JUSTC
Article
http://justc.ustc.edu.cn
Received: May 08, 2023; Accepted: August 04, 2023

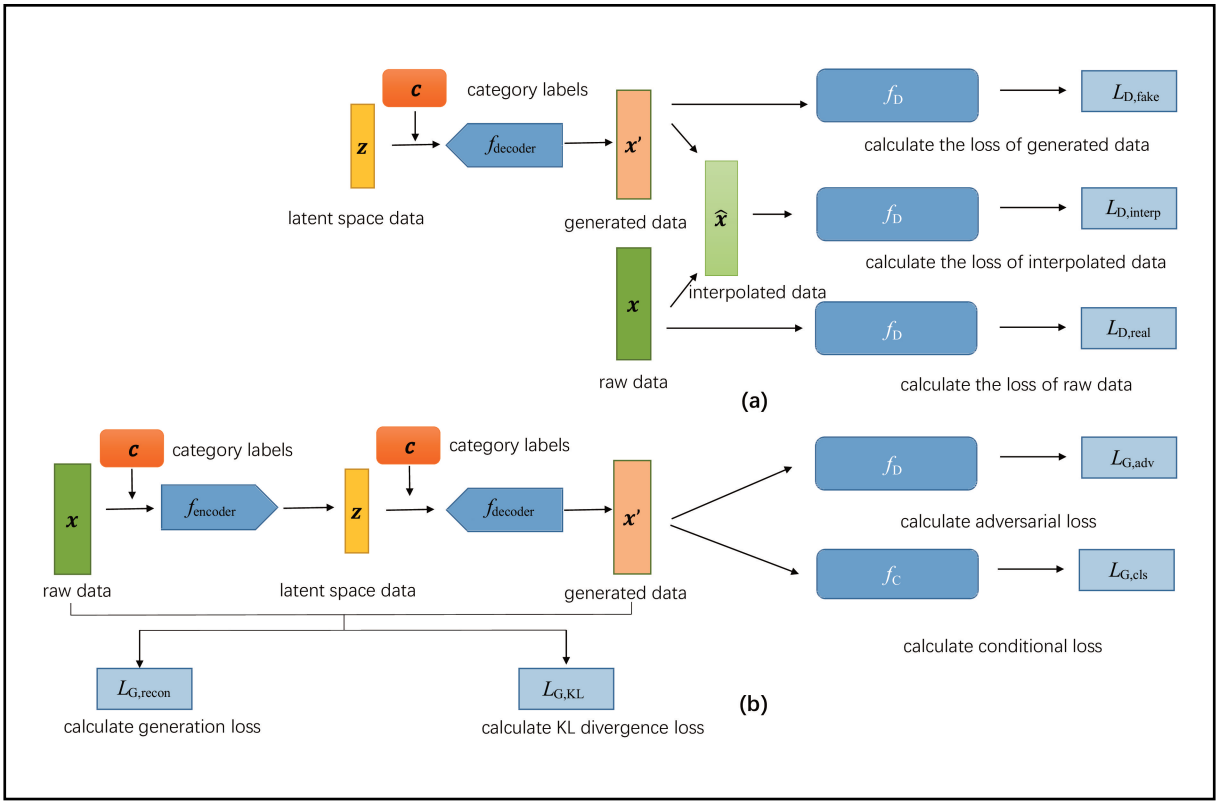
Oversampling for class-imbalanced learning in credit risk assessment based on CVAE-WGAN-gp model

Kaiming Wang[1] ✉, and Qing Yang[2] ✉

[1]School of Management, University of Science and Technology of China, Hefei 230026, China;
[2]International Institute of Finance, School of Management, University of Science and Technology of China, Hefei 230026, China

✉Correspondence: Kaiming Wang, E-mail: wk986864@mail.ustc.edu.cn; Qing Yang, E-mail: yangq@ustc.edu.cn

Graphical abstract



(a) The training dynamics of the discriminator f_D in the CVAE-WGAN-gp model. (b) The generator f_G's training journey in the same model.

Public summary

- Essential for risk management: This study emphasizes credit risk assessment's critical role in banking, aiming to diminish non-performing loans.
- Tackling data imbalance: It addresses class imbalance in credit default datasets, enhancing machine learning models' predictive accuracy.
- Applying CVAE-WGAN-gp: The novel use of conditional variational autoencoder-Wasserstein generative adversarial network with gradient penalty improves credit risk prediction, as validated on various bank loan datasets.

Citation: Wang K M, Yang Q. Oversampling for class-imbalanced learning in credit risk assessment based on CVAE-WGAN-gp model. *JUSTC*, **2025**, 55(7): 0704. CSTR: 32290.14.JUSTC-2023-0084. DOI: 10.52396/JUSTC-2023-0084

JUSTC

Article

http://justc.ustc.edu.cn

Received: May 08, 2023; Accepted: August 04, 2023

Oversampling for class-imbalanced learning in credit risk assessment based on CVAE-WGAN-gp model

Kaiming Wang[1] ✉, and Qing Yang[2] ✉

[1]*School of Management, University of Science and Technology of China, Hefei 230026, China;*
[2]*International Institute of Finance, School of Management, University of Science and Technology of China, Hefei 230026, China*

✉Correspondence: Kaiming Wang, E-mail: wk986864@mail.ustc.edu.cn; Qing Yang, E-mail: yangq@ustc.edu.cn

Cite This: *JUSTC*, **2025**, 55(7): 0704 (12pp) Read Online SI Supporting Information

Abstract: Credit risk assessment is a crucial task in bank risk management. By making lending decisions based on credit risk assessment results, banks can reduce the probability of non-performing loans. However, class imbalance in bank credit default datasets limits the predictive performance of traditional machine learning and deep learning models. To address this issue, this study employs the conditional variational autoencoder-Wasserstein generative adversarial network with gradient penalty (CVAE-WGAN-gp) model for oversampling, generating samples similar to the original default customer data to enhance model prediction performance. To evaluate the quality of the data generated by the CVAE-WGAN-gp model, we selected several bank loan datasets for experimentation. The experimental results demonstrate that using the CVAE-WGAN-gp model for oversampling can significantly improve the predictive performance in credit risk assessment problems.

Keywords: credit risk assessment; class imbalance; oversampling; conditional variational autoencoder (CVAE); generative adversarial network (GAN)

CLC number: TP183 **Document code:** A

1 Introduction

In recent years, with the rapid development of China's economy, residents' willingness to borrow has significantly increased. According to the 2021 financial institution loan statistics report released by the People's Bank of China, by the end of 2021, the balance of various RMB loans of financial institutions was 192.69 trillion CNY, an increase of 11.6% year-on-year; RMB loans increased by 19.95 trillion CNY throughout the year, a year-on-year increase of 3.15 trillion CNY. Among them, the non-performing loan ratio of financial institutions reached 1.73%, maintaining a relatively high level. Therefore, assessing the credit risk of bank customers before lending is particularly important. However, due to the excessively large group of bank loan customers, traditional manual assessment methods are not only time-consuming and laborious, but also difficult to achieve the expected results.

Credit risk assessment aims to analyze customer profile information (such as income, age) to determine whether a customer can repay on time. This is essentially a binary classification problem. Compared to traditional manual assessment methods, machine learning[1,2] and deep learning[3] methods demonstrate significant advantages in prediction accuracy, efficiency, and calculation speed, gradually becoming the mainstream method for credit risk assessment in banks. However, unlike other binary classification problems, credit risk assessment has the characteristic of class imbalance. According to the *China Financial Stability Report (2021)*①, the nonperforming loan ratio of Chinese banks in recent years has been between 1% and 2%. Therefore, in credit risk assessment, the number of samples with a label value of 0 (non-default loans) is significantly higher than that of samples with a label value of 1 (default loans). Johnson et al.[4] demonstrated through experiments that, in cases of significant class imbalance, both machine learning models and deep learning models tend to learn information from majority class samples while ignoring information from minority class samples, which also leads to minority class samples being more likely to be misclassified as majority class samples. Although such models may have a high accuracy rate (due to the larger proportion of non-defaulting customers), they often fail to accurately predict potential credit defaulters in credit risk assessment.

The most common method to address class imbalance in credit risk assessment is data augmentation. Currently, two widely used data augmentation methods are undersampling and oversampling. Undersampling reduces the number of majority class samples in the dataset to be closer to the number of minority class samples, while oversampling increases the number of minority class samples in the dataset to be closer to the number of majority class samples. Mohammed et al.[5] pointed out that in problems with severe class imbalance, such as credit assessment and fraud detection, using oversampling methods to enhance the dataset usually results in better prediction performance than undersampling methods which often leads to poor prediction results due to discarding

① http://www.pbc.gov.cn/goutongjiaoliu/113456/113469/4332768/2021-090315580868236.pdf

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

a large amount of data. Therefore, traditional oversampling methods, including random oversampling and synthetic minority oversampling technique (SMOTE) oversampling, have become the most commonly used methods for handling class imbalance problems in credit assessment[6,7].

However, the aforementioned oversampling methods are based solely on sampling minority class samples, ignoring the majority class samples in the dataset, which often results in a generated data distribution that is inconsistent with the original data distribution. Wang et al.[8] pointed out that when the distribution of training data and testing data is inconsistent, machine learning models tend to overfit, significantly reducing the model's prediction performance on the test set. Therefore, to address this issue, more sophisticated oversampling methods are needed. Oversampling methods that generate samples based on both majority class samples and minority class samples can better maintain data distribution consistency and improve the model's generalization ability.

we use generative adversarial networks (GANs) to generate new data. A generative adversarial network consists of a generator and a discriminator. The generator creates new data based on the original data, and then the mixed original and new data are fed into the discriminator. The discriminator is responsible for detecting whether the generated data is original or newly created. In this way, the generator and the discriminator are trained alternately, ultimately producing new data that approximates the original data distribution. This helps avoid overfitting in machine learning models and maintains the predictive capabilities of the model.

However, when using GANs for oversampling in credit risk assessment problems, the following issues still exist: Traditional GANs typically use fully connected layers or convolutional neural networks as generators. However, since credit assessment datasets often contain a large amount of irrelevant information, these generators may not be able to handle these features effectively. Therefore, we need to find new generator models that can better adapt to the characteristics of credit assessment data and extract useful information from the dataset.

To solve this problem, we adopt the conditional variational autoencoder (CVAE) model as the generator for the GAN model, and construct a conditional variational autoencoder-Wasserstein generative adversarial network with gradient penalty (CVAE-WGAN-gp) model for oversampling in credit assessment problems. The CVAE model is based on an encoding-decoding mechanism, where the encoding mechanism compresses the original data into a low-dimensional latent space, filtering out a large amount of irrelevant information in the dataset; the decoding mechanism generates new data from the noisy data in the latent space, and the resulting data is more consistent with the original data distribution. Additionally, we train the model using the WGAN-gp method, which can enhance the stability of the model and avoid problems such as gradient vanishing to a certain extent during training. Furthermore, the "conditional" mechanism of the CVAE-WGAN-gp model ensures that we can generate data of specified categories, achieving the purpose of generating minority class data.

In multiple credit risk assessment datasets, we compared the data quality generated by the oversampling method based on the CVAE-WGAN-gp model with that of other oversampling methods. The experimental results show that our proposed CVAE-WGAN-gp model has a significant advantage in dealing with imbalanced credit risk assessment data. To further analyze the model performance, we designed two ablation studies. In the first experiment, we investigated whether using WGAN-gp loss in the CVAE-WGAN-gp model could improve the quality of oversampled data. In the second experiment, we studied the impact of latent space dimension k on the data generation effect of the CVAE-WGAN-gp model.

The main contributions of this paper are as follows: (ⅰ) We propose a CVAE-WGAN-gp based oversampling method for credit assessment problems. Compared to other oversampling methods, our approach can generate new data closer to the original data distribution, thereby improving the predictive capabilities of machine learning and deep learning models on credit assessment datasets. (ⅱ) By using the CVAE model as the generator for the WGAN-gp model, we can filter out irrelevant information in the dataset and generate new data from the latent space that conforms to the original data distribution, as well as generate data for specific categories. (ⅲ) By training the model with the WGAN-gp method, we enhance the stability of the model and avoid issues such as vanishing gradients during training.

2 Preliminaries

In this section, we mainly introduce the fundamental knowledge and related research work associated with this paper. In the first two subsections, we provide an overview of the basic concepts of VAE models (including CVAE models) and GAN models separately. The third subsection, which is particularly relevant to our work, focuses on discussing the application of GAN models in oversampling methods.

2.1 VAE and CVAE

The variational autoencoder (VAE)[9] is a generative deep learning model that learns the latent representation of input data for data compression and generation. VAE consists of an encoder and a decoder. The encoder transforms the input data into a distribution in the latent vector space, typically represented by a Gaussian distribution. The decoder transforms the latent vector into output data. The loss function includes the reconstruction loss and KL divergence loss. VAE has been widely used in the field of image processing[10,11].

The conditional variational autoencoder (CVAE)[12] improves upon VAE by introducing conditional information to generate specific categories of data. The loss function of the CVAE consists of reconstruction loss, KL divergence loss, and conditional loss. CVAE has been widely used in fields such as image generation[13,14] and natural language processing[15,16], among other fields.

2.2 GANs

Generative adversarial networks (GANs)[17] are a type of deep learning model that generates data through the cooperation of a generator and a discriminator neural network. The loss function of GANs combines the generator and discriminator loss

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

functions, where the generator aims to generate fake data similar to real data, and the discriminator aims to distinguish between real and fake data. WGAN[18] is an improved GAN model that uses the Wasserstein distance as a loss function, leading to more stable training and avoiding issues such as mode collapse and gradient vanishing during the training process. WGAN-gp[19] further improves upon WGAN by introducing a gradient penalty term that constrains the discriminator output to be continuous between real and fake data, more effectively solving the gradient vanishing problem during training.

In recent years, GANs have made significant progress in various areas such as image generation[20,21], text generation[22], and text-to-image synthesis[23], providing strong support for the future development of deep learning. Researchers have also explored the combination of GANs with variational autoencoders (VAEs)[24], which has shown promising results in image processing tasks.

3 Related work

Although GANs have been predominantly used in image and text data generation, recently, there have been some oversampling methods based on GANs. These methods are aimed at creating synthetic data that is similar to the original dataset and can effectively improve the prediction performance of machine learning and deep learning models on imbalanced datasets.

The medGAN method[25] is specifically designed to handle diagnosis code datasets related to patient visits. This type of dataset is high-dimensional and sparse, and consists only of binary columns. Notably, medGAN uses a pretrained encoder to simplify the task of the generator in handling this unique data structure. Baowaly et al.[26] replaced the standard GAN loss with the Wasserstein GAN gradient penalty (WGAN-gp) loss in the context of medGAN. Experimental results showed that the WGAN-gp loss function, which is effective in the field of image generation, is also effective in oversampling tabular data.

Some architectures are specifically designed for tabular data with categorical and numerical attributes. TGAN[27] creates each column sequentially in the generator using LSTM units with learned attention weights, which allows the generator to model the relationships between columns. However, another work, CTGAN[28], abandoned the use of LSTM units and instead used ordinary fully connected layers because LSTM units were computationally inefficient and did not significantly improve performance. TGAN used one-hot encoding for categorical feature columns and a method based on Gaussian mixture models for numerical feature columns. CTGAN also used the WGAN-gp method for training, which ensured the stability of the training process and helped to generate higher quality data, resulting in better performance in generating tabular data.

Mottini et al.[29] proposed a model for generating passenger name record data used in the airline industry. Notably, they used cross-layers[30] in both the generator and discriminator to explicitly calculate feature interactions. This method passed soft one-hot encoding to the discriminator using categorical embeddings, thus avoiding the addition of noise, as in the TGAN and CTGAN models. Additionally, they tested Gaussian mixture models for numerical columns and found that this reduced the quality of the generated data. To improve WGAN[31], the method used the multivariate extension of the Cramér distance as the loss function.

Justin et al.[32] explored the use of conditional Wasserstein GANs (CWGANs) to address class imbalance. CWGANs are based on WGANs and introduce conditional variables that enable them to generate data with specific class attributes. The CWGAN training objective function includes the loss functions of the generator and discriminator, as well as an auxiliary classifier network loss function. The auxiliary classifier network can help CWGAN generate samples of minority classes on demand, thus balancing the class distribution of the dataset. CWGAN combines the stability of WGAN-gp with the class generation ability of ACGAN, making it perform exceptionally well in addressing class imbalance issues and improving classifier performance on imbalanced datasets.

4 CVAE-WGAN-gp model

The CVAE-WGAN-gp model is composed of three modules: a generator f_{G}, a discriminator f_{D}, and a classifier f_{C}. The generator's role is to generate synthetic data that closely resembles real data, while the discriminator's purpose is to differentiate real data from generated synthetic data. The classifier's task is to verify whether the generated synthetic data meets the specified class condition. This model steers the generator's learning through reconstruction loss, KL divergence loss, adversarial loss, and conditional loss to produce synthetic data that adheres to both the real data distribution and the given class condition. In the following three subsections, we introduce the fundamental architecture of the CVAE-WGAN-gp model. In the fourth and fifth subsections, we detail the training process of the CVAE-WGAN-gp model and how to apply this model in practice, respectively. Finally, we summarize the overall training process of the CVAE-WGAN-gp model in Algorithm 1.

4.1 Generator

The generator f_{G} of the CVAE-WGAN-gp model is built upon the CVAE model, which consists of an encoder f_{encoder} and a decoder f_{decoder}. Both the encoder and decoder are constructed using three-layer fully connected layers, with a batch normalization layer added after each of the first two fully connected layers to enhance model stability and generalization. The ReLU function serves as the activation function after each layer to capture the network's nonlinearity.

The CVAE model's training process can be divided into two stages: the training stage and the data generation stage. In the training stage, the model uses real data to calculate the generator loss and updates the parameters of the encoder and decoder based on the generator loss to achieve better reconstruction of input data and generation of new data. In the data generation stage, the model randomly samples a vector from the latent space and inputs it to the decoder to generate fake data. The parameters of the encoder and decoder are not updated in this stage.

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

During the training stage, the CVAE model accepts the original data $\boldsymbol{x}$ and class condition $\boldsymbol{c}$ as input to the encoder f_{encoder}. The low-dimensional space where the learned latent representation vector $\boldsymbol{z}$ resides is called the latent space. The encoder compresses the original data $\boldsymbol{x}$ into the latent space by learning the distribution $P(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})$, and then obtains the latent representation vector $\boldsymbol{z}$ of the original data in the latent space:

$$\boldsymbol{z} = f_{\text{encoder}}(\boldsymbol{x},\boldsymbol{c}). \tag{1}$$

Then, we use the latent representation vector $\boldsymbol{z}$ and class condition $\boldsymbol{c}$ as input to the decoder f_{decoder}. The decoder tries to learn the conditional distribution $P(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c})$ to generate new data $\boldsymbol{x}'$ similar to the original data $\boldsymbol{x}$:

$$\boldsymbol{x}' = f_{\text{decoder}}(\boldsymbol{z},\boldsymbol{c}). \tag{2}$$

Finally, we train the generated data using the discriminator f_{D} and the classifier f_{C}, and update the parameters of the encoder and decoder through gradient descent. Specifically, the parameters of the encoder and decoder are updated to optimize the generation ability of the CVAE model, thereby improving the similarity between the generated data $\boldsymbol{x}'$ and the original data $\boldsymbol{x}$.

In the data generation stage, we randomly generate a set of noise $\tilde{\boldsymbol{z}}$ from the latent space determined in the model training stage. Input it to the decoder f_{decoder} together with the selected class label $\tilde{\boldsymbol{c}}$ to generate fake data $\tilde{\boldsymbol{x}}$:

$$\tilde{\boldsymbol{x}} = f_{\text{decoder}}(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{c}}). \tag{3}$$

The reparameterization method is used to train the CVAE model to avoid mode collapse in the CVAE-WGAN-gp model and to prevent gradient vanishing, thereby improving the model training. This method assumes that the vector $P(\boldsymbol{z}|\boldsymbol{x})$ in the latent space follows a normal distribution with mean μ and variance σ^2, and the encoder f_{encoder} does not directly map the original data to the representation vector $\boldsymbol{z}$ in the latent space, but maps the data to μ and σ. In the data generation phase, we sample a random variable $\boldsymbol{\epsilon}$ from a standard normal distribution and obtain the latent variable $\boldsymbol{z}$ through the following formula:

$$\boldsymbol{z} = \mu + \sigma \odot \boldsymbol{\epsilon}, \tag{4}$$

where $\odot$ represents element-wise multiplication. Through the reparameterization method, we can combine the randomness and sampling operations in the model with the parameter updating process, thereby avoiding mode collapse in the CVAE-WGAN-gp model.

Our generator, as shown in Fig. 1, consists of two main stages: the training stage and the data generation stage. In the training stage, the model samples from the transformed data to obtain the original features $\boldsymbol{x}$ and their corresponding class labels $\boldsymbol{c}$. Then they are input into the encoder. The encoder consists of three fully connected layers (F1, F2, and F3), with each layer's dimension decreasing sequentially to achieve encoding. After each of the first two fully connected layers, we add a normalization layer (BN1, BN2) to improve the model's stability and generalization. Then the ReLU function is adopted as the activation function. The final fully connected layer F3 maps the input data to the mean μ and standard deviation σ in the latent space. Next, we sample a random vector $\boldsymbol{\epsilon}$ and obtain the representation vector $\boldsymbol{z}$ in the latent space based on Eq. (1). Then, $\boldsymbol{z}$ and the class label $\boldsymbol{c}$ are input into the decoder f_{decoder}. The decoder also consists of three fully connected layers (F4, F5, and F6), with each layer's dimension increasing sequentially to achieve decoding. In the first two fully connected layers (F4 and F5), we use a normalization layer (BN4 and BN5) to ensure the

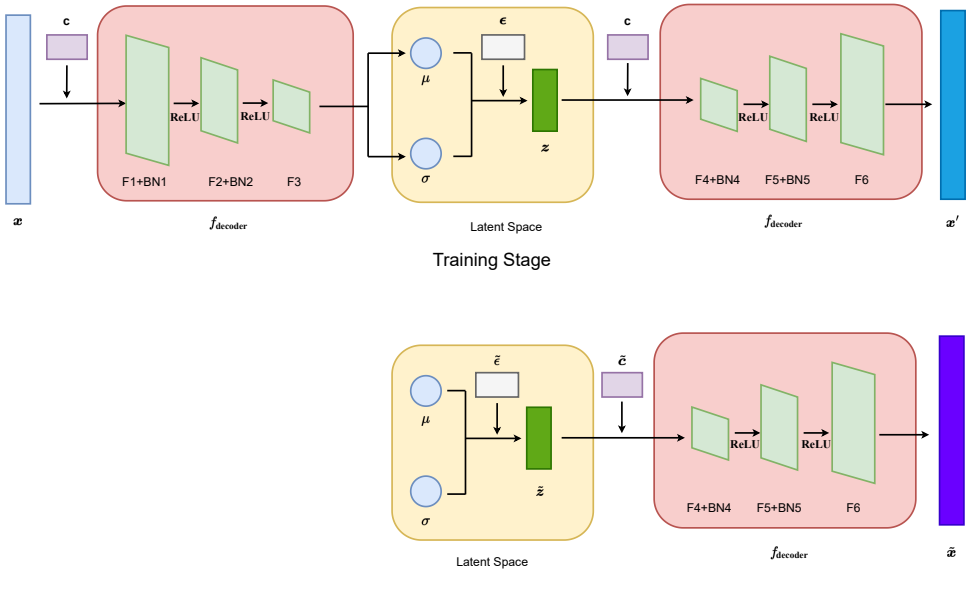


Fig. 1. Generator architecture diagram. The generator of the CVAE-WGAN-gp model consists of a training stage and a data generation stage. The training stage involves sampling the original features and corresponding class labels, as well as using a three-layer fully connected encoder. The data generation stage uses a decoder to map back to the original data space, yielding the reconstructed data. Both the encoder and decoder designs include a normalization layer to enhance the stability and generalization of the model.

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

stability of the decoder f_{decoder}. Finally, we use a fully connected layer F6 to obtain the reconstructed data $\boldsymbol{x}'$ with the same dimension as the original data $\boldsymbol{x}$.

In the data generation stage, we randomly sample the representation vector $\tilde{\boldsymbol{z}}$ from the latent space obtained in the training stage (assuming the latent space follows a Gaussian distribution, so the latent space can be simplified to mean μ and standard deviation σ) and the selected class label $\tilde{\boldsymbol{c}}$. Then, we input the representation vector $\tilde{\boldsymbol{z}}$ and selected class label $\tilde{\boldsymbol{c}}$ into the decoder f_{decoder} to generate the fake data $\tilde{\boldsymbol{x}}$ that resembles the original data.

4.2 Discriminator

In the CVAE-WGAN-gp model, the discriminator f_{D} is responsible for distinguishing the generated data from the real data by predicting the label of the generated data as 0 and the label of the real data as 1. Specifically, we use a three-layer residual block as the discriminator, with a dropout layer added after each layer. We use the leaky ReLU function as the activation function and apply the Sigmoid function after the last layer to map the input to 0−1, to determine whether the input data is real data. The dropout rate of the dropout layer is set to 0.5, and the alpha value of leaky ReLU is set to 0.02. The residual network can effectively avoid the problem of vanishing gradients and ensure the convergence speed of the network even when the model is deep.

4.3 Classifier

We use a three-layer fully connected network as the classifier f_{C} in the CVAE-WGAN-gp model, with a similar architecture to the discriminator. Since we are solving a binary classification problem, we use the Sigmoid function as the activation function in the last layer to determine whether the data corresponds to a customer that is likely to have a credit default issue.

4.4 Training process of CVAE-WGAN-gp model

During the training process of the CVAE-WGAN-gp model, we first train the classifier f_{C} by sampling real data $\boldsymbol{x}$ and its corresponding label $\boldsymbol{c}$ from the training set. We use $\boldsymbol{x}$ as input to the classifier and apply the cross-entropy function to the predicted results $f_{\text{C}}(\boldsymbol{x})$ and the true label $\boldsymbol{c}$, which forms the following classifier loss function L_{C}:

$$L_{\text{C}} = L_{\text{CE}}(f_{\text{C}}(\boldsymbol{x}), \boldsymbol{c}). \tag{5}$$

Here, L_{CE} denotes the cross-entropy loss function. We use gradient descent to minimize L_{C} and update the parameters of the classifier. After completing the training of the classifier, we fix its parameters and use it to predict the category of the generated data by the generator, to ensure the consistency between the generated data and the real data in terms of category.

Next, we train the discriminator f_{D} of the CVAE-WGAN-gp model based on the WGAN-gp method. Since the WGAN-gp uses a non-continuous loss function, the discriminator may become unstable. Therefore, in the WGAN-gp, we need to train the discriminator for multiple rounds to ensure the stability of the model. Assume that the discriminator reaches a stable state after N_{D} rounds of training. In each round of training, we first use the reparameterization method to randomly sample a set of vectors $\tilde{\boldsymbol{z}}$ from the latent space and generate a set of random class conditions $\tilde{\boldsymbol{c}}$. We then use the decoder f_{decoder} to generate a set of fake data $\tilde{\boldsymbol{x}}$. Next, we input the generated fake data and real data into the discriminator f_{D}, compute its loss function, and update the discriminator's weights.

The discriminator's loss function consists of three parts. The first part is the expected score of the discriminator's prediction on real data, known as the real data loss $L_{\text{D,real}}$, which is defined as:

$$L_{\text{D,real}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[f_{\text{D}}(\boldsymbol{x})]. \tag{6}$$

The second part is the expected score of the discriminator's prediction on fake data, known as the generated data loss $L_{\text{D,fake}}$, which is defined as:

$$L_{\text{D,fake}} = \mathbb{E}_{\tilde{\boldsymbol{z}}\sim p_{\tilde{\boldsymbol{z}}}}[f_{\text{D}}(\tilde{\boldsymbol{x}})]. \tag{7}$$

Finally, we introduce a gradient penalty term to ensure that the gradient of the discriminator stays within a certain range. This penalty term can be computed on the linear interpolation between the real and fake data, known as the interpolated data loss $L_{\text{D,interp}}$ as follows:

$$L_{\text{D,interp}} = \mathbb{E}_{\hat{\boldsymbol{x}}\sim p_{\hat{\boldsymbol{x}}}}[(\|\nabla_{\hat{\boldsymbol{x}}} f_{\text{D}}(\hat{\boldsymbol{x}})\|_2 - 1)^2]. \tag{8}$$

Here, $\hat{\boldsymbol{x}}$ is the linear interpolation between real data $\boldsymbol{x}$ and fake data $\tilde{\boldsymbol{x}}$, i.e., $\hat{\boldsymbol{x}} = \alpha\boldsymbol{x} + (1-\alpha)\tilde{\boldsymbol{x}}$, $\alpha \in (0,1)$. Specifically, the loss function of the discriminator L_{D} is defined as follows:

$$L_{\text{D}} = -L_{\text{D,fake}} - L_{\text{D,real}} + \lambda L_{\text{D,interp}}, \tag{9}$$

where λ is the weight of the penalty term. Note that the negative sign is introduced here because in CVAE-WGAN-gp, we need to minimize the discriminator's loss. After training the discriminator, its parameters are fixed, and it is used to compute the loss function L_{G} of the CVAE-WGAN-gp model's generator f_{G}.

Finally, we train the generator f_{G}. We feed the real data $\boldsymbol{x}$ and the class condition $\boldsymbol{c}$ into the encoder f_{encoder}, and generate the data $\boldsymbol{x}'$ through the decoder f_{decoder} with the latent representation vector $\boldsymbol{z}$ in the latent space. Next, we send the generated data $\boldsymbol{x}'$ into the discriminator f_{D} to measure the similarity between the generated data and the real data, and at the same time, we send it into the classifier f_{C} to check if it satisfies the given class condition $\boldsymbol{c}$. To improve the generator's generation performance, the parameters of the generator are updated by minimizing the distance between the generated data and the original data, and making them consistent with each other in terms of class conditions.

Therefore, the generator's loss function L_{G} in CVAE-WGAN-gp consists of four parts. The first one is the reconstruction loss $L_{\text{G,recon}}$, which measures the similarity between the generated sample $\boldsymbol{x}'$ and the original data $\boldsymbol{x}$. We use the mean square error to measure the reconstruction loss,

$$L_{\text{G,recon}} = \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2. \tag{10}$$

The second part is the KL divergence loss $L_{\text{G,KL}}$, which ensures that the distribution of generated samples is similar to the prior distribution. By using the reparameterization trick, the model's KL divergence loss $L_{\text{G,KL}}$ can be expressed as:

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

$$L_{\mathrm{G,KL}} = \frac{1}{2}(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} + \sum(\exp(\sigma) - \sigma - 1)). \tag{11}$$

The third part is the adversarial loss $L_{\mathrm{G,adv}}$, which makes the generated samples deceive the discriminator f_{D} and be recognized as real data. We use the cross-entropy function L_{CE} to measure the adversarial loss,

$$L_{\mathrm{G,adv}} = L_{\mathrm{CE}}(f_{\mathrm{D}}(\boldsymbol{x}'), \mathbf{1}). \tag{12}$$

The fourth part is the conditional loss $L_{\mathrm{G,cls}}$, which determines whether the generated fake data satisfies the given category conditions. We also use the cross-entropy function L_{CE} to measure the model's conditional loss,

$$L_{\mathrm{G,cls}} = L_{\mathrm{CE}}(f_{\mathrm{C}}(\boldsymbol{x}'), \boldsymbol{c}). \tag{13}$$

The final generator loss function L_{G} can be expressed as a linear combination of the four loss functions, where the weight of each loss function is adjusted by the corresponding hyperparameters,

$$L_{\mathrm{G}} = \lambda_{\mathrm{G,recon}} L_{\mathrm{G,recon}} + \lambda_{\mathrm{G,KL}} L_{\mathrm{G,KL}} + \lambda_{\mathrm{G,adv}} L_{\mathrm{G,adv}} + \lambda_{\mathrm{G,cls}} L_{\mathrm{G,cls}}, \tag{14}$$

where $\lambda_{\mathrm{G,recon}}$, $\lambda_{\mathrm{G,KL}}$, $\lambda_{\mathrm{G,adv}}$, and $\lambda_{\mathrm{G,cls}}$ are hyperparameters that balance the relative importance of the four losses. Algorithm 1 summarizes the CVAE-WGAN-gp training process.

Algorithm 1: CVAE-WGAN-gp training algorithm

Require: latent space dimension k; number of discriminator training rounds N_{D}; discriminator penalty weight λ; generator loss hyperparameter $\lambda_{\mathrm{G,recon}}$, $\lambda_{\mathrm{G,KL}}$, $\lambda_{\mathrm{G,adv}}$, $\lambda_{\mathrm{G,cls}}$; learning rate α_{G}, α_{D}, α_{C};

Initialization: generator f_{G} (encoder f_{encoder} and decoder f_{decoder}), discriminator f_{D}, classifier f_{C};

while θ_{G} has not converged **do**

  $L_{\mathrm{C}} \leftarrow \mathrm{loss}(f_{\mathrm{C}}(\boldsymbol{x}), \boldsymbol{c})$;

  $\theta_{\mathrm{C}} \leftarrow \theta_{\mathrm{C}} - \nabla_{\theta_{\mathrm{C}}}(L_{\mathrm{C}})$;

  for $i = 1$ to N_{D} **do**

    Sample $\tilde{\boldsymbol{z}}$ and $\tilde{\boldsymbol{c}}$;

    Generate fake data $\tilde{\boldsymbol{x}} = f_{\mathrm{decoder}}(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{c}})$;

    $L_{\mathrm{D}} \leftarrow -L_{\mathrm{D,fake}} - L_{\mathrm{D,real}} + \lambda L_{\mathrm{D,interp}}$;

    $\theta_{\mathrm{D}} \leftarrow \theta_{\mathrm{D}} - \nabla_{\theta_{\mathrm{D}}}(L_{\mathrm{D}})$;

  end for;

  Encode real data $\boldsymbol{x}$ with label $\boldsymbol{c}$ to obtain z;

  Generate data $\boldsymbol{x}' = f_{\mathrm{decoder}}(z, \boldsymbol{c})$;

  $L_{\mathrm{G,recon}} \leftarrow \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2$;

  $L_{\mathrm{G,KL}} \leftarrow \mathrm{KL}(q(z|\boldsymbol{x}, \boldsymbol{c}) \| p(z))$;

  $L_{\mathrm{G,adv}} \leftarrow f_{\mathrm{D}}(\boldsymbol{x}', \boldsymbol{c})$;

  $L_{\mathrm{G,cls}} \leftarrow f_{\mathrm{C}}(\boldsymbol{x}')$;

  $L_{\mathrm{G}} \leftarrow \lambda_{\mathrm{G,recon}} L_{\mathrm{G,recon}} + \lambda_{\mathrm{G,KL}} L_{\mathrm{G,KL}} + \lambda_{\mathrm{G,adv}} L_{\mathrm{G,adv}} + \lambda_{\mathrm{G,cls}} L_{\mathrm{G,cls}}$;

  $\theta_{\mathrm{G}} \leftarrow \theta_{\mathrm{G}} - \nabla_{\theta_{\mathrm{G}}} L_{\mathrm{G}}$.

end while.

4.5 Application of CVAE-GAN-gp model in practice

This section provides a detailed explanation of how the CVAE-WGAN-gp model is applied in practice to generate new data. First, we input the imbalanced original dataset into the CVAE-WGAN-gp model to generate new data which are highly similar to the original default customer data. Subsequently, these newly generated data are merged with the original dataset, and the combined dataset is shuffled randomly. The purpose of randomly shuffling the data is to ensure that each batch, when using a predetermined batch size for model training, contains both original and generated data, allowing the model to encounter a rich variety of data samples during the training process. Finally, we train the prediction model on this merged dataset. Since the combined dataset alleviates the class imbalance problem, the predictive model exhibits better predictive performance compared to a model trained solely on the original dataset. In Section 5, we demonstrate through empirical evidence that the application of the CVAE-WGAN-gp model for data generation can effectively enhance the predictive performance of the model for credit risk assessment.

Furthermore, the CVAE-WGAN-gp model has a wide range of application prospects. Given that this model does not impose any mandatory assumptions on the distribution of input data, it can not only handle the credit risk assessment problem studied in our current research but also extend to various other binary imbalanced problems, including but not limited to fields such as medical diagnosis and fraud detection.

5 Experiments

To evaluate the effectiveness of the CVAE-WGAN-gp model in addressing the class imbalance issue in credit assessment, various oversampling methods are selected as baselines, and their prediction results on different credit assessment datasets are compared with the CVAE-WGAN-gp oversampling method. To measure their performance in handling credit assessment data for category prediction, multiple models are chosen as predictive models. Three commonly used evaluation metrics for class imbalance issues are employed: AUC-ROC (area under the receiver operating characteristic curve, AR), AUC-PR (area under the precision-recall curve, AP), and F1-score (F1). Furthermore, the Friedman test is used to verify whether there is a significant difference in the oversampling performance between the CVAE-WGAN-gp method and its competitors in credit assessment problems. Lastly, two ablation studies are designed to evaluate the oversampling performance of the CVAE-WGAN-gp model, with the first experiment investigating the impact of WGAN-gp loss, and the second experiment studying the effect of the latent space dimension k.

5.1 Datasets and data preprocessing

To validate the universality of the proposed method in credit assessment problems, six credit assessment datasets with class imbalance characteristics (Italy, Germany, Taiwan of China, Express, Card, Kaggle) are selected for the experiment. Table 1 shows the sources and related information of these six datasets. Currently, all datasets are publicly available resources. For each dataset, multiple imputation is first used

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

Table 1. Characteristics of the regorworld credit scoring datasets used for our evaluation.

Name	Source	Samples	Num. columns	Cat. columns	Total columns	Imbalance ratio
Italy	Kaggle	49732	7	9	16	7.6
Germany	UCI MLR	1000	4	5	9	12.9
Taiwan of China	UCI MLR	30000	14	9	23	12.4
Express	American Express	45528	10	3	13	21.2
Card	JanataHack	21000	13	10	23	23.6
Kaggle	Kaggle	45528	9	7	16	27.3

to handle missing values, as GANs cannot generate new data on data with missing values. It is worth noting that during the oversampling phase, feature normalization and feature selection are not performed. This is because traditional oversampling methods, such as SMOTE oversampling, do not require normalization of the dataset, and in oversampling methods based on GANs, the GAN generator has a batch normalization layer that can perform batch normalization of the raw data. Moreover, feature selection may result in generated new data having different dimensions from the original data, making it impossible to achieve the goal of oversampling minority class data.

5.2 Evaluation metrics

In this experiment, we choose AUC-ROC, AUC-PR, and F1-score as the metrics to measure the model's prediction performance. These metrics are robust in the face of class imbalance problems and can reflect the classifier's performance under various oversampling methods from multiple perspectives.

The AUC-ROC is a metric that evaluates the trade-off between the true positive rate and the false positive rate at different thresholds, revealing the classifier's overall performance in distinguishing between positive and negative classes. Similar to AUC-ROC, AUC-PR is also a ranking metric, but it focuses more on the prediction effect of the positive class. The F1-score is the harmonic mean of precision and recall, considering the classifier's prediction accuracy and coverage in the positive class. In class imbalance problems, the F1-score can effectively balance precision and recall, providing a comprehensive performance evaluation metric. Through these evaluation metrics, we can comprehensively assess the impact of different oversampling methods on classifier performance.

For imbalanced classification problems, the AUC-PR metric has a distinct advantage over the AUC-ROC metric, which is the main reason why we chose to use AUC-PR as our evaluation metric. While the AUC-ROC metric is widely used, it does not significantly consider the impact of false negatives during the evaluation process. In credit assessment problems, where the identification of positive instances is particularly important, we need to pay attention to the model's ability to recognize positive instances and minimize false negatives. In this regard, the "numbness" of the AUC-ROC makes it relatively weak in dealing with imbalanced classification problems. On the other hand, the AUC-PR metric is extremely sensitive to both false negatives (positive instances that the model failed to identify) and false positives (negative instances misclassified as positives by the model), making it more suitable for evaluating the performance of models in imbalance classification problems. Therefore, although the AUC-PR metric is less common than the AUC-ROC metric, its sensitivity to imbalanced classification problems has led us to select it as the primary evaluation metric for this study[33,34].

5.3 Baseline oversampling methods

To evaluate the oversampling method based on the CVAE-WGAN-gp, we compare it with no oversampling (None), traditional oversampling methods (such as SMOTE, B-SMOTE, and ADASYN), and GAN-based oversampling methods (such as CTGAN and CWGAN).

The synthetic minority oversampling technique (SMOTE)[35] method expands random oversampling by generating synthetic new minority class samples, using the k-nearest neighbor algorithm of minority class samples to generate new samples. Borderline-SMOTE (B-SMOTE)[36] focuses on borderline minority class samples by finding each minority class sample's m-nearest neighbors and using the SMOTE algorithm to generate synthetic minority class samples. Adaptive synthetic oversampling (ADASYN)[37] is similar to SMOTE, but it selects minority class samples proportionally based on the number of majority class samples in the k-nearest neighbors of the two classes, thus generating more samples for minority samples near the majority samples. In this experiment, we implement these baseline oversampling methods using the imbalanced-learn package[38].

We selected CTGAN and CWGAN as the benchmark models for comparison in the context of GAN-based oversampling methods. This choice primarily rests upon two factors. First, both CTGAN and CWGAN have been demonstrated to effectively transfer to the generation of credit risk assessment data. This is particularly significant because certain GAN models, such as medGAN and the models proposed by Mottini et al.[29], are only applicable in specific domains, such as the generation of diagnostic code datasets and aviation data. Second, CTGAN and CWGAN have shown superior performance compared to other GAN models that can be transferred to the generation of credit risk assessment data. For example, in comparison to the TGAN model, the CTGAN model has shown superiority in data generation performance, a point empirically validated in the original CTGAN paper[28]. Based on these considerations, we selected

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

CTGAN and CWGAN as the GAN-based comparison models.

5.4 Prediction models

In this study, we test each oversampling method with 6 prediction models, including k-nearest neighbors (KNN), decision tree (DT), random forest (RF), light gradient boosting machine (LGB), extreme gradient boosting (XGB), and multilayer perceptron (MLP). These models cover commonly used single learner models (KNN and DT), ensemble learning models (RF, LGB, and XGB), and deep learning models (MLP) in the credit assessment field.

Among the above models, KNN and DT, as single learner models, have intuitiveness and easy-to-understand characteristics but have higher computational complexity and are prone to overfitting. In contrast, RF, LGB, and XGB, as ensemble learning models, have strong generalization ability and lower overfitting risk but require parameter tuning. These ensemble methods combine the predictions of multiple base classifiers, thus providing more accurate and stable results in many problems. MLP, as a deep learning model, performs well in handling complex problems and large-scale data but may have a slower training process and requires a large amount of data to achieve good generalization performance. The application of these models in credit scoring has received widespread attention and validation, and they can accurately evaluate the prediction effects of different oversampling methods on credit assessment problems.

In the experimental setup, we choose the number of nearest neighbors for KNN to be 5 to ensure an appropriate number of neighbors. For RF, LGB, and XGB, we set the number of individual learners to 500 to balance model complexity and computational cost. For the MLP model, we use a three-layer network structure with layer sizes set to 100, 50, and 10. After each layer, we add a ReLU activation function to introduce nonlinear factors. The learning rate is set to 0.001 for stable convergence. For the KNN and MLP models, we perform feature normalization to improve model performance since these two types of models are sensitive to feature scale during computation. However, for tree-based models (such as DTs, RFs, LGB, and XGB), feature normalization is usually not required because these models do not rely on the scale of features.

5.5 Hyperparameter settings

We set the dimension k of the generator's latent space in the CVAE-WGAN-gp model to 128, and the number of discriminator training rounds N_{D} is set to 5. The generator reconstruction loss parameter $\lambda_{\mathrm{G,recon}}$ is set to 3, the generator KL divergence loss hyperparameter $\lambda_{\mathrm{G,KL}}$ is set to 1, the generator adversarial loss hyperparameter $\lambda_{\mathrm{G,adv}}$ is set to 1, and the generator conditional loss hyperparameter $\lambda_{\mathrm{G,cls}}$ is set to 5. We set the generator's learning rate α_{G} to 1×10^{-4}, the discriminator's learning rate α_{D} to 5×10^{-5}, and the classifier's learning rate α_{C} to 1×10^{-5} to ensure the classifier can achieve better classification results.

5.6 Model evaluation results

To validate the oversampling performance of the CVAE-WGAN-gp model in credit assessment problems, we employ a 5-fold cross-validation to compare the predictive performance of 6 commonly used machine learning or deep learning models in the credit assessment field on 6 datasets using 7 oversampling methods, and assess the models' predictive performance using 3 evaluation metrics. Due to the large amount of data, the final experimental results are provided in Table S1 and Table S2 in the Supporting Information. In the results, "None" represents that no oversampling method, while "AR", "AP", and "F1" represent the AUC-ROC, AUC-PR, and F1-score metrics, respectively. For ease of reading, we have bolded the highest values of each model's predictions for each dataset.

To more clearly present the experimental results, the ranking of various oversampling methods under different datasets and prediction models is summarized in Table 2. When dealing with the same ranking, we adopt the commonly used average ranking method in statistics, which is to take the average of the two rankings. This approach ensures the fairness of the calculation results and reduces the impact on the statistical results. In Table 2, "Ave" represents the average ranking of different oversampling methods on the current dataset under different prediction models.The data is sorted in ascending order according to the average ranking results, and the smallest ranking indicator under each prediction model is bolded. In addition, "Ours" represents the CVAE-WGAN-gp model we proposed.

Next, we perform the Friedman test on the average ranking results of different models and oversampling methods across all datasets and present the results in Table 3. The Friedman test is a nonparametric statistical method for evaluating the performance differences of k different algorithms on n datasets. It is a ranking-based method that does not require the assumption that data conforms to a specific probability distribution. The Friedman test examines the presence of significant differences between algorithms by calculating their average rankings. If the results of the Friedman test are significant, we can conclude that at least one algorithm performs significantly different from the others. We use the chi-square distribution and p-value to measure the prediction results of the Friedman test. In the Friedman test, the degrees of freedom for the chi-square distribution are the number of algorithms k-1. Since we have a total of 7 algorithms, we use $\chi^2(6)$ to represent the results of the Friedman test's chi-square distribution. We set the significance level of the Friedman test to 0.05, meaning that when $p<0.05$, the null hypothesis is rejected, and it is concluded that at least one algorithm has significantly different performance from the others under the current prediction indicator. For ease of viewing, we have bolded the results with $p<0.05$ in Table 3. According to the results of the Friedman test in Table 3, under the condition of a significant level of $p<0.05$, we can reject the null hypothesis of equal performance for all classifier-metric combinations, except for the AUC-ROC and AUC-PR metrics of the KNN model and the AUC-ROC metric of the DT model. This indicates that compared to the AUC-ROC and AUC-PR metrics, the F1-score may be more sensitive to the performance differences of oversampling methods. In addition, compared

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

Table 2. Overview of average rankings for classifiers in each dataset, sorted by best average rank, with top ranks in bold.

Dataset	Method	KNN	DT	RF	LGB	XGB	MLP	Ave
Italy	Ours	2.7	**1.3**	**1.7**	1.8	**1.7**	**1.8**	**1.8**
	CWGAN	6.0	2.7	2.8	**1.2**	2.5	2.8	3.0
	CTGAN	4.0	4.0	3.0	3.0	4.0	2.2	3.4
	None	5.7	2.7	2.7	5.7	1.8	3.2	3.6
	B-SMOTE	3.0	5.3	6.0	4.0	6.2	5.0	4.9
	SMOTE	**1.7**	5.7	4.8	5.7	5.5	6.0	4.9
	ADASYN	5.0	6.3	7.0	6.7	6.3	7.0	6.4
Germany	Ours	2.7	**2.0**	**1.0**	**1.5**	**1.7**	2.8	**2.0**
	B-SMOTE	**2.3**	7.0	3.2	**1.5**	3.0	4.7	3.6
	SMOTE	**2.3**	2.3	4.3	5.0	6.3	**2.5**	3.8
	None	7.0	**2.0**	4.0	5.0	4.3	4.7	4.5
	ADASYN	4.8	4.7	4.5	3.7	4.3	5.0	4.5
	CWGAN	3.8	4.7	4.3	5.3	5.3	4.3	4.6
	CTGAN	5.0	5.3	6.7	6.0	3.0	4.0	5.0
Taiwan of China	Ours	4.5	**1.0**	**1.0**	2.7	3.7	**1.0**	**2.3**
	CWGAN	3.7	2.0	3.7	**1.3**	**2.2**	2.3	2.5
	None	**2.5**	3.0	4.0	2.0	2.8	4.7	3.8
	B-SMOTE	3.0	4.7	3.8	5.3	3.7	4.3	4.1
	CTGAN	6.5	7.0	5.3	4.0	4.7	3.3	5.1
	SMOTE	3.3	4.3	4.5	5.7	7.0	6.0	5.1
	ADASYN	4.5	6.0	5.7	7.0	6.0	6.3	5.9
Express	Ours	3.3	**1.0**	**1.3**	**1.2**	3.7	**1.3**	**2.0**
	B-SMOTE	3.0	2.7	4.3	4.3	3.8	3.7	3.6
	SMOTE	2.7	3.3	5.3	5.8	3.2	3.7	4.0
	CTGAN	6.7	3.0	2.7	2.0	4.0	5.3	4.0
	CWGAN	5.7	6.0	3.5	2.8	**2.2**	4.2	4.1
	ADASYN	**1.0**	6.3	4.8	5.5	3.5	4.3	4.2
	None	5.7	5.7	5.3	6.3	6.7	5.5	5.9
Card	Ours	2.3	**1.3**	**1.3**	**1.0**	**1.3**	**1.3**	**1.4**
	CWGAN	5.5	5.3	3.0	3.0	3.0	5.7	4.3
	B-SMOTE	**2.0**	6.0	4.2	4.7	4.7	5.0	4.4
	CTGAN	5.3	5.7	1.7	4.3	4.0	5.3	4.4
	None	6.8	4.0	3.7	3.8	4.3	4.0	4.4
	SMOTE	3.2	5.0	5.5	4.0	4.3	5.0	4.5
	ADASYN	2.8	7.0	6.7	6.8	6.3	2.3	5.3
Kaggle	CTGAN	**1.0**	**2.0**	2.8	**1.3**	2.3	2.7	**2.0**
	Ours	3.0	2.3	**1.5**	1.7	2.5	**1.3**	2.1
	CWGAN	4.3	4.0	4.0	3.7	**2.0**	3.0	3.5
	SMOTE	3.7	3.7	4.7	4.3	5.0	5.0	4.4
	None	5.7	5.0	4.7	5.3	3.2	4.3	4.7
	B-SMOTE	5.0	6.3	5.3	5.0	6.3	5.3	5.5
	ADASYN	5.3	4.7	5.0	5.8	6.7	7.0	5.8

Table 3. Results of the Friedman test. Bold indicates $p < 0.05$.

Model	Metric	$\chi^2(6)$	p
KNN	AR	11.532	0.073
	AP	7.885	0.247
	F1	13.006	**0.043**
DT	AR	11.071	0.086
	AP	21.000	**0.002**
	F1	15.826	**0.015**
RF	AR	14.639	**0.023**
	AP	22.072	**0.001**
	F1	14.639	**0.023**
LGB	AR	17.164	**0.009**
	AP	19.811	**0.003**
	F1	15.960	**0.014**
XGB	AR	14.828	**0.022**
	AP	16.237	**0.013**
	F1	12.770	**0.047**
MLP	AR	12.604	**0.049**
	AP	13.236	**0.039**
	F1	18.180	**0.006**

to more complex ensemble learners (such as RF, LGB, XGB) and deep learners (such as MLP), single learners (such as KNN, DT) are simpler, so they may be less sensitive when using ranking metrics to measure the performance differences of various oversampling methods. However, in most cases, we can reject the null hypothesis of equal performance for all classifier-metric combinations in the Friedman test, which means that the performance of a certain oversampling method may be significantly better than other oversampling methods.

Further analysis of the experimental results in Table 2 is conducted to determine the optimal oversampling method. According to the experimental results in Table 2, we can find that our proposed CVAE-WGAN-gp oversampling method has a significantly better average ranking in different prediction models on the other five datasets, excluding the Kaggle dataset, compared to other oversampling methods. At the same time, on the Kaggle dataset, the average ranking of the prediction models using the CVAE-WGAN-gp oversampling method is only second to that of the models using the CTGAN oversampling method.

5.7 Ablation study

5.7.1 Ablation study on loss

To investigate the impact of the WGAN-gp loss on the oversampling performance of the CVAE-WGAN-gp model in credit assessment problems, we replace the WGAN-gp loss with traditional GAN loss, WGAN loss, and ACGAN loss, to compare the advantages and disadvantages of these loss functions with the model's predictive performance when using the WGAN-gp loss function. It is worth noting that when using ACGAN loss, since the discriminator has already performed the classification task, we remove the classifier architecture

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

from the model. On the six datasets mentioned earlier, we conduct experiments using the seven prediction models mentioned in the previous text and continue to use AUC-ROC (AR), AUC-PR (AP), and F1-scores (F1) as the evaluation metrics for the models. Due to the limitation of article length, we list the average results (retaining three significant digits) of the CVAE-WGAN-gp model using different loss methods on all datasets under the three indicators of different prediction models in Table 4. To facilitate reader review, we bold the loss methods with the highest results for each indicator.

As can be observed from Table 4, in the CVAE-WGAN-gp model, employing WGAN and WGAN-gp as loss functions results in better predictive performance in most evaluation metrics. This could be because, compared to traditional GAN loss and ACGAN, WGAN provides a more stable training process for GANs, making the generated dataset closer to the distribution of the original data. In addition, compared to WGAN, WGAN-gp also achieves better predictive performance in most metrics. This may be due to the introduction of the gradient penalty term in the WGAN-gp loss, ensuring that the discriminator satisfies the Lipschitz constraint, thus further improving the training stability. According to the ablation study results, we can conclude that in the oversampling of credit assessment problems, the CVAE-WGAN-gp model using WGAN-gp loss has more significant advantages compared to traditional GAN loss, ACGAN loss, and WGAN loss.

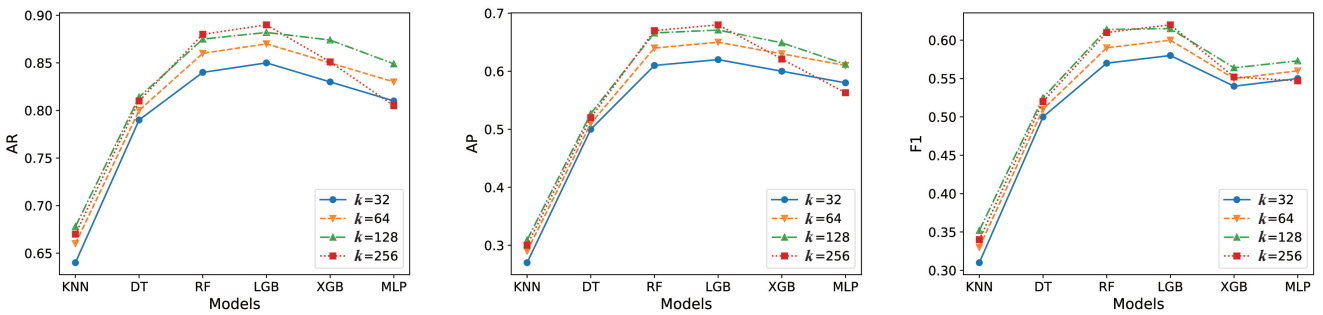
5.7.2 Ablation study on latent space dimension

In the CVAE-WGAN-gp model, the latent space dimension k is a crucial parameter. Our goal is to investigate its influence on the oversampling performance of the CVAE-WGAN-gp model in credit assessment problems. In this ablation study, we select $k = 32$, 64, 128, 256 as the latent space dimension values. We choose these values because they are all multiples of 2, which allows computers to perform calculations more efficiently. Furthermore, these values cover different orders of magnitude of latent space dimensions, helping us observe the performance variation of the model across different dimensions. Specifically, we employ three evaluation metrics, namely AUC-ROC (AR), AUCPR (AP), and F1-score (F1). The experimental results are presented as the average values across six datasets. Fig. 2 shows the changing trends of the three indicators for each prediction model under different latent space dimensions k.

Table 4. CVAE-WGAN-gp model loss comparison.

Model	Loss	AR	AP	F1
KNN	GAN	0.657	0.290	0.333
	WGAN	0.670	0.301	0.345
	ACGAN	0.662	0.295	0.358
	WGAN-gp	0.678	0.309	0.352
DT	GAN	0.793	0.512	0.509
	WGAN	0.804	0.520	0.516
	ACGAN	0.801	0.518	0.514
	WGAN-gp	0.814	0.527	0.525
RF	GAN	0.852	0.642	0.597
	WGAN	0.860	0.651	0.604
	ACGAN	0.858	0.649	0.601
	WGAN-gp	0.875	0.666	0.614
LGB	GAN	0.860	0.657	0.601
	WGAN	0.870	0.680	0.607
	ACGAN	0.865	0.658	0.603
	WGAN-gp	0.882	0.671	0.615
XGB	GAN	0.840	0.630	0.549
	WGAN	0.850	0.650	0.556
	ACGAN	0.848	0.637	0.554
	WGAN-gp	0.874	0.649	0.564
MLP	GAN	0.825	0.601	0.563
	WGAN	0.836	0.608	0.569
	ACGAN	0.832	0.605	0.575
	WGAN-gp	0.849	0.612	0.573

Based on the experimental results in Fig. 2, the prediction models with a latent space dimension of $k = 128$ perform better after CVAE-WGAN-gp oversampling compared to other dimensions. This may be due to the excessive information loss at lower dimensions and the retention of too much unnecessary information (noise) at higher dimensions. Therefore, when the latent space dimension is $k = 128$, the CVAE-WGAN-gp model can achieve better oversampling results in credit risk assessment problems.



(a) Performance of models for AR (b) Performance of models for AP (c) Performance of models for F1

Fig. 2. Comparison of CVAE-WGAN-gp model prediction results under different latent space dimensions for various metrics.

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

6 Conclusions

In credit risk assessment, the class imbalance problem is a common challenge. The proportion of non-default data to default data in credit default datasets can be extremely imbalanced, possibly reaching ratios of 50∶1 to 100∶1, which affects the model's prediction accuracy. To address this issue, this paper employs the CVAE-WGAN-gp model to oversample the minority class data in credit risk datasets, balancing the sample proportions. Experiments show that the data generated by this model is closer to the original data and outperforms other oversampling methods. Compared to no oversampling or other methods, the CVAE-WGAN-gp model oversampling can significantly improve model prediction performance, making bank credit risk assessments more accurate and efficient, reducing the risk of bad loans, and increasing revenue.

In future work, we will continue to optimize the CVAE-WGAN-gp model oversampling method to improve the performance and robustness of credit risk assessment models. At the same time, we will introduce other oversampling methods and deep learning models to compare the performance of different methods on oversampled data. We aim to improve the accuracy of credit risk assessment while minimizing the rejection of loan applicants, reducing the risk of bad loans, further increasing bank profitability, and bringing greater value to the banking industry.

Supporting information

The supporting information for this article can be found online at https://doi.org/10.52396/JUSTC-2023-0084. The supporting information includes two tables.

Acknowledgements

This work was supported by National Key R&D Program of China (2022YFA1008000), the National Natural Science Foundation of China (12571297, 12101585), the CAS Talent Introduction Program (Category B), and the Young Elite Scientist Sponsorship Program by CAST (YESS20220125).

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Kaiming Wang is a postgraduate student at the School of Management, University of Science and Technology of China. His research mainly focuses on machine learning and financial engineering.

Qing Yang is an associate professor at the School of Management, University of Science and Technology of China. She received her Ph.D. degree from Nanyang Technological University. Her current research interests include high-dimensional statistical inference and random matrix theory.

References

[1] Lappas P Z, Yannacopoulos A N. A machine learning approach combining expert knowledge with genetic algorithms in feature selection for credit risk assessment. *Applied Soft Computing*, **2021**, *107*: 107391.

[2] Arora N, Kaur P D. A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. *Applied Soft Computing*, **2020**, *86*: 105936.

[3] Shen F, Zhao X, Kou G, et al. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing*, **2021**, *98*: 106852.

[4] Johnson J M, Khoshgoftaar T M. Survey on deep learning with class imbalance. *Journal of Big Data*, **2019**, *6*: 27.

[5] Mohammed R, Rawashdeh J, Abdullah M. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In: 2020 11th International Conference on Information and Communication Systems (ICICS). IEEE, **2020**, *1*: 243–248.

[6] Shen F, Zhao X, Li Z, et al. A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Physica A: Statistical Mechanics and Its Applications*, **2019**, *526*: 121073.

[7] Sun J, Lang J, Fujita H, et al. Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. *Information Sciences*, **2018**, *425*: 76–91.

[8] Wang X, Liang G, Zhang Y, et al. Inconsistent performance of deep learning models on mammogram classification. *Journal of the American College of Radiology*, **2020**, *17* (6): 796–803.

[9] Kingma D P, Welling M. Auto-encoding variational Bayes. arXiv: 1312.6114, **2013**.

[10] Dai B, Wipf D. Diagnosing and enhancing VAE models. arXiv: 1903.05789, **2019**.

[11] Razavi A, van den Oord A, Vinyals O. Generating diverse high-fidelity images with VQ-VAE-2. In: Advances in Neural Information Processing Systems 32. Red Hook, New York: Curran Associates, Inc., **2019**, *32*: 14799–14809.

[12] Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models. In: Advances in Neural Information Processing Systems 28. Red Hook, New York: Curran Associates, Inc., **2015**, *28*: 3483–3491.

[13] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, **2022**: 10674–10685.

[14] Hara T, Mukuta Y, Harada T. Spherical image generation from a few normal-field-of-view images by considering scene symmetry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2022**, *45* (5): 6339–6353.

[15] Wang T, Wan X. T-CVAE: Transformer-based conditioned variational autoencoder for story completion. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI). Marina Del Rey, CA, USA: International Joint Conferences on Artificial Intelligence, **2019**, *1*: 5233–5239.

[16] Zhang Y, Wang Y, Zhang L, et al. Improve diverse text generation by self labeling conditional variational auto encoder. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, **2019**: 2767–2771.

[17] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. *Communications of the ACM*, **2020**, *63* (11): 139–144.

[18] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: 34th International Conference on Machine Learning (ICML 2017). Red Hook, New York: Curran Associates, Inc., **2017**: 298–321.

[19] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of Wasserstein GANs. In: Advances in Neural Information Processing

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704

Systems 30. Red Hook, New York: Curran Associates, Inc., **2017**, *30*: 5767–5777.

[20] Chan E R, Monteiro M, Kellnhofer P, et al. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, **2021**: 5795–5805.

[21] Liang G, On B W, Jeong D, et al. A text GAN framework for creative essay recommendation. *Knowledge-Based Systems*, **2021**, *232*: 107501.

[22] Tao M, Tang H, Wu S, et al. DF-GAN: Deep fusion generative adversarial networks for text-to-image synthesis. arXiv: 2008.05865, **2020**.

[23] Ruan S, Zhang Y, Zhang K, et al. DAE-GAN: Dynamic aspect-aware GAN for text-to-image synthesis. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, **2021**: 13940–13949.

[24] Bao J, Chen D, Wen F, et al. CVAE-GAN: Fine-grained image generation through asymmetric training. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, **2017**, *1*: 2745–2754.

[25] Choi E, Biswal S, Malin B, et al. Generating multi-label discrete patient records using generative adversarial networks. In: Proceedings of the 2nd Machine Learning for Healthcare Conference. PMLR, **2017**, *68*: 286–305.

[26] Baowaly M K, Lin C, Liu C, et al. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, **2019**, *26* (3): 228–241.

[27] Ding Z, Liu X, Yin M, et al. TGAN: Deep tensor generative adversarial nets for large image generation. arXiv: 1901.09953, **2019**.

[28] Xu L, Skoularidou M, Cuesta-Infante A, et al. Modeling tabular data using conditional GAN. In: Advances in Neural Information Processing Systems 32. Red Hook, New York: Curran Associates, Inc., **2019**, *32*: 7333–7343.

[29] Mottini A, Lheritier A, Acuna-Agost R. Airline passenger name record generation using generative adversarial networks. arXiv: 1807.06657, **2018**.

[30] Wang L, Wang F, Eric Gershwin M. Human autoimmune diseases: A comprehensive update. *Journal of Internal Medicine*, **2015**, *278* (4): 369–395.

[31] Bellemare M G, Danihelka I, Dabney W, et al. The Cramer distance as a solution to biased Wasserstein gradients. arXiv: 1705.10743, **2017**.

[32] Engelmann J, Lessmann S. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, **2021**, *174*: 114582.

[33] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine learning. New York: Association for Computing Machinery, **2006**: 233–240.

[34] Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS ONE*, **2015**, *10* (3): e0118432.

[35] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **2002**, *16*: 321–357.

[36] Han H, Wang W Y, Mao B H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Advances in Intelligent Computing. ICIC 2005. Berlin, Heidelberg: Springer, **2005**: 878–887.

[37] He H, Bai Y, Garcia E A, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, **2008**: 1322–1328.

[38] Lemaître G, Nogueira F, Aridas C K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, **2017**, *18*: 559–563.

CSTR: 32290.14.JUSTC-2023-0084
DOI: 10.52396/JUSTC-2023-0084
JUSTC, **2025**, 55(7): 0704