Effective Data Generation for E-banking Transactions Using Cycle-Consistent Adversarial Networks

Xiaoguo Wang and Han Zhao

College of Electronics and Information Engineering, Tongji University, Shanghai, China.
Email: 1830828@tongji.edu.cn

Abstract. In the anti-fraud research, the small amount of fraudulent transactions leads to extremely class imbalanced data. This problem becomes a bottleneck of fraud detection. In this paper, we apply the Cycle-Consistent Adversarial Networks (CycleGAN) to generate data for the minority class. Based on real e-banking transaction data, generators and discriminators are designed to generate synthetic data that meets the characteristics of e-banking transactions. Synthetic samples and real samples are mixed into the training of fraud detection model, and multiple metrics are used to verify the effect. Experimental results show that the synthetic data generated by CycleGAN can effectively improve the performance of the fraud detection model.

1. Introduction

Recently, the fraud risk of e-banking has aroused great concern in the industry. In the anti-fraud research, the small amount of fraudulent transactions leads to extremely imbalances in data class. This problem becomes a bottleneck of fraud detection. To deal with the class imbalanced problem, one way is to generate synthetic data for the minority class. How to design a suitable generation method has become a significant demand for fraud detection.

CycleGAN is able to translate a sample from domain X to domain Y by learning the mapping of samples in two unpaired datasets. Finally, it realized the mutual conversion of samples in two different data domains.

In this paper, we propose a data generation method based on CycleGAN. With the help of the transaction data from e-bank, including fraudulent and normal, it can generate sufficient and reliable data for fraud detection model training.

2. Related Work

2.1. Oversampling Methods

Oversampling methods generate synthetic examples for the minority class and add them to the training set. A simple approach is Random Oversampling, which creates new data by copying minority examples randomly. The drawback of this approach is that the exact replication of training examples can lead to overfitting since the classifier is exposed to the same information. Chawla et al. proposed SMOTE in 2002[1], which generates synthetic data along the line segment that joins minority class samples. Compared to random oversampling, SMOTE alleviates the overfitting problem. Hui Han et al. modified the SMOTE algorithm and proposed Borderline-SMOTE[2]. This method generates synthetic data only for minority samples on the borderline between majority and minority class. ADASYN[3] is proposed to generate synthetic samples adaptively. More synthetic samples are created for minority samples which have more majority samples in their K nearest neighbors compared to those have fewer majority
neighbors. ADASYN algorithm reduces the bias caused by imbalanced class, and shifts the decision boundary to those samples that are difficult to learn by the classifier adaptively. These algorithms relieve the problem of class imbalance. However, after adding generated samples, the data distribution in original dataset may be changed.

2.2. Generative Adversarial Network
The GAN algorithm can capture the actual data distribution and enrich samples for minority class based on this distribution. Ian Goodfellow proposed Generative Adversarial Network (GAN) in 2014 [4], generates synthetic images through the competition between two neural networks, a generator and a discriminator. A conditional Generative Adversarial Network (cGAN) extends the GAN model by conditioning the training procedure on external information. The cGAN model can be used to generate images of a specific class. Radford proposed DCGAN [6] in 2015, using a deep convolutional neural network to implement the generator and discriminator. DCGAN algorithm improves the resolution of generated images. Generators of the models above often face the problem of gradient vanishing. To solve this problem, WGAN [7] uses Wasserstein distance to measure the distance between two probability distributions. This method also ensures the diversity of generated samples. Recently, Zhu J.Y. proposed Cycle-Consistent Adversarial Networks (CycleGAN) [8] to solve the mutual transformation of unpaired images. In this paper, we will apply CycleGAN to the generation of e-banking transaction data.

3. CycleGAN Based Synthetic Data Generation

3.1. CycleGAN Model
The principle of CycleGAN can be summarized as transforming data from one type into another type. That is, there are two data domains, X and Y, CycleGAN hopes to convert the samples in domain X into the samples in domain Y, and the samples in domain Y into the samples in domain X. As shown in figure 1.

![Figure 1. CycleGAN](image)

CycleGAN is essentially two mirror-symmetric GANs, forming a ring network. The two GANs share two generators, each with a separate discriminator. Totally, this model has two generators G and F and two discriminators $D_x$ and $D_y$, which process the data from two different domains (X, Y) respectively.

The generator $G$ aims to learn the mapping of $X \rightarrow Y$. The inputs of $G$ are real samples from domain X, and outputs are synthetic samples similar to the samples in domain Y. The generator $F$ aims to learn the mapping of $Y \rightarrow X$. It takes samples from domain Y, and outputs synthetic data similar to samples in domain X. The two discriminators can be considered as fraud detection models. They aim to identify whether the input sample is real or synthetic, and output the probability that the input sample is real. $D_x$ discriminates samples from domain X, and $D_y$ discriminates samples from domain Y.

Both Real and synthetic samples generated by the generator are input to the corresponding discriminators. The task of the discriminator is to distinguish them and identify synthetic samples. To ensure that these synthetic samples are not recognized by the discriminator, the generator will generate
new data that is very close to the real samples in related domain. In the continuous competition between generator and discriminator, the distribution of the generated data gradually approaches the actual distribution. Finally, Nash equilibrium is reached between the generator and the discriminator.

3.2. Loss Function

CycleGAN’s loss function consists of adversarial loss and cycle consistency loss. The adversarial loss includes the reconstruction loss of the generator and the discrimination loss of the discriminator. The reconstruction loss hopes to produce data that is as similar as possible to the original data. The discrimination loss hopes that the generated data and the real data can be distinguished as much as possible. As shown in equation 1:

\[
L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log (1 - D_Y(G(x)))]
\]  

(1)

Using only the adversarial loss may cause generator G to map all samples in domain X to a same sample in domain Y, and F to map the sample in domain Y to different samples in domain X. G and F deceive the discriminator jointly, causing the loss to be invalidated. The cycle consistency loss requires \( G((F(x)) \approx x \) and \( F((G(y)) \approx y \), that is, after the sample in domain X is converted to domain Y, it can be successfully transferred back to domain X. By adding this cycle consistency loss, the model is prevented from converting all samples in domain X to the same sample in domain Y. As shown in equation 2:

\[
L_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\| F(G(x)) - x \|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\| G(F(y)) - y \|_1]
\]  

(2)

Finally, the total loss function is:

\[
L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F)
\]  

(3)

The optimization target of CycleGAN model is to optimize the total loss function. The goal of the discriminator is to maximize the total loss function, and the generator is to minimize the total loss function. As shown in equation 4:

\[
G^*, F^* = \arg \min_{G,F} \max_{D_X,D_Y} L(G, F, D_X, D_Y)
\]  

(4)

4. Experiments and Analysis

4.1. Experimental Setup

In this section, we construct the CycleGAN model. The model has 2 generators and 2 discriminators. The network architecture of two generators are identical, as shown in table 1. Similarly, as shown in table 2, the two discriminators are identical too.

| Table 1. The network architecture of generators. |
|-------------------------------------------------|
| Layer   | Output Size | Activation Function |
|---------|-------------|---------------------|
| Input   | (1,12)      | -                   |
| FC1     | (1,64)      | Sigmoid             |
| FC2     | (1,128)     | Sigmoid             |
| FC3     | (1,128)     | Sigmoid             |
| FC4     | (1,50)      | Sigmoid             |
| FC5     | (1,12)      | Tanh                |
Table 2. The network architecture of discriminators.

| Layer | Output Size | Activation Function |
|-------|-------------|---------------------|
| Input | (1,12)      | -                   |
| FC1   | (1,64)      | Leaky ReLU          |
| FC2   | (1,256)     | Leaky ReLU          |
| FC3   | (1,72)      | Leaky ReLU          |
| FC4   | (1,1)       | Softmax             |

We choose the mean square error (MSE) to replace the cross-entropy loss in CycleGAN. Adam algorithm is selected as the optimizer to update the CycleGAN network, making the model training process more stable and optimized gradually. During training, the batch size of gradient descent is set to 1, and the initial learning rate is $1e^{-5}$, which gradually decreases along with iterations. In one cycle, the generator and discriminator are trained once.

4.2. Experimental Results on Transaction Dataset

4.2.1. Dataset

We constructed the original dataset using historical transactions of personal e-banking from our cooperative bank. The time span is from September 2018 to November 2018. There are 15,78611 transactions in the original dataset, of which 1,758,591 are normal and 20 are fraudulent. Divide the training set and test set on November 1, 2018. The training set has 1034097 transactions, including 9 frauds. The test set has 544,514 transactions, including 11 frauds.

In order to verify the validity of CycleGAN model in generating data, we constructed enhanced datasets. Based on the original dataset, enhanced datasets add the fraud transactions generated by CycleGAN model to training set, and test set remains unchanged. Details of the original dataset and one of enhanced datasets are shown in table 3.

Table 3. Details of experimental datasets.

| Dataset   | Total   | Number of normal | Number of fraud |
|-----------|---------|------------------|-----------------|
| Original  | 1578611 | 1578591          | 20              |
| Original Train | 1034097 | 1034088          | 9               |
| Original Test | 544514  | 544503           | 11              |
| Enhanced  | 1579008 | 1578591          | 417             |
| Enhanced Train | 1034494 | 1034088          | 406             |
| Enhanced Test  | 544514  | 544503           | 11              |

4.2.2. Metrics

In the original dataset used in the experiment, normal transactions accounted for more than 99.9%. Even if all the data are judged as normal transactions, the accuracy is close to 100%. Using only Accuracy cannot effectively measure the effectiveness of the model. Therefore, based on the confusion matrix, metrics such as Sensitivity, Recall, Precision, G-mean and F1 Score were selected to evaluate the model performance and verify the effectiveness of the data generated by CycleGAN.

(1) OA:

$$OA = (TN + TP) \times (TP + FP + FN + TN)^{-1}$$  \hspace{1cm} (5)

(2) Specificity:

$$Specificity = TN \times (TN + FP)^{-1}$$  \hspace{1cm} (6)

(3) Recall:

$$Recall = TP \times (TP + FN)^{-1}$$  \hspace{1cm} (7)

(4) Precision:
\[ Precision = TP \times (TP + FP)^{-1} \] (8)

(5) G-mean:
\[ G - mean = (Recall \times Specificity)^{1/2} \] (9)

(6) F1 Score:
\[ F1\, Score = (1+\beta^2) \times Recall \times Precision \frac{\beta^2 \times Recall + Precision}{\beta^2} \] (10)

4.2.3. Analysis of results
In this paper, we used a fraud detection model based on adacost to verify the validity of the CycleGAN model. The original dataset and the enhanced data set were respectively put into the fraud detection model. The experimental results in table 4 show the fraud detection performance of the models trained on different datasets.

| Table 4. Test results on different datasets |
|--------------------------------------------|
| Metrics          | Original Dataset | Enhanced Dataset |
|------------------|------------------|------------------|
| OA               | 0.98967          | 0.98307          |
| Specificity      | 0.98967          | 0.98307          |
| G-mean           | 0.59990          | 0.73227          |
| Recall           | 0.36364          | 0.54545          |
| Precision        | 0.00057          | 0.00052          |
| F1 Score         | 0.00114          | 0.00105          |

After adding data generated by the CycleGAN model, Sensitivity, g-mean and Recall were significantly improved, while other metrics remained stable. The synthetic data generated by the CycleGAN model can effectively improve the performance of the fraud detection model.

5. Conclusion
Aiming at the class imbalanced problem caused by insufficient fraud transactions of e-banking, this paper proposes a data generation algorithm based on CycleGAN. The algorithm learns the mapping relationship between normal transactions and fraudulent transactions through continuous confrontation between generators and discriminators, and realizes mutual conversion of data. After training, generators can generate synthetic samples with high similarity to the real data. Synthetic samples and real samples are mixed into the fraud detection model, and multiple metrics are used to verify the effect. The experimental results show that the synthetic data generated by CycleGAN improves the performance of the fraud detection model. Models trained with the enhanced dataset are better than without mixed synthetic data in terms of Sensitivity, G-mean, and Recall. The next stage will further improve the quality of the synthetic data and explore the application of CycleGAN to multiple classes of imbalance problems.

6. References
[1] Chawla N V, Bowyer K W, Hall L O and Kegelmeyer W P 2002 Smote: synthetic minority oversampling technique. Journal of Artificial Intelligence Research, 16(1), 321-357
[2] Hui Han, Wang W Y and Mao B H 2005 Proceedings of the 2005 international conference on Advances in Intelligent Computing vol 1 (Berlin: Spinger)
[3] He H, Bai Y, Garcia E A and Li S 2008 IEEE International Joint Conference on IEEE (IEEEWorld Congress on Computational Intelligence)
[4] Dinh L, Sohl-Dickstein J and Bengio S 2016 Density estimation using real nvp arXiv:1605.08803
[5] Mirza M and Osindero S 2018 Conditional Generative Adversarial Nets arXiv:1411.1784
[6] Radford A, Metz L and Chintala S 2015 Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks \textit{arXiv:1511.06434} \\
[7] Arjovsky M, Chintala S and Bottou L 2017 \textit{34th International Conference on Machine Learning (ICML)} vol 1 p298-321 \\
[8] Zhu J, Park T, Isola P and Efros A A 2017 \textit{IEEE International Conference on Computer Vision (ICCV)} p 2242-51