to overcome the impact of long-tail data. Lee et al. [11] used GAN for the augmentation of network traffic data, and then used Random Forest to train on the augmented training set, and the F1-Score was 95.04%. Zhang et al. [27] used GANs to address data scarcity and long-tail issues in intrusion detection systems. Park et al. [28] used TGAN to expand the minority class sample flow, borrowed the idea of congestion avoidance, and performed fitting avoidance in advance, and its F1-Score was 93.98%. Vu et al. [29] used ACGAN to generate data to balance the dataset for better results in network traffic classification. However, due to the limited size of the convolution kernel, only the relationship between the local regions of the sample can be learned, and details may be lost. Douzas [30] applied CGAN to imbalanced datasets by adding additional conditional information to the GAN to learn the true data distribution based on the global information of the minority class data, generating synthetic data for the minority class, but did not address the training instability question.

A summary of the related work on the above-mentioned long-tail classification technology is shown in Table 1.

Table 1. Summary of work related to long-tail classification technology.

Research Direction	Literature	Algorithm or Model
based on traditional sampling	[14]	RUS、ROS
	[15]	SMOET
	[16]	ADASYN
	[17]	M-SMOTE+ENN
based on deep learning	[18]	LSTM
	[19]	ANN
based on advanced boosting and DNN	[20]	Ensemble_moe
	[21]	I-SiamIDS
	[22]	CSE-IDS
based on GAN	[23–26]	GAN
	[27]	styleGAN
	[28]	TGAN
	[29]	ACGAN
	[30]	CGAN

In summary, although domestic and foreign researchers have proposed many long-tail classification algorithms and achieved certain results, there are still some unsolved problems. In the literature [14–17], the method based on traditional sampling is less difficult to implement and has high versatility, but it is essentially repeated learning for minority class samples, and there is a problem of easy overfitting. The deep learning-based techniques in the literature [18,19] reconstruct the data well,

but do not introduce new features for the data, which is also the biggest difference from GAN. In the literature [20–22], the method based on advanced boosting and DNN introduces the idea of modularization, and the scope of changes to the model structure is small, but because it is a combination of multiple models, it often increases the computational cost exponentially. In the literature [23–30], the method based on GAN introduces new features while adding minority class samples, which effectively improves the classification effect. However, due to the low quality of some data generation, the improvement of the classification effect is limited, so this paper will conduct more in-depth research based on this direction in the future.

2.2. GAN's training stability

As a new type of deep learning model, GAN can generate completely new data samples through adversarial learning. It provides a new method to solve the long-tail problem of raw data. However, GAN is prone to instability and gradient disappearance during training. Therefore, researchers proposed Wasserstein GAN (WGAN) [31] to improve related problems. WGAN uses Wasserstein distance to replace JS divergence, and uses weight clipping to approximate the l-Lipschitz constraint, but in fact these two conditions are not equivalent, so the effect is not very ideal. Gulrajani et al. [32] showed in subsequent work that gradient penalty (GP) is more effective. Based on previous work on GP and weight clipping, Miyato et al. [33] proposed Spectral Normalization (SN) to normalize the weight matrix in the discriminator to the corresponding spectral norm, effectively controlling the Lipschitz constant of the discriminator makes the parameter change more stable during the training process and is less prone to gradient explosion.

There are also many applications and developments for the above improved ideas for GAN's training stability. Ding et al. [34] designed a tabular auxiliary classifier generative adversarial networks model (TACGAN) for balancing normal samples and attack samples in intrusion detection. A new regularization function is designed in the generator of the TACGAN model, so that the generated minority class data is more in line with the sample distribution of the original data. Zhang et al. [35] used Conditional Wasserstein GAN-Gradient Penalty (CWGAN-GP) to generate synthetic samples for minority classes in imbalanced datasets. CWGAN-GP not only generates more realistic data, but also overcomes the problems of mode collapse and training instability. Xu et al. [36] introduced spectral normalization into GAN-based medical image super-resolution to control the performance of the discriminator.

A summary of the related work on the above-mentioned GAN's training stability is shown in Table 2.

Table 2. Summary of work related to GAN's training stability.

Research Direction	Literature	Algorithm or Model
based on regularization	[31]	WGAN
	[32]	WGAN-GP
	[34]	TACGAN
	[35]	CWGAN-GP
based on normalization	[33][36]	SN-GAN

It can be seen that the work on solving the problem of GAN's training stability has mainly focused

on two methods: regularization and normalization. By regularizing or normalizing the discriminator, the gradient space of the discriminator becomes smoother and the stability is improved. However, compared with model-level constraints, spectral normalization methods are module-level constraints, and their layer capacity is limited, which reduces the learning ability of neural networks [13]. The gradient penalty method is soft-constrained, the tightness of the constraints fluctuates due to regularization, the upper limit is not limited, and the performance is not as good as the hard constraints. Therefore, which method can better stabilize GAN training and further improve the generation performance of GAN is still a concern of researchers.

3. Mobile traffic data augmentation algorithm based on SA-ACGAN-GN

3.1. Mobile traffic classification workflow based on SA-ACGAN-GN data augmentation algorithm

The SA-ACGAN-GN algorithm proposed in this paper is used to classify and identify mobile traffic after processing the long-tail dataset. The overall workflow is shown in Figure 1. It is mainly divided into three modules: data processing, data augmentation and traffic classification. The data processing module introduces a genetic imaging coding scheme, combines mobile traffic classification with image recognition to processes mobile traffic, and converts them into single-channel grayscale images. In the data augmentation module, an ACGAN algorithm combined with the self-attention mechanism is used to process the minority class samples in the mobile traffic, and the gradient normalization strategy is introduced at the same time. The enhanced dataset is sent to the traffic classification module, and the traffic classification and identification are carried out in the constructed CNN classifier.

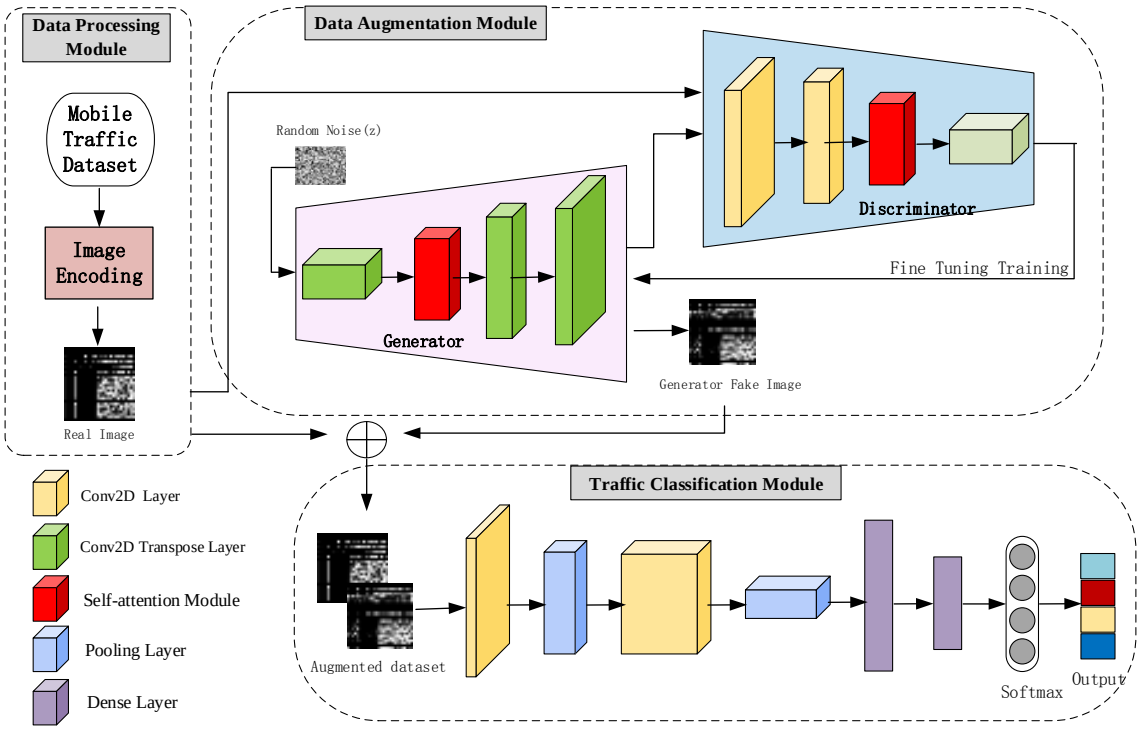


Figure 1. Mobile traffic classification workflow.

3.2. Data processing

This paper uses the public mobile traffic dataset MIRAGE-2019 for experiments, which was created by Giuseppe Aceto et al. in 2019. Among them, the overall characteristics of the data flow include the statistical characteristics such as packet length, number of packets, and interval time, which is, 17 statistical eigenvalues including mean, extreme value, median, etc. Considering the bidirectionality of the data packet sequence, there are a total of 102 features used to characterize mobile application traffic. Using a genetic coding scheme, the above features are converted into feature grayscale matrices, and the feature matrices of different mobile applications are presented in the form of grayscale images. The matrix size is defined as 28 × 28 when instantiating the transformation algorithm for subsequent experimental modeling. The specific implementation process is divided into the following four steps:

1) Determine the matrix T of size N × M, where N represents the number of training samples, M represents the traffic characteristics;

2) Apply the t-SNE algorithm to calculate the probability distribution $p_{j|i}$ of the feature x_j relative to x_i in the high-dimensional space, see Eq (1).

$$p_{j|i} = \frac{exp(-\left||x_i - x_j\right||^2/2\sigma^2)}{\sum_{k\neq i} exp(-\left||x_i - x_j\right||^2/2\sigma^2)} \tag{1}$$

where σ is the variance of the Gaussian function centered on x_j.

For the mappings y_i and y_j of x_i and x_j in the low-dimensional space, the t-distribution with 1 degree of freedom is used to represent the low-dimensional joint probability distribution, see Eq (2).

$$q_{j|i} = \frac{(1+\left||y_i - y_j\right||^2)^{-1}}{\sum_{k\neq i} (1+\left||y_i - y_j\right||^2)^{-1}} \tag{2}$$

In order to ensure that the joint probability distribution $q_{j|i}$ between the mapping points in the low-dimensional space can better simulate the joint probability distribution $p_{j|i}$ between the data points in the high-dimensional space, the gradient descent is used to minimize the KL divergence of all data points to obtain the best simulated point in the low-dimensional space. After the calculation of the t-SNE algorithm is completed, the matrix T (N × M) is transformed into the matrix tsne(T)(N × 2), and the columns of tsne(T) are the two-dimensional coordinates of the feature position tsne(T)[i,1] and tsne(T)[i,2].

3) Use the Convex Hull Algorithm to find the smallest rectangle covering all points associated by t-SNE transformation, frame the feature points to the two-dimensional Cartesian plane, and calculate the minimum and maximum coordinates of the feature points along the two-dimensional Cartesian axis x and axis y to get a rectangle with length max(x)-min(x) and width max(y)-min(y).

4) Assign pixels one-to-one to each feature to generate an image.

3.3. Model structure and algorithm implementation based on SA-ACGAN-GN

The generator in the SA-ACGAN-GN model generates a batch of sample data mixed with real samples and sends them to the discriminator by establishing the mapping relationship between the

noise sample distribution and the real sample distribution. The discriminator performs true and false discrimination and back-propagates the gradient information to update the parameters of the model. The discriminator and generator are updated alternately until a Nash equilibrium is reached. The data augmentation model based on SA-ACGAN-GN is shown in Figure 2.

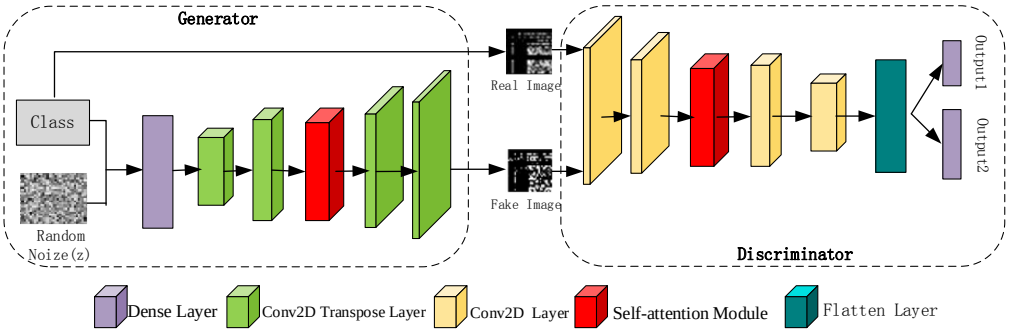


Figure 2. Data augmentation model based on SA-ACGAN-GN.

This paper uses two distinct loss functions: 1) the binary cross-entropy function used to train the discriminator and estimate the true probability of the input image; 2) the categorical cross-entropy function for predicting the class labels of the images, respectively. Its expressions are shown in Eqs (3) and (4).

$$L_s = E[\log P(S = real \mid X_{real})] + E\left[\log P\left(S = fake \mid X_{fake}\right)\right] \tag{3}$$

$$L_c = E[\log P(C = c \mid X_{real})] + E\left[\log P\left(C = c \mid X_{fake}\right)\right] \tag{4}$$

Here, L_S represents the probability that the data is true, L_C represents the probability that the data is correctly classified, c represents the class label, X_{real} represents the real data, and X_{fake} represents the generated data. The training objective of the discriminator is maximize ($L_S + L_C$), that is, to achieve the best results in sample classification and authenticity discrimination; the training objective of the generator is maximize ($L_C - L_S$), that is, the generated samples are as much as possible to deceive the generation and match the given label.

3.3.1. The structure of generator

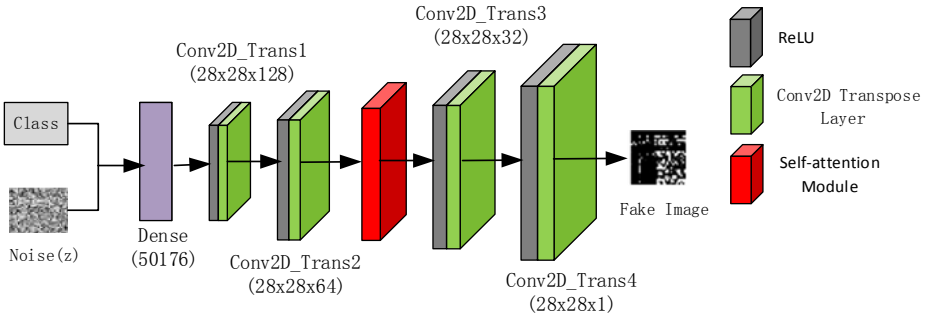


Figure 3. The structure of SA-ACGAN-GN generator.

The structure of the SA-ACGAN-GN generator is shown in Figure 3.

1) The first layer is the input layer. The generator splices the input 100-dimensional random noise z and 10 application class labels into a noise vector and sends it to the Dense layer for sampling.

2) The second layer is a Dense layer with 50176 parameters, responsible for dimensional transformation of the input, and its output dimension is (b, 50176), where b is the number of batch samples. The output dimension of the Dense layer after reshape reset is (b, 14, 14, 256).

3) The third and fourth layers are two transposed convolution layers with similar structures. The size of the convolution kernel is 5 × 5, the stride is 2, and there are 128 and 64 convolution kernels respectively. The output dimensions are (b, 28, 28, 128) and (b, 28, 28, 64), the expression of the transposed convolution operation is shown in Eq (5):

$$G_i = ReLU\ (G_{i-1} *\ T_i) \tag{5}$$

where G_i is the output of the i-th layer, and ReLU is used as the activation function for each layer of the generator.

4) In order to strengthen the model's ability to associate long-distance pixels and model feature content, and improve the quality of model generation, a self-attention module is added after the fourth layer. The specific principle of the module is shown in Figure 4.

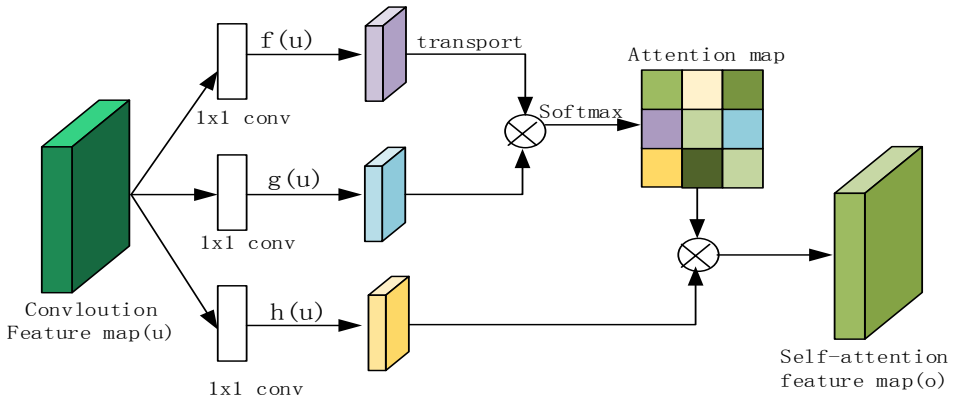


Figure 4. Self-attention module structure.

The input of the self-attention module is the image feature map $u \in G_4^{CxN}$ extracted by the fourth layer , where C is the number of channels and N=H x W is the number of pixels. It is divided into three branches f(u), g(u), h(u), which represent the three feature spaces obtained by multiplying image features by different weight matrices. Transpose the output of f(u) and multiply it with the output of g(u), ⊗ stands for matrix multiplication. The attention map β_{ji} is then normalized by the classifier Softmax, and its calculation expression is shown in Eq (6):

$$\begin{aligned} f(\mathrm{u}) &= W_f u \\ g(\mathrm{u}) &= W_g u \\ h(\mathrm{u}) &= W_h u \\ b_{ij} &= f^T(\mathrm{u}) \otimes \mathrm{g}(\mathrm{u}) \\ \beta_{ji} &= \frac{\exp(\ b_{ij})}{\sum_{i=1}^{N} \exp(\ b_{ij})} \end{aligned} \tag{6}$$

In Eq (6), b_{ij} is the correlation matrix, β_{ji} represents the degree of attention to the i-th position in the j-th area.

After performing matrix multiplication of the attention map β_{ji} and the feature space $h(u_i)$, the expression of the self-attention feature map is obtained in Eq (7):

$$O_i = \sum_{i=1}^{N} \beta_{ji} h(u_i) \tag{7}$$

The output of the attention layer is further multiplied by the parameters and added back to the input feature map. The final output s_i is shown in Eq (8), where r is the transition parameter and the initial value is 0, so that the model learns from domain information and gradually assigns weights to other distant feature details:

$$s_i = rO_i + u_i \tag{8}$$

5) After obtaining a self-attention feature map with a size of 784 × 64, the sixth layer performs a transposed convolution operation similar to the previous one, with a stride of 1, and 32 convolution kernels. The output dimension is (b, 28, 28, 32), and the calculation expression is shown in Eq (5).

6) The seventh layer is still a transposed convolution layer, with a stride of 1, only one convolution kernel, and its output dimension is (b, 28, 28, 1), and the activation function adopts the sigmoid function. Its output is the new sample generated by the generator, and the calculation expression is shown in Eq (9):

$$output = sigmoid\ \ (G_6 *\ \ T_7) \tag{9}$$

3.3.2. The structure of discriminator

The structure of the discriminator is symmetrical with the generator, as shown in Figure 5.

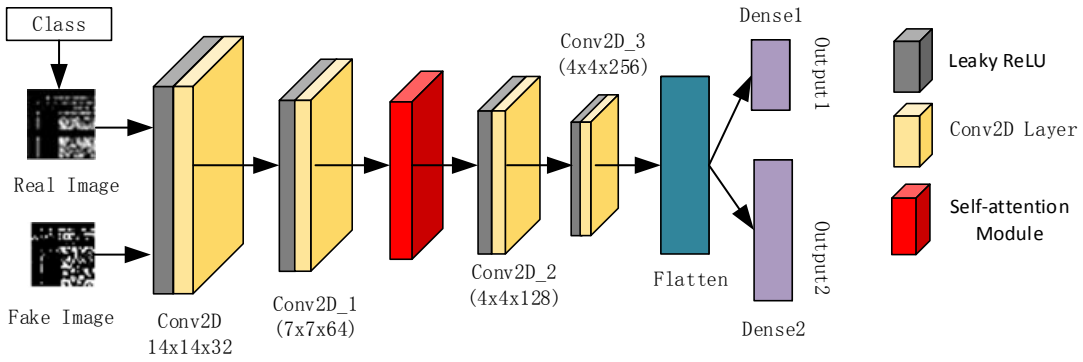


Figure 5. The structure of SA-ACGAN-GN discriminator.

1) The first layer is the input layer, the input data consists of real data and data generated by the generator, and the image data format is (b, 28, 28, 1).

2) The second and third layers are convolutional layers with similar structures. The size of the convolution kernel is 5x5, the number of convolution kernels is 32 and 64, and the stride is set to 2. After two feature extractions of the convolutional layer, the size of the image channel is continuously

increased, and the size of the image is gradually reduced, thereby reducing the amount of calculation and the occupation of memory. The output dimensions are (b, 14, 14, 32) and (b, 7, 7, 64), the calculation expression is shown in Eq (10).

$$D_i = LeakyReLU\ (D_{i-1} *\ C_i) \tag{10}$$

Among them, D_i is the output of the i-th layer, and the LeakyReLU function is used as the activation function of each layer of the discriminator.

3) The fourth layer adds the same self-attention module as the generator, and through a series of calculations in Eq (6), a self-attention feature map with a size of 49 × 64 is obtained to enrich feature details.

4) The fifth and sixth layers are convolutional layers similar in structure to the second and third layers. The size of the convolution kernel is 5 × 5, the number of convolution kernels is 128 and 256 respectively, and the stride of the fifth layer is 2, the stride of the sixth layers is 1. After two convolutions, the output dimensions are (b, 4, 4, 128) and (b, 4, 4, 256) respectively, and the calculation expression is shown in Eq (10).

5) Finally, the output of the sixth layer is sent to the Dense layer through the seventh layer (Flatten layer) to calculate the authenticity of the input sample and the probability that the sample belongs to a certain category.

3.4. Gradient normalization

In order to stabilize the training process of SA-ACGAN-GN, this paper introduces a gradient normalization method, which enables model-level constraints and hard constraints to be realized at the same time.

Considering the generative network model as a multi-layer neural network, the input-output relationship of the k-th layer can be expressed as:

$$f_k(x) = \emptyset_k(W_k \cdot f_{k-1}(x) + b_k) \tag{11}$$

Among them, $\emptyset_k$ is the nonlinear activation function of the network in this layer, and the ReLU activation function can be used, W_k is the network parameter matrix, and b_k is the bias of the network.

The Lipschitz constraint is a requirement for the gradient of f(x), which is defined as the following regularization function:

$$\hat{f}(x) = \frac{f(x)}{||\nabla_x f(x)|| + f(x)} \quad \in [-1, 1] \tag{12}$$

The norm of its gradient is bounded and satisfies the 1-Lipschitz constraint. Its expression is shown in Eq (13):

$$||\nabla_x \hat{f}(x)|| = \left| \frac{f(x)}{||\nabla_x f(x)|| + |f(x)|} \right|^2 \leq 1 \tag{13}$$

It can be seen that normalization can be achieved by directly constraining the gradient norm to satisfy the constraint of Lipschitz = l. Therefore, this paper makes the training more stable by introducing gradient normalization in the generator and discriminator networks.

4. Experiments and discussion

4.1. Experimental environment

The experimental environment used in this paper is windows 10 operating system, the CPU is Intel(R) Core(TM)i7-10875H, the GPU is NVDIA GeForce RTX 3060, the memory is 16 GB, and Tensorflow is used as the backend support.

4.2. Dataset partitioning and data preprocessing

This paper selects 10 kinds of traffic from the MIRAGE-2019 dataset to construct the dataset used in the experiments in this section. The types and quantities of these 10 kinds of application traffic are shown in Table 3, with a total of 35,205 pieces of data (this dataset will be referred to as O later). It can be seen from the constructed dataset that there is a long-tailed distribution problem in the experimental dataset, in which the proportion of the Motain type in the dataset O reaches 34.81%, while the Hypah type only accounts for 3.02%. In order to verify the data enhancement effect of each balancing algorithm, the above experimental dataset O is divided into 5 mutually exclusive subsets of similar size according to the unbalanced ratio in the experiment (this subset is called O-Sub$_i$ later, and the subscript i is 1–5 to indicate 5 subsets), the number of data of each application type in each subset is shown in Table 3, each time 4 subsets are selected to form an unbalanced dataset to conduct cross-validation experiments for each balanced algorithm, and this part of the cross-validation needs to be performed 5 times. Each cross-validation generates a balanced data subset (subsequently this dataset is called B-Sub$_i$, and the subscript i is 1-5 to indicate 5 subsets). This data subset contains 10 application types of data. A total of 2000 traffic data per application, and a balanced data subset has a total of 20,000 traffic data. 5 times of cross-validation will generate 5 balanced data subsets, a total of 100,000 traffic data, these data are used as the experimental dataset for the final CNN classification model.

Table 3. Number of applications in the MIRAGE dataset.

APP	Dataset O	O-Sub$_i$	Proportion
Dropbox	1162	232	3.30%
Duolingo	1646	329	4.67%
Facebook	1468	294	4.17%
Hypah	1064	213	3.02%
Iconology	8279	1656	23.51%
Joelapenna	2261	452	6.42%
Motain	12,257	2451	34.81%
Subito	4346	869	12.34%
Tripadvisor	1497	299	4.25%
Youtube	1228	246	3.49%
Total	35,205	7041	100.00%

According to the coding scheme described in Section 3.2, the mobile traffic data is preprocessed, and the resulting grayscale traffic map is shown in Figure 6. It can be seen that the characteristics of the flow image of each application, such as the color of each pixel, the overall outline, etc., are different.

Therefore, this traffic image can be classified according to its characteristics.

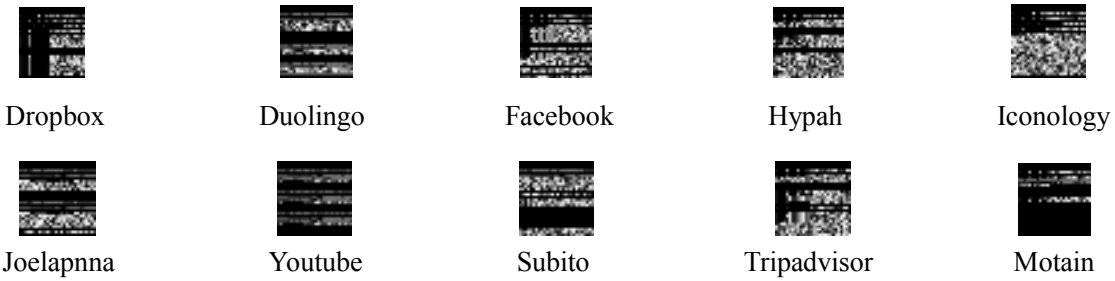


Figure 6. Conversion results of single-channel grayscale traffic image dataset.

4.3. Experimental settings

1) Parameter setting: The training parameters used in the SA-ACGAN-GN model proposed in this paper are shown in Table 4.

Table 4. SA-ACGAN-GN training parameter settings.

Training Parameters	Generator	Discriminator
Optimizer	RMSprop	RMSprop
Learning rate	0.0001	0.0002
Epoches	4000	4000
Mini_batch	64	64

2) Performance indicators: The performance indicators used to evaluate mobile traffic classification are Precision, Recall, F1-Scroe and Confusion Matrix; the performance indicators used to evaluate training stability are Accuracy and Loss.

3) Baseline algorithm: The paper adopts 4 methods to balance the long-tailed dataset for comparison purpose. The 1st and 2nd are ROS and SMOTE mentioned in Section 2, respectively. The 3rd is the variational auto-encoder (VAE). The 4th is the original ACGAN.

4) Experimental procedure:

Step1: Preprocess the mobile traffic data, and convert the one-dimensional original feature vector into a two-dimensional image form.

Step2: The dataset O is divided into 5 O-Sub_i (i is 1–5), and ROS, SMOTE, VAE, ACGAN, SA-ACGAN and this paper's algorithm are used to generate minority class sample data to balance the dataset. Each crossover experiment generates a balanced data subset B-Sub_i (i is 1–5). B-Sub_i contains 10 application types of data, and the number of each application is balanced to 2000, a total of 20000 pieces of traffic data, and the final 100,000 pieces of data are used as the experimental dataset for the CNN classification model.

Step3: Convolutional Neural Networks (CNN) are used as classifiers. In order to reduce the influence of the instability of the classifier, 100,000 pieces of data obtained by balancing ROS, SMOTE, VAE, ACGAN, SA-ACGAN and this paper's algorithm to perform 5-fold cross-validation. Each time, 4 subsets in B-Sub_i are taken as the training set, and 1 subset that does not participate in the

training is used as the test set, and finally the average of the 5 results is compared to verify the impact of different balancing algorithms on the classification performance. In addition, 5-fold cross-validation was performed on dataset O to compare the classification performance of the unbalanced dataset.

Step4: The SA-ACGAN-GN algorithm is compared with SA-ACGAN, SA-ACGAN-SN and SA-ACGAN-GP to verify the influence of this paper's algorithm and no stabilization strategy and the use of other stabilization strategies on the training stability.

4.4. Experimental results

4.4.1. Comparison of classification effects between long-tailed datasets and balanced datasets

According to the experimental scheme described in Section 4.3, the MIRAGE-2019 mobile traffic dataset was classified and identified. Table 5 shows the classification performance indicators and their average values obtained by different balance algorithms in the cross-validation experiment.

Table 5. Cross-validation of the classification performance indicators of different balance algorithms.

Performance indicators	Balance algorithm	1st time	2nd time	3rd time	4th time	5th time	Average
precision (%)	unbalanced	60.6	62.1	60.8	62.1	69.7	63.1
	ROS	71.5	79.0	66.7	80.6	81.3	75.8
	SMOTE	81.2	81.4	71.9	80.8	81.1	79.3
	VAE	84.2	87.7	85.9	87.6	88.4	86.8
	ACGAN	84.2	87.3	86.7	90.3	90.4	87.8
	SA-ACGAN	88.1	91.8	90.9	91.7	92.2	90.9
	SA-ACGAN-GN	92.5	93.4	93.1	94.8	95.3	**93.8**
Recall (%)	unbalanced	45.2	50.4	48.5	50.2	51.3	49.1
	ROS	65.9	76.1	65.7	66.8	67.8	68.5
	SMOTE	66.1	80.3	63.8	69.1	70.7	70.0
	VAE	69.8	75.7	71.7	77.7	78.3	74.6
	ACGAN	75.6	82.7	77.1	79.6	80.0	79.0
	SA-ACGAN	81.4	82.0	83.7	82.2	82.8	82.4
	SA-ACGAN-GN	80.4	86.2	85.7	85.9	86.9	**85.0**
F1-Score (%)	unbalanced	48.0	51.2	49.2	51.3	52.2	50.4
	ROS	66.4	72.4	68.0	73.3	75.9	71.2
	SMOTE	67.5	74.7	73.4	75.3	77.6	73.7
	VAE	72.6	77.6	73.0	81.7	85.9	78.2
	ACGAN	81.2	83.9	82.8	86.0	88.6	84.5
	SA-ACGAN	85.7	87.5	86.5	90.3	91.5	88.3
	SA-ACGAN-GN	90.9	92.7	91.7	92.1	93.7	**92.2**

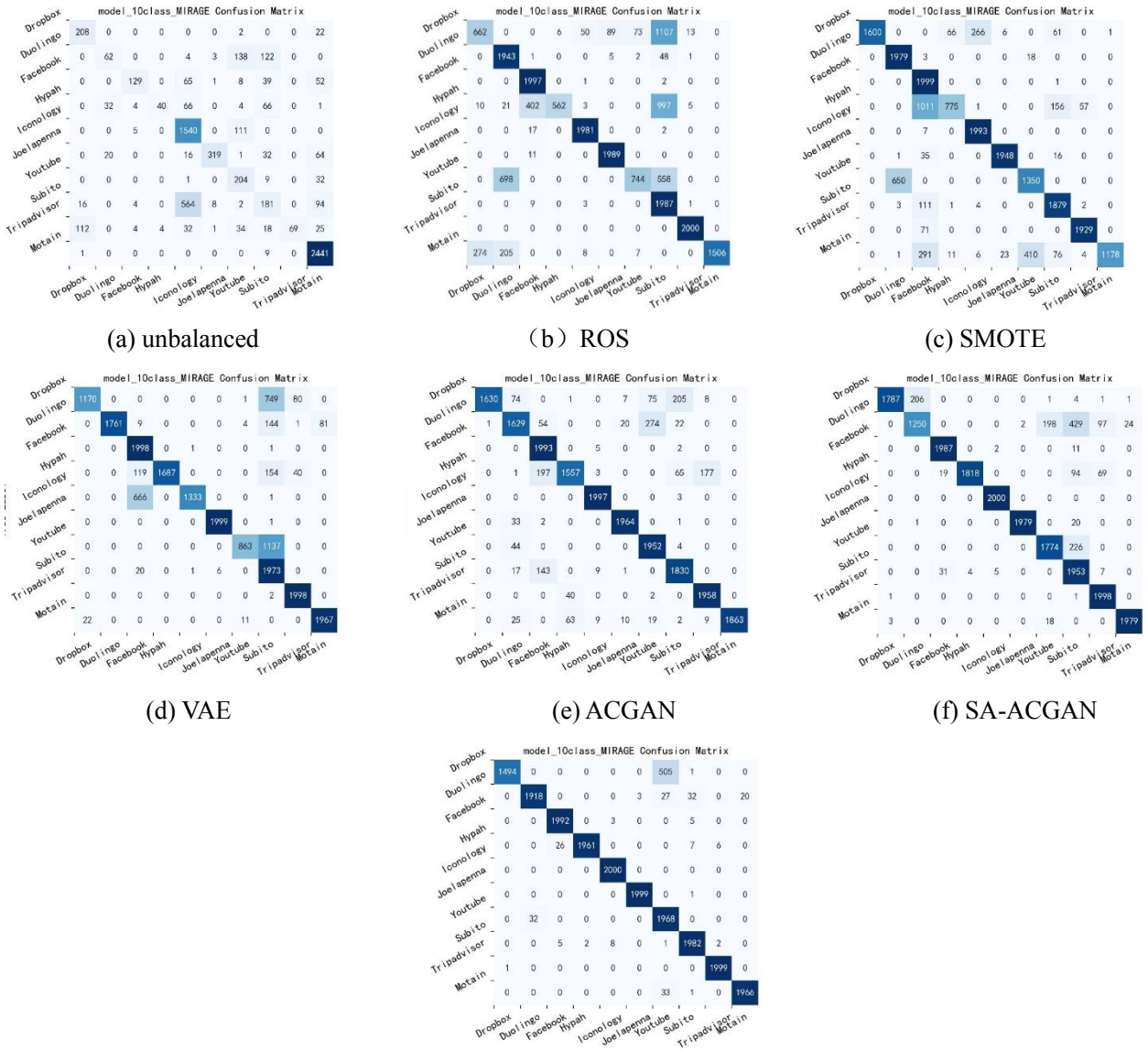
As can be seen from Table 5, the classification performance of different datasets is slightly different due to the randomness of data selection. However, according to the overall performance indicators, the long-tailed dataset does not perform as well as any balanced dataset on the classification model. The

classification performance indicators of the balanced datasets between the ROS and SMOTE methods are relatively close, but the improvement of the indicators is limited because no new features have been learned; the classification indicators of VAE and ACGAN are better than the two traditional balanced datasets, but comparatively the generation result of VAE is relatively vague, so the classification performance index of its balanced dataset is lower than that of ACGAN; the SA-ACGAN-GN data augmentation algorithm used in this paper firstly introduced the self-attention mechanism into ACGAN to solve the problem of unclear local details in the long-distance space of the image, so the classification performance of the SA-ACGAN balanced dataset is further improved; and combined with the GN strategy, it can not only speed up the convergence speed of the model, but also improve the stability of model training, and better improve the quality of image generation. It can be seen from the experimental data that the average classification performance of the algorithm is higher than that of the other five balanced algorithms, the precision can reach 93.8%, the recall rate can reach 85.0%, and the F1-Score can reach 92.2%.

Due to the limited space of the paper, when B-Sub_2 is selected as the test set, the confusion matrix of the CNN classification model is used to analyze the specific classification performance of 10 application types in the test dataset. The confusion matrix is shown in Figure 7. The elements on the diagonal of the confusion matrix are the correctly classified elements, and the shade of color represents the quantity. Figure 7(a) is the confusion matrix of classification results for datasets with long-tailed distribution, which compared with the confusion matrix of the dataset processed by each balancing algorithm represented by Figure 7(b)–(g), the recognition error rate after the dataset with long-tailed distribution is directly sent to the classifier for training is significantly higher than the training and classification recognition error rate of the dataset after the balance of other algorithms. For example, the number of correctly identified samples of the minority class application Hypah in the long-tailed distribution dataset is only 40, accounting for only 18.8% of the dataset, and the dataset balanced by each balancing algorithm has a significantly improved accuracy rate of this class during identification. Especially after using the SA-ACGAN-GN algorithm balanced dataset for training classification, the number of samples that Hypah was correctly identified was 1961, accounting for 98.1% of the dataset. Other minority applications, such as Duoling, Facebook, Joelapenna, Youtube, Tripadvisor, have different degrees of improvement in the recognition accuracy. Therefore, it can be seen that processing the long-tailed dataset can effectively improve the recognition effect of the classifier on the minority class samples.

Figure 7(b)–(e) correspond to the confusion matrices of the dataset classification after balancing the ROS, SMOTE, VAE, and ACGAN algorithms, respectively, it can be seen from the figure that the misclassification of minority class applications has eased, but there is still a high degree of misjudgment. For example, the minority Hypah application is mainly misjudged as Facebook, Subito and Tripadvisor after being balanced by the above four algorithms. The total number of misclassifieds for these three categories in the four algorithms is 1404, 1224, 313 and 439, respectively. This is because the traffic image characteristics of Hypah are similar to the traffic image parts of the above three applications to a certain extent. If the image generation quality is not high, it is easy to cause misjudgment. The total number of Hypah types that were misjudged as Facebook, Subito and Tripadvisor after being balanced by the SA-ACGAN algorithm was 182, as shown in Figure 7(f), indicating that adding the self-attention mechanism to the ACGAN algorithm can grasp the global and local information more comprehensively, improve the ability of network feature extraction, and help the classifier to classify. Finally, after Hypah balanced with the SA-ACGAN-GN algorithm, the total

number of the three categories that were wrongly judged to be Facebook, Subito and Tripadvisor was only 39, as shown in Figure 7(g), indicating that adding in the SA-ACGAN algorithm After GN, the quality of data generation is further improved, so that the classifier has better recognition performance for various application data.



(a) unbalanced (b) ROS (c) SMOTE

(d) VAE (e) ACGAN (f) SA-ACGAN

(g) SA-ACGAN-GN

Figure 7. Confusion matrix for CNN-based mobile traffic classification.

Table 6 shows the average value of the precision index obtained by cross-validation on the trained CNN model in the dataset with long-tailed distribution and 6 different balanced datasets. Focus on the balanced classification indicators of applications that account for less than 10% of the initially constructed dataset. These application types include Dropbox, Duoling, Facebook, Hypah, Joelapenna, Youtube, and Tripadvisor. For the convenience of comparison, the above 7 type names are displayed in bold. Comparing the balanced dataset of the paper's algorithm with the dataset with long-tailed distribution, it can be seen that the precision of several minority applications has been significantly

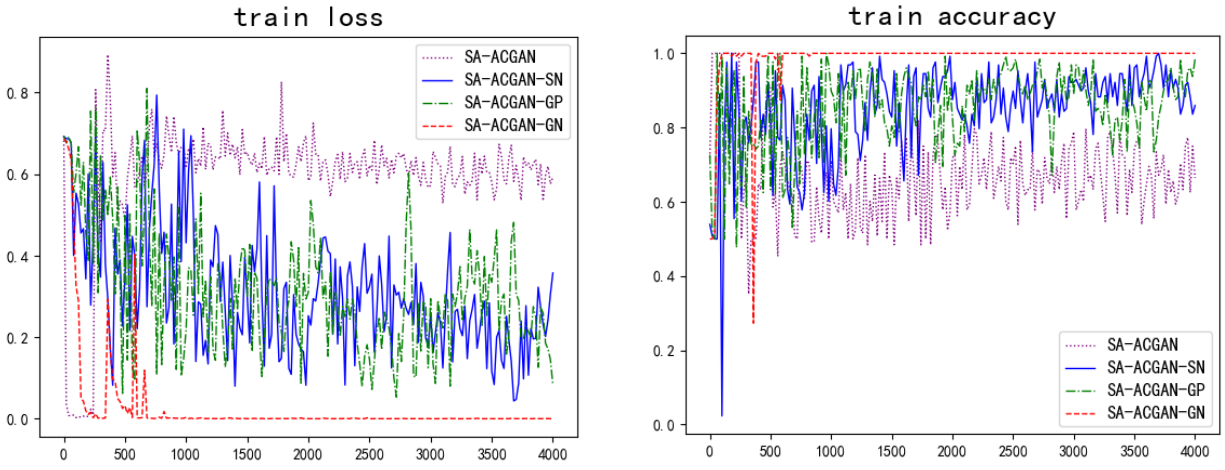
improved. Among them, the average classification precision of Joelapenna and Youtube applications has changed most obviously, both of which have increased by more than 1 times. Although the identification indicators of individual applications (Duolingo and Tripadvisor) are slightly lower than other balanced datasets, from the overall average index, the classification performance of the dataset balanced by the SA-ACGAN-GN algorithm is still higher than the other five balanced datasets, indicating that the improved method is more effective for data augmentation.

Table 6. 5-fold cross-validation experiment classification precision average of various applications.

APP	Unbalanced	ROS	SMOTE	VAE	ACGAN	SA-ACGAN	SA-ACGAN-GN
Dropbox	81.0	85.9	81.3	93.2	86.3	89.4	92.3
Duolingo	66.9	87.1	95.8	96.7	92.7	92.3	93.0
Facebook	72.7	81.7	77.8	86.1	82.8	94.3	93.7
Hypah	70.7	30.6	51.7	67.9	67.9	70.6	83.0
Iconology	29.7	88.4	89.2	93.0	94.2	92.4	92.0
Joelapenna	46.7	79.7	48.6	59.4	83.6	91.6	92.5
Youtube	47.1	32.7	60.9	90.9	65.9	94.7	93.7
Subito	25.3	90.9	80.9	83.6	85.9	87.1	94.0
Tripadvisor	85.0	92.2	90.1	99.2	95.1	90.0	93.4
Motain	99.6	86.9	89.7	95.7	94.7	94.0	93.7

4.4.2. Stability comparison of training process

In order to verify the improvement of the model training stability of the algorithm proposed in this paper, the SA-ACGAN-GN algorithm that introduces gradient normalization is compared with SA-ACAGN, SA-ACGAN-SN and SA-ACGAN-GP that also add self-attention mechanism. The SA-ACGAN, SA-ACGAN-SN and SA-ACGAN-GP all acquire the same network structure and parameter settings as SA-ACGAN-GN. The classification loss and classification accuracy of the above four model training processes are shown in Figure 8.



(a) classification loss

（b）classification accuracy

Figure 8. Comparison of training results of different models.

Figure 8(a) shows the comparison of classification losses in the training process of the above four models. It can be seen that with the same epoch, mini-batch, and number of network layers, SA-ACGAN has the largest value of classification loss; the classification loss of the other two models, SA-ACGAN-SN and SA-ACGAN-GP, gradually drop nearly to zero with the training process, but the decline of speed is slow and there are large fluctuations. For example, there are many sudden increase points of loss values during the period of 2000 to 3500 iterations, reflecting the instability of the generation performance of these two models. The value of classification loss of the SA-ACGAN-GN model decreases significantly faster than the other three contrasting models. It has stabilized after about 800 iterations, and the fluctuation range of the loss curve is smaller. It shows that the SA-ACGAN model has a great improvement in stability and convergence speed after the introduction of the GN mechanism.

Figure 8(b) is a comparison of the classification accuracy of different models. It can be seen from Figure 8(b) that although the classification accuracy of SA-ACGAN can rapidly rise to around 1 in the early stage of training, it has a certain degree of decline after 250 iterations, and there are large fluctuations. The classification accuracy of SA-ACGAN-SN and SA-ACGAN-GP are not much different, but the fluctuation range is still obvious. For example, the number of iterations is between 800 and 2000, indicating that these three models have the problem of training instability. The classification accuracy of the SA-ACGAN-GN model proposed in the paper rises to around 1 in about 600 iterations with the smallest fluctuation, indicating that its classification performance is better and it has a certain effect on improving the stability of model training.

4.4.3. Experimental summary

The above experimental data show that:

1) The original imbalanced dataset is expanded by the method of this paper, which can reduce the impact of long-tailed data on the classification precision and effectively reduce the misjudgment rate of traffic classification.

2) The self-attention mechanism is introduced into the generator and discriminator of ACGAN, which can fully extract the global features of the samples, improve the quality of the traffic image data generated by the model, and the traffic classification performance as well.

3) Using the gradient normalization method can overcome the problems of gradient disappearance and slow convergence speed, further improve the data augmentation effect and strengthen the stability of model training.

5. Conclusions and future work

Aiming at the problem of low accuracy of traffic classification and recognition due to the lack of mobile traffic minority application data, this paper proposes a data augmentation algorithm based on SA-ACGAN-GN, and verifies the superiority of the method on the public mobile traffic dataset. The algorithm has three major advantages. One is to effectively expand the long-tailed dataset and improve the recognition ability of the classification model for minority applications; the other is to combine the self-attention mechanism with ACGAN to enhance the feature extraction ability of the

network, the quality of the generated data is improved, and the ideal data enhancement purpose is achieved; third, the gradient normalization strategy is introduced, which further improves the classification effect and stability of the model. The experimental results show that, compared with other data augmentation algorithms, the algorithm proposed in this paper performs better in classification effect and training stability.

In the future research, after adding the self-attention mechanism, the division weights of the loss function can be designed according to the different influences of the feature points on the classification effect, so that it can focus more on the data samples that are difficult to classify, and further improve the data generation effect. In addition, with the wide application of smart mobile devices, the lightweight design of the model structure is also one of the future work on the premise of keeping the quality of the generated data without significant degradation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62002285).

Conflict of interest

All authors declare no conflict of interest in this paper.

References

1. T. T. T. Nguyen, G. Armitage, A survey of techniques for internet traffic classification using machine learning, *IEEE Commun. Surv. Tutorials*, **10** (2008), 56–76. https://doi.org/10.1109/SURV.2008.080406
2. A. Razaghpanah, A. A. Niaki, N. Vallina-Rodriguez, S. Sundaresan, J. Amann, P. Gill, Studying TLS usage in Android apps, in *Proceedings of the 13th International Conference on emerging Networking Experiments and Technologies*, (2017), 350–362. https://doi.org/10.1145/3232755.3232779
3. M. Talyansky, A. Tumarkin. System and method for optimizing inter-node communication in content distribution network, *P*, US14383062, 2018-01-09.
4. D. Li, Y.F. Zhu, W. Lin. Traffic identification of mobile Apps base on variational autoencoder network. in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, (2017), 287–291. http://doi.org/10.1109/CIS.2017.00069
5. G. Aceto, D. Ciuonzo, A. Montieri, A .Pescapé , Mobile encrypted traffic classification using deep learning, in *2018 IEEE/ACM Network Traffic Measurement and Analysis Conference* , (2018), 1–8. http://doi.org/10.23919/TMA.2018.8506558
6. G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapé, MIMETIC: Mobile encrypted traffic classification using multimodal deep learning, *Comput. Networks,* **165** (2019). http://doi.org/10.1016/j.comnet.2019.106944
7. G. Aceto, D. Ciuonzo, A. Montieri, A .Pescapé , DISTILLER: Encrypted traffic classification via multimodal multitask deep learning, *Network Comput. Appl.*, **183–184** (2021), 102985. https://doi.org/10.1016/j.jnca.2021.102985

8. C. Liu，L. He，G. Xiong，Z. Cao，Z. Li, FS-Net: A flow sequence network for encrypted traffic classification, in *2019 IEEE Conference on Computer Communications*, (2019), 1171–1179. http://doi.org/10.1109/INFOCOM.2019.8737507
9. A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, A. Pescapé, XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures, *IEEE Trans. Network Serv. Manage.*, **18** (2021), 4225–4246. http://doi.org/10.1109/TNSM.2021.3098157
10. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern.*, **42** (2012), 463–484. https://doi.org/10.1109/TSMCC.2011.2161285
11. J. Lee, K. Park, Gan-based imbalanced data intrusion detection system, *Pers. Ubiquitous Comput.*, 25(2021), 121–128. https://doi.org/10.1007/s00779-019-01332-y
12. Y. Hong, U. Hwang, J. Yoo, S. Yoon, How generative adversarial networks and their variants work: An overview, *ACM Comput. Surv.*, **52** (2019), 43. https://doi.org/10.1145/3301282
13. Y. L. Wu, H. H. Shuai, Z. R. Tam, H. Y. Chiu, Gradient normalization for generative adversarial networks, in *International Conference on Computer Vision*, 2021. https://doi.org/10.48550/arXiv.2109.02235
14. N. Japkowicz, Learning from imbalanced data sets: A comparison of various strategies, in *AAAI workshop learning from imbalanced data sets*, (2000), 10–15.
15. N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: Synthetic minority over-sampling technique, *Artif. Intell. Res.*, **16** (2002), 321–357. https://doi.org/10.1613/jair.953
16. H. He, Y. Bai, E. A. Garcia, ADASYN: Adaptive synthetic sampling approach for imbalanced learning., in *Proceedings of International Joint Conference on Neural Networks*, (2008), 1322–1328.
17. Z. Xu, D. Shen, T. Nie, Y. Kou, A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data, *Biomed. Inform.*, (2020). https://doi.org/10.1016/j.jbi.2020.103465
18. R. Hasibi, M. Shokri, M. Dehghan, Augmentation scheme for dealing with imbalanced network traffic classification using deep learning, *Comput. Sci. Networking Int. Archit.*, 2019. https://doi.org/10.48550/arXiv.1901.00204
19. M. A. Arefeen, S. T. Nimi, M. S. Rahman, Neural network-based undersampling techniques, *IEEE Trans. Syst. Man Cybern. Syst.*, 2020. https://doi.org/10.1109/TSMC.2020.3016283
20. F. Folino, G. Folino, M. Guarascio, F. S. Pisani, L. Pontieri, On learning effective ensembles of deep neural networks for intrusion detection, *Inform. Fusion*, **72** (2021), 48–69. https://doi.org/10.1016/j.inffus.2021.02.007
21. P. Bedi, N. Gupta, V. Jindal, I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems, *Appl. Intell.*, **51** (2021), 1133–1151. https://doi.org/10.1007/s10489-020-01886-y
22. N. Gupta, V. Jindal, P. Bedi, CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems, *Comput. Secur.*, **112** (2022), 102499. https://doi.org/10.1016/j.cose.2021.102499
23. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, (2014), 2672–2680.
24. L. C. Yang, S. Y. Chou, Y. H. Yang, MidiNet: A convolutional generative adversarial network for symbolic-domain music generation, preprint, arXiv:1703.10847.

25. S. Yu, S. Zhang, B. Wang, H. Dun, L. Xu, X. Huang, et al., Generative adversarial network based data augmentation to improve cervical cell classification model, *Math. Biosci. Eng.*, **18** (2021), 1740–1752. https://doi.org/10.3934/mbe.2021090
26. R. Durai, R. N. Abirami, Identity preserving multi-pose facial expression recognition using fine tuned VGG on the latent space vector of generative adversarial network, *Math. Biosci. Eng.*, **18** (2021), 3699–3717. https://doi.org/10.3934/mbe.2021186
27. H. Zhang, X. Yu, P. Ren, C. Luo, G. Min, Deep adversarial learning in intrusion detection: A data augmentation enhanced framework, preprint, arXiv:1901.07949.
28. S. Park, H. Park, Combined oversampling and undersampling method based on slow-start algorithm for imbalanced network traffic, *Computing*, **103** (2021), 401–424. https://doi.org/10.1007/s00607-020-00854-1
29. L. Vu, B. C. Thanh, U. Nguyen, A deep learning based method for handling imbalanced problem in network traffic classification, in *Eighth International Symposium on Information and Communication Technology*, (2017), 333–339. https://doi.org/10.1145/3155133.3155175
30. G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, *Expert Syst. Appl.*, **91** (2017), 464–471. https://doi.org/10.1016/j.eswa.2017.09.030
31. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, preprint, arXiv:1701.07875.
32. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of Wasserstein gans, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, (2017), 5769–5779.
33. T. Miyato, T. Kataoka, M. Koyama,Y. Yoshida, Spectral normalization for generative adversarial networks, in *International Conference on Learning Representations*, 2018.
34. H. W. Ding, L. Y. Chen, L. Dong, Z. W. Fu, X. H. Cui, Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection, *Future Gener. Comput. Syst.*, **131** (2022), 240–254. https://doi.org/10.1016/j.future.2022.01.026
35. M. Zheng, T. Li, R. Zhu, Y. H. Tang, M. J. Tang, L. L. Lin, et al., Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification, *Inform. Sci.*, **512** (2020), 1009–1023. https://doi.org/10.1016/j.ins.2019.10.014
36. L. M. Xua, X. H. Zeng, Z. W. Huang, W. S. Li, H. Zhang, Low-dose chest X-ray image super-resolution using generative adversarial nets with spectral normalization, *Biomed. Signal Proc. Control*, **55** (2020), 101600. https://doi.org/10.1016/j.bspc.2019.101600

AIMS Press