IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 July 2023, accepted 26 July 2023, date of publication 7 August 2023, date of current version 11 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3302339

RESEARCH ARTICLE

Credit Card Fraud Detection Based on Improved Variational Autoencoder Generative Adversarial Network

YUANMING DING, WEI KANG, JIANXIN FENG, BO PENG, AND ANNA YANG
Communication and Network Key Laboratory, Dalian University, Dalian 116622, China

Corresponding author: Yuanming Ding (dingyuanming@dlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61901079.

ABSTRACT The rapid spread of mobile banking and e-commerce has coincided with a dramatic increase in fraudulent online payments in recent years. Although machine learning and deep learning are widely used in credit card fraud detection, the typical credit card transaction data set is unbalanced, and the fraud data is much less than the normal transaction data, limiting the effectiveness of traditional binary classification algorithms. To overcome this issue, researchers oversample minority class data and utilize ensemble learning classification algorithms. However, oversampling still has disadvantages. Hence, we improve the generator part of the Variational Autoencoder Generative Adversarial Network (VAEGAN) and propose a new oversampling method that generates convincing and diverse minority class data. The training set is enhanced by generating minority class fraud data to train the ensemble learning classification model. The method is tested on an open credit card dataset, with the experimental results demonstrating that the oversampling method utilizing the improved VAEGAN is superior to the oversampling method of Generative Adversarial Network (GAN), Variational Autoencoder (VAE), and Synthetic Minority Oversampling Technique (SMOTE) in terms of Precision, F1_score, and other indicators. The oversampling method based on the improved VAEGAN effectively deals with the classification problem of imbalanced data.

INDEX TERMS Credit card fraud, ensemble learning, variational autoencoder generative adversarial network, oversampling.

I. INTRODUCTION

Imbalanced data refers to the situation where the number of samples of different classes in the data set varies significantly. For example, the dataset is imbalanced in a binary classification problem if the number of positive samples is much less than that of negative samples. The class with a large number of samples is usually called the majority class, and the class with a small number of samples is the minority class [1]. In practical applications, unbalanced data sets can appear in various fields, such as medicine, natural language processing, image recognition, industrial defect detection, and finance [2]. In the financial field, the incidence of fraudulent transactions is very low, so the number of positive samples is much less than the number of negative samples in fraud detection datasets. In an unbalanced dataset, due to the small number of positive samples, the model may be more inclined to predict the negative class while ignoring the positive class, decreasing the model's classification performance, especially for the minority class [3]. At the same time, the model's generalization performance declines, and its performance evaluation deviates [4], [5].

The credit card fraud detection problem studied in this paper belongs to the classification problem of imbalanced data. Credit card fraud detection refers to identifying and preventing fraudulent behavior in credit card transactions based on relevant characteristic variables in the customer's past transaction records. Although fraudulent transactions are a minority, the losses caused by misjudging fraudulent transactions are often greater than those caused by misjudging

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

non-fraudulent transactions [6]. Currently, the main solutions for credit card fraud detection are as follows, supervised machine learning algorithm [6], [7], [8], semi-supervised machine learning algorithm [9], [10], and unsupervised machine learning algorithm [8]. There are mainly two strategies to solve the class imbalance problem in credit card fraud detection. On the data processing level, oversampling and undersampling techniques are used to balance the original data [11], [12], and on the algorithm level, ensemble learning and cost Sensitive learning further improve the effectiveness of classifiers [8], [13]. However, despite many studies exist, these present various problems and require further refinement and improvement.

This paper uses the most widely used supervised learning method to detect credit card transaction data [14], compares and analyzes the classification performance of five classification algorithms, and selects the classification model with the best precision and F1 score performance [15]. In order to solve the negative impact caused by data imbalance, researchers use undersampling or oversampling methods to improve the results of fraud detection classification [16], [17], [18], [19]. Undersampling reduces the number of majority class samples, removing some useful hidden information and thus affecting the model's classification performance. Typically, researchers adopt the oversampling method [20], [21].

This paper utilizes the improved Variational Autoencoder Generative Adversarial Network (VAEGAN) deep learning method to generate positive data. Specifically, the minority class data in the original training set is used as the training set of the deep learning method, which is then used to generate false minority class data. After that, the generated fake data and the original training set are combined to form an enhanced training set. Experimental evaluations demonstrate that classification models trained on the augmented training set attain an improved classification performance compared to models solely trained on the original training set. Although our framework is developed for credit card fraud detection, it is quite general and can be easily extended to other application domains.

The remainder of this paper is structured as follows. Section II systematically introduces the related work on credit fraud, and Section III presents some basic theoretical knowledge of the model. Section IV elaborates on the research methodology, and section V discusses the relevant experimental content. Finally, Section VI summarizes our objectives and findings.

II. RELATED WORK

Credit card fraud detection has always been a concern for many researchers. Supervised learning methods based on machine learning and deep learning are on credit card fraud detection. To improve the impact of the imbalance of credit card data on the classification results, the researchers have proposed two solutions. One is to improve the classifier and select a better-performance classification mode, and the other is to deal with the imbalanced data. In [22], the authors combined manual and automatic classification, compared different machine learning algorithms, and used data mining techniques to solve fraud detection and similar problems. In [23], eight machine learning algorithms were compared to credit card fraud detection. The Logistic Regression (LR), C5.0 decision tree algorithm, and Support Vector Machine (SVM) were selected as the final classification method. In [13], researchers compared two random forests with different base classifiers and analyzed their credit card fraud detection performance. Other solutions applied artificial neural networks to credit card fraud detection. For instance, Asha RB [24] utilizes various machine learning algorithms and artificial neural network (ANN) to predict the occurrence of fraud. The experimental results show that it provides higher accuracy than unsupervised learning. The work of [25] formulated the fraud detection problem as a sequence classification task and used a long short-term memory (LSTM) network to incorporate transaction sequences.

For the data imbalance problem, studies have shown that oversampling and undersampling methods perform well for ensemble classification models such as AdaBoost, XGBoost, and Random Forest [26]. Indeed, [27] proposed an All K-Nearest Neighbors (AllKNN) undersampling technique, which, although it improved the classification performance on some indicators, lost important information in the data, leading to flawed Trained classifiers [28]. Currently, oversampling has become the main data preprocessing method to deal with imbalanced data, with [29] oversampling the minority class using SMOTE. Besides, Majzoub et al. [30] proposed a Hybrid Cluster Affinity Boundary Line SMOTE (HCAB-SMOTE) oversampling technique that improves SMOTE. Recently, deep learning models have also been applied with data oversampling. Fiore et al. [31] expanded the credit card fraud data using a GAN to generate virtual fraud samples. The results show that this method is superior to the SMOTE oversampling method. Additionally, Tingfei et al. [32] proposed using a VAE model as an oversampling module to augment the original training data with generated data. The experimental results show that the VAE oversampling model slightly improved over the GAN network.

This work employs the VAEGAN model as an oversampling module to rebalance the training set by generating fake minority class data and injecting the generated data into the original training set. At the same time, we improved and optimized the VAEGAN model, enhancing its expressive ability to output more realistic and diverse data.

III. RELATED THEORY

A. SMOTE

SMOTE is an algorithm dealing with class-imbalanced datasets, which balances the class distribution in the dataset by generating some synthetic samples. Specifically, the SMOTE algorithm selects some minority samples, selects several nearest neighbor samples for each sample, and generates new samples through random interpolation. By changing the interpolation ratio, the SMOTE algorithm adjusts the influence of generating synthetic samples on the training set. This process is represented by the following formula:

$$x' = x + rand\,(0, 1) * |a - b| \tag{1}$$

B. GAN

GAN is a deep learning model comprising two neural networks: a generator (G) and a discriminator (D), as depicted in Figure 1. The generator receives a random noise vector z as input and generates false data x′ through a series of transformations. The discriminator (D) receives real data x and fake data x′ and tries to distinguish which data is real or fake.

GAN's innovation lies in introducing the confrontational training concept, enabling the generator to generate more realistic data. Equation 2, 3 presents the objective function of the GAN network:

$$\max_{D} E_{x\sim p_d(x)}\,[D\,(x)] + E_{z\sim p_z(z)}\,[1 - D\,(G\,(z))] \tag{2}$$

$$\min_{G} E_{z\sim p_z(z)}\,[1 - D\,(G\,(z))] \tag{3}$$

where $P_d(x)$ represents the distribution of real data, $P_z(z)$ is the distribution of noise, $G\,(z)$ denotes the generated sample, and $D\,(x)$ is the output probability.

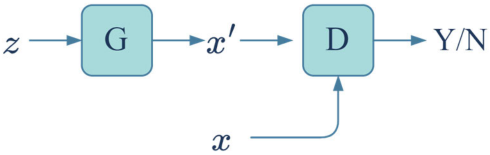


FIGURE 1. Input random noise z to train the generator G, whose output x′ is combined with the original training set x to train the discriminator.

C. VAE

The VAE model is a generative model that generates new data by learning the distribution of latent variables. As illustrated in Figure 2, it comprises the encoder (E) and decoder (D). The encoder maps the input data x to a probability distribution z in the latent space, and the decoder samples and reconstructs the original data x′ from the latent space. The training objective of the VAE model is to minimize the reconstruction error and the KL divergence of the latent variables. The VAE model can learn a continuous representation in the latent space, which can be reasoned through variational inference, presenting a certain interpretability. However, the data quality generated by VAE is inferior to generative models such as GAN. The encoding principle of the encoder is presented in Equation 4:

$$z = \sigma\,(x) * N\,(0, 1) + \mu\,(x) \tag{4}$$

where σ(x) and μ(x) are the standard deviation and mean of the real data, respectively. This way, the encoder's distribution can be directly decoded and generate the data.

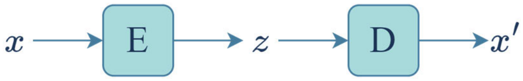


FIGURE 2. Input raw data x to train encoder E, whose output z is used to train decoder D and generate data x′.

D. VAEGAN

VAEGAN is a generative model that combines the advantages of VAE and GAN to learn the latent space representation and the distribution of the generated samples through two-stage training. VAEGAN uses the discriminator of GAN to assist training, and the discriminant result is employed as the loss function of VAEGAN. Compared with VAE and GAN networks, VAEGAN has three main advantages: the learned latent space representation is more discriminative and can better distinguish different samples; the generator learns the advantages of VAE and GAN simultaneously and can generate more Realistic and diverse samples; mapping latent vectors to interpretable feature spaces helps to analyze and understand the model's representation ability. The advantages of the VAEGAN model afford a wider application prospect in the image and video generation.

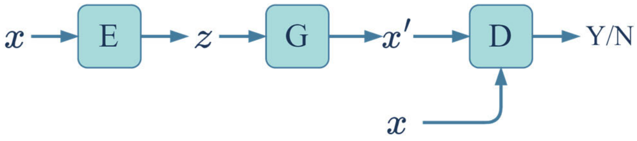


FIGURE 3. Input the original data x to train the encoder E. Its output z is used to train the generator G in the generative adversarial network. The generated data x′ is merged with the original training set x, and then the discriminator D is trained using the enhanced training set.

As illustrated in Figure 4, the real data is inputted into the VAEGAN's encoder, which encodes it into mean and variance codes. The mean and variance codes are then reparameterized to generate latent codes. VAEGAN's decoder generates fake data by decoding the latent codes. Finally, the real and generated fake data are fed into the VAEGAN's discriminator to determine whether the input data is real or fake.

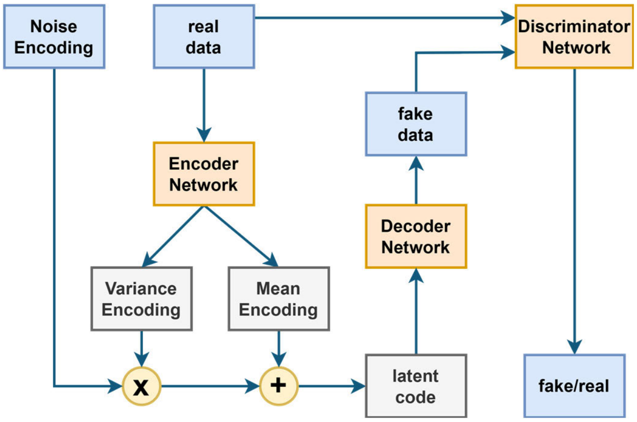


FIGURE 4. Flowchart of VAEGAN: Input real data to generate false data and judge the authenticity of the data.

IV. RESEARCH METHODOLOGY

A. CREDIT CARD FRAUD DETECTION FRAMEWORK

This paper studies four oversampling models: SMOTE, GAN, VAE, and VAEGAN, and improves the VAEGAN model. Five classification models, i.e., Logistic, decision tree, random forest, neural network, and XGBoost, are compared and analyzed, considering their effect on credit card fraud detection. The credit card fraud detection framework studied in this article comprises two parts. In the first part, five classification models are trained separately without balancing the credit card data while maintaining the original data distribution ratio. The classification models suitable for credit card fraud detection are selected based on some evaluation metrics. In the second part, the five oversampling models studied are used to balance the original training set, and the most effective oversampling method is selected to improve the classification effectiveness. The credit card fraud detection framework can be summarized as follows:

1) Train Logistic, decision tree, random forest, neural network, and XGBoost models, and perform cross-validation and grid search. The classification model C with the best classification effect is selected as the baseline method.
2) Screen all fraudulent samples from the original training set T to form a set F.
3) Use SMOTE, GAN, VAE, VAEGAN, and the improved VAEGAN models to oversample F and increase the number and diversity of fraudulent samples. Then generate a new synthetic instance F′.
4) Construct an enhanced training set T′, and merge the synthetic sample F′ generated by the oversampling method with the original training set T.
5) Retrain the classification model C on the enhanced training set T′.
6) The difference in the performance indicators between the original classification model C and the enhanced classification model C′ is compared on the independent test set S. The improvement effect of different oversampling methods and enhanced training sets on the effectiveness of the classification model is verified.

Through the above experimental process, the impact of oversampling methods and enhanced training sets on credit card fraud detection can be scientifically evaluated, improving the classification model's effectiveness and robustness, thus providing a reliable and effective solution for practical applications.

B. DATASET DESCRIPTION

This study exploits the credit card fraud detection data released on the Kaggle platform, which contains European cardholders' credit card transaction data within two days in September 2013. The data set contains 30 features, including 28 numerical features V1, V2…V28 that have undergone PCA dimensionality reduction, and two features, "Time" and "Amount", that have not undergone PCA conversion. The target variable of the data set is "Class", which is used to mark whether the transaction is a fraudulent transaction, where 0 indicates a normal transaction, and 1 is a fraudulent transaction. The dataset consists of 284,807 transactions, of which only 492 are fraudulent transactions (0.17% of the total), and the remaining 284,315 transactions are normal transactions. This is a typical imbalanced classification problem. Figure 5 shows the class distribution of fraudulent and non-fraudulent transactions in the credit card fraud dataset, revealing an extremely imbalanced distribution between normal and fraudulent transactions in the credit card dataset.

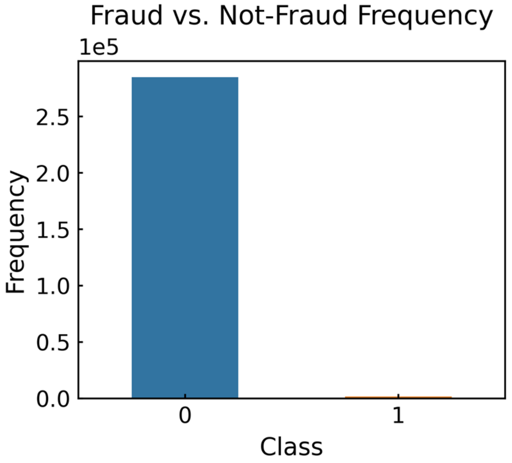


FIGURE 5. The credit card data distribution, 1 is fraudulent data, 0 is normal data.

In order to improve the performance and efficiency of the fraud detection algorithm, the characteristics of the transaction amount are normalized to avoid a large impact on the model weight. This paper adopts the normalization method based on the median and interquartile range, and the normalization rule is as follows:

$$x_i' = \frac{x_i - \text{median}}{\text{IQR}} \tag{5}$$

where x_i represents a certain sample value, the median represents the sample's median, and IQR is the interquartile range of the sample.

Finally, the data set is divided into a training set, accounting for 70% of the total samples, including 199,365 transaction data, of which 337 involve fraud data (0.169%). The remaining data are used as a test set, accounting for 30% of the total samples, including 85442 transaction data, of which 155 involve fraud data (0.181% incidence rate).

C. FRAUD DETECTION CLASSIFICATION ALGORITHMS

We evaluated five fraud detection and classification algorithms, including machine learning, ensemble learning, and neural networks, and selected the optimal classification algorithm as the baseline model for credit card fraud detection through experimental comparative analysis.

1) XGBOOST

XGBoost (eXtreme Gradient Boosting) is an algorithm based on GBDT. It builds multiple decision trees iteratively, optimizing each iteration's loss function while using gradient-boosting techniques to speed up training and improve accuracy. Equation 6 is the objective function of XGBoost:

$$L(\Phi) = \sum_i l\left(y_i, \hat{y}_i\right) + \sum_k \Omega\left(f_k\right) \tag{6}$$

where L(Φ) is the expression on the linear space, i is the i-th sample, k is the k-th tree, and $\hat{y}_i$ is the predicted value of the i-th sample x_i.

2) OTHER CLASSIFICATION ALGORITHMS

Logistic Regression is a classic machine learning algorithm for binary or multi-class classification problems. It combines the input features linearly and then uses the sigmoid function to map the result into a probability output between 0 and 1.

A Decision Tree is a classification model based on a tree structure created by selecting the best features for node splitting. During the test, it starts from the root node, traverses in order according to the feature value, and finally reaches the leaf node. The category of the leaf node is the prediction result.

Random Forest is an ensemble learning algorithm based on decision trees, which reduces overfitting and improves prediction accuracy by building multiple decision trees on different random samples and features. The random forest votes through all decision trees at test time to determine the final prediction.

A Neural Network is a machine learning model that imitates the structure and function of the human nervous system. It comprises multiple layers of neuron nodes, and weights connect each layer. The neural network passes the input signal to the output layer through forward propagation and then uses the backpropagation algorithm to adjust the weights to realize the nonlinear transformation of the input and predict the output.

D. IMPROVED VAEGAN OVERSAMPLING METHOD

To achieve better oversampling results, we tested adding extra encoders or increasing the number of layers. The credit card fraud data only has 30 dimensions, and the data features are not very complex. Using two encoders can improve the model's representation ability, as each encoder can learn different feature representations. However, using a deeper encoder leads to overfitting and result in a decrease in sampling effectiveness.

The original VAEGAN model has only one encoder, which cannot easily capture the data's complex structure and multilevel features, resulting in limited model and generalization performance. The insufficient representation ability of the latent space may lead to the lack of diversity and realism of the generated samples while affecting the model's generalization performance. Additionally, an encoder limits the model's scalability, making handling more complex data and tasks challenging. Therefore, it is necessary to improve the VAEGAN model, increase the number of encoders, improve the expressiveness of the latent space, and enhance the model's scalability.

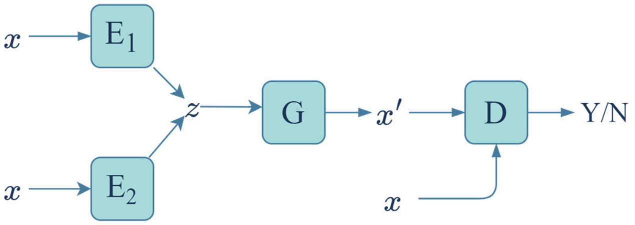


FIGURE 6. Input the original data x to train the encoder E1 and the encoder E2 respectively, and the fusion output z is used to train the generator G in the Generative Adversarial Networks. The generated data x′ is merged with the original training set x, and the combined enhanced training set is used to train the discriminator D.

Spurred by the above problems, this paper improves the VAEGAN model by adding an encoder to the VAE part of the original VAEGAN model (Figure 6). Input the fraud data into encoder E1 and encoder E2 separately. Both E1 and E2 can encode the input real data into mean and variance codes respectively. By merging the mean and variance codes from both encoders, we generate the latent code. Then, the decoder generates fake data by decoding the latent code.

The key step to realizing the above idea is fusing the mean and variance encoded by the two encoders. Thus, the results of the two encodings in VAEGAN are fused by multiplying two normal distribution probability density functions. Assuming that the probability density functions of two normal distributions are distributed as:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \tag{7}$$

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \tag{8}$$

Multiplying the two gives:

$$h(x) = A \cdot \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \tag{9}$$

The value of A is:

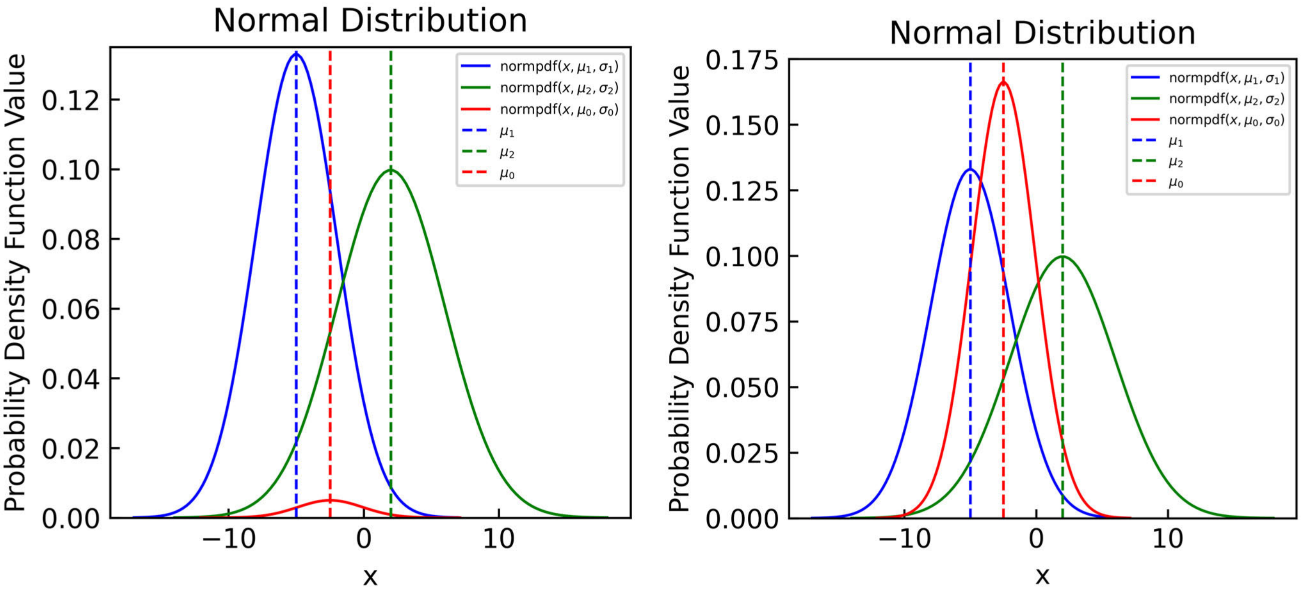
$$A = \frac{e^{-\frac{(\mu_1-\mu_2)^2}{2\left(\sigma_1^2+\sigma_2^2\right)}}}{\sqrt{2\pi\left(\sigma_1^2+\sigma_2^2\right)}} \tag{10}$$

The value of μ_0 is:

$$\mu_0 = \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{11}$$

The value of σ_0^2 is:

$$\sigma_0^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{12}$$



(a)

(b)

FIGURE 7. (a) : The result of multiplying normal distribution $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ is distribution $N(\mu_0, \sigma_0^2)$. (b) After the normal distribution $N(\mu_1, \sigma_1^2)$ is multiplied by $N(\mu_2, \sigma_2^2)$, the scaling factor is deleted, and the result is $N(\mu_0, \sigma_0^2)$.

where $h(x)$ is the result of a normal distribution $N(\mu_0, \sigma_0^2)$ multiplied by the scaling factor A, μ_0 is the mean of a normal distribution, and σ_0^2 is the Variance of a normal distribution. We conclude that multiplying the probability density functions of two normal distributions that obey $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ is equivalent to the normal distribution $N(\mu_0, \sigma_0^2)$ multiplied by the scaling factor A.

Therefore, the product of two Gaussian distributions is a scaled Gaussian distribution. However, scaling factor A changes the density value corresponding to the value of each selected random variable and does not change the expected sum after the product Variance, i.e., the distribution relationship after the product is unaffected by the scaling factor. When fusing the mean and variance encoded by the two encoders, we delete the scale factor A to ensure that the distribution after fusion is also a normal distribution. use $h'(x)$ replace $h(x)$.

$$h'(x) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \tag{13}$$

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. BASELINE MODEL

To screen out the optimal classification model, this paper uses the Logistic, decision tree, random forest, neural network, and XGBoost algorithms to detect credit card fraud. We performed cross-validation, grid search, and other strategies on the five classification models to ensure the generalization ability and performance of the selected models. After synthesizing the five indicators, we determined the final parameter settings per model (Table 1). Table 2 records the classification performance indicators of the five models, revealing that the XGBoost model has obvious advantages in the Recall and F1_score indicators. To evaluate the model's effectiveness more comprehensively, the PR curve is drawn based on different classifiers (Figure 8), and the ROC curve is illustrated in Figure 9. Combining Figures 8 and 9 reveals that the classification results of the XGBoost classifier are better than the other classifiers.

Finally, we choose the XGBoost classification model as the baseline method for fraud detection. This baseline model will be a reference model for subsequent model performance improvements.

TABLE 1. Model parameters.

Model	Parameters
Logisitic	C = 0.01, max_iter=100, penalty = 'l2'

DT	max_depth=2, min_samples_split=2
	min_samples_leaf=3
	criterion=entropy, max_leaf_nodes= None

RF	max_depth=3, min_samples_split=2, min_samples_leaf=1
	criterion=entropy, max_leaf_nodes=None
	n_estimators= 60, max_features= sqrt

XGBoost	max_depth=4, min_child_weight=None,Subsample=None
	Gamma=None, colsample_bytree=None
	n_estimators=100, learning_rate=None

DNN	learning_rate=0.001, optimizer=Adam
	Lossfunction=sparse_categorical_crossentropy
	batchsize=64, epochs=20, activation function=Relu

TABLE 2. Base model classification indicator results.

Model	Precision	Recall	F1_score	Specificity
Logisitic	0.97778	0.44898	0.61538	0.99998
DT	0.89286	0.51020	0.64935	0.99989
RF	0.98214	0.56122	0.71428	0.99998
XGBoost	0.95833	0.70408	0.81176	0.99995
DNN	0.98276	0.58163	0.73077	0.99998

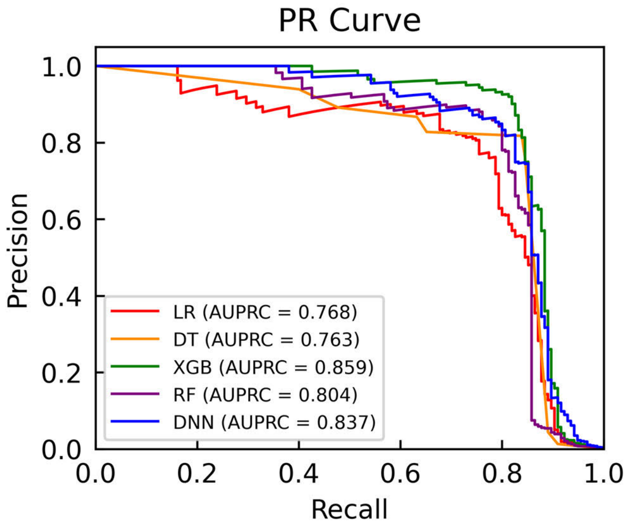


FIGURE 8. Basic model classification precision-recall curve.

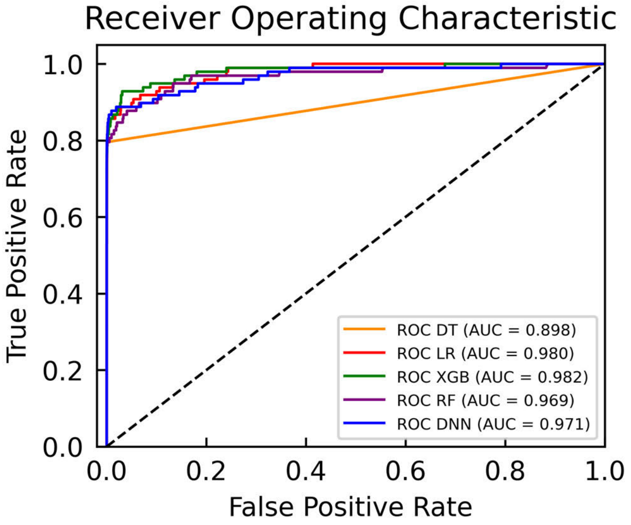


FIGURE 9. Basic model classification receiver operating characteristic curve.

B. ANALYSIS OF OVERSAMPLING METHODS

This article compares and analyzes five oversampling methods, with Table 3 reporting the GAN and VAE parameters. The generator and discriminator in GAN are three-layer networks and the encoder and decoder in VAE are two-layer networks. Table 4 presents the parameters of VAEGAN and improved VAEGAN. Besides, VAE in VAEGAN uses a two-layer network, GAN uses a three-layer network, VAE in improved VAEGAN also uses a two-layer network, and GAN uses a three-layer network.

TABLE 3. Model parameters.

Parameters	Model	
	GAN	VAE
Optimizer	Adam	Adam
Batchsize	6	6
Epochs	3000	3000
Activation function	Relu	Relu
Learning rate	0.001	0.001
Loss function	Least Squares GAN	MSE+KLD

TABLE 4. Model parameters.

Parameters	Model	
	VAEGAN	Improved VAEGAN
Optimizer	Adam	Adam
Batchsize	32	32
Epochs	3000	3000
Activation function	Relu	Relu
Learning rate	0.0003(betas = (0.5, 0.999))	0.0003(betas = (0.5, 0.999))
Loss function	Least Squares GAN+KLD	Least Squares GAN+KLD

In the oversampling experiment, the false fraud samples were synthesized based on 337 fraud samples of the original training set. The synthetic fraud samples to the real fraud samples in the original training set have a ratio of 0.25, 0.5, 1, 2, 3, 4, 8, 10, 20, and 100. The synthetic samples and the original training set form an enhanced training set for model training.

Finally, five tests were conducted on the same test set, and then the average value of Precision, Recall, F1_socre, Specificity, and AUC was taken as the final experimental result.

By combining Figure 10 and Table 5, we conclude that the SMOTE and GAN oversampling methods negatively impact the model's precision, and the overall trend decreases as the number of generated samples increases. The VAE, VAEGAN, and improved VAEGAN oversampling methods significantly improve classification precision. Under the experimental expansion ratio, the precision of these three oversampling methods is higher than that of the baseline model. The improved VAEGAN model improves the precision more than the other models at each scale. The precision of the improved VAEGAN model is 0.0281 higher than the baseline model, and VAE and VAEGAN are 0.0184 and 0.0159 higher, respectively.

By combining Figure 11 and Table 5, we conclude that the VAE, VAEGAN, and improved VAEGAN methods can improve the F1 value more significantly, and the improved VAEGAN has the best effect, which is far better than the other

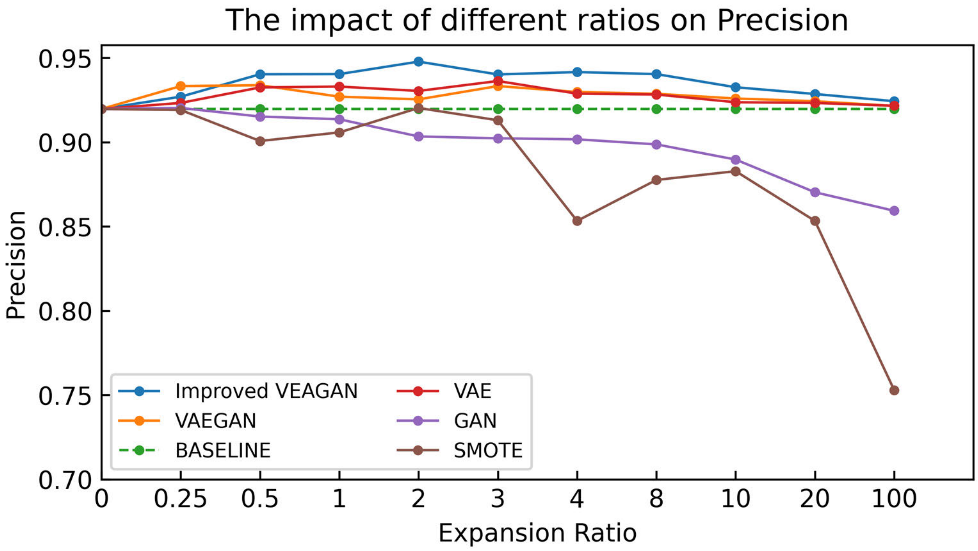


FIGURE 10. Oversampling model and baseline model classification precision under different training set expansion ratios.

TABLE 5. Precision and F1_score as the number Ng of generated examples vary.

N_g	Precision					F1_score				
	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN
0	0.9197	0.9197	0.9197	0.9197	0.9197	0.8630	0.8630	0.8630	0.8630	0.8630
84	0.9191	0.9203	0.9233	0.9333	0.9270	0.8591	0.8668	0.8687	0.8689	0.8775
168	0.9010	0.9152	0.9325	0.9338	0.9403	0.8581	0.8703	0.8763	0.8728	0.8827
337	0.9058	0.9136	0.9330	0.9270	0.9404	0.8532	0.8658	0.8798	0.8698	0.8805
674	0.9203	0.9034	0.9304	0.9254	0.9478	0.8669	0.8673	0.8704	0.8681	0.8789
1011	0.9130	0.9023	0.9363	0.9333	0.9424	0.8601	0.8623	0.8678	0.8689	0.8779
1348	0.8533	0.9017	0.9288	0.9298	0.9416	0.8390	0.8546	0.8750	0.8765	0.8836
2696	0.8776	0.8987	0.9283	0.9287	0.9404	0.8543	0.8527	0.8737	0.8754	0.8823
3370	0.8828	0.8897	0.9237	0.9259	0.9326	0.8533	0.8523	0.8716	0.8747	0.8819
6740	0.8533	0.8703	0.9233	0.9243	0.9286	0.8390	0.8504	0.8678	0.8698	0.8796
33700	0.7529	0.8593	0.9216	0.9215	0.9243	0.7964	0.8399	0.8673	0.8677	0.8793

oversampling models in all proportions. Indeed, the F1 value increased from 0.863 to 0.884. The VAE and VAEGAN methods performed similarly regarding the F1 value. Additionally, the GAN method promotes the F1 value when the expansion ratio is less than three, and the effect is not as good as the baseline model when the expansion ratio is greater than three. The performance of the SMOTE method is generally inferior to the baseline model in terms of the F1 value. Figure 12 and Table 6 highlight that the oversampling method has improved recall, and the overall trend is wavy. In some cases, SMOTE and GAN models are slightly lower than the recall of the baseline model. For instance, improving VAEGAN is more stable and prominent. Compared with the baseline method, the Recall of improved VAEGAN, VAE, VAEGAN, and GAN methods improved by 0.0256, 0.0194, 0.0161, and 0.0211, respectively.

The experimental results on Specificity and AUC are reported in Tables 6 and 7, respectively. The results suggest the improved VAEGAN oversampling method achieves the best effect on Specificity. Moreover, the improved VAEGAN does not perform equally well to SMOTE on the classification indicator AUC but has improved performance compared to other oversampling and baseline models.

In reference [32], Deep Neural Network (DNN) was used as the classification algorithm and VAE was used as the oversampling algorithm for credit card fraud detection. We compared our experimental results with those of Tingfei et al. The results show that our method achieved a higher precision at all augmentation ratios with an increase of 0.0203 in precision. In terms of F1-score, our method has a significant advantage at most augmentation ratios. However, in some cases, the recall of our method is slightly lower than that of Tingfei et al. F1-score combines precision and recall, so it can more comprehensively evaluate its performance. The proposed method in this paper is more suitable for imbalanced data classification problems.

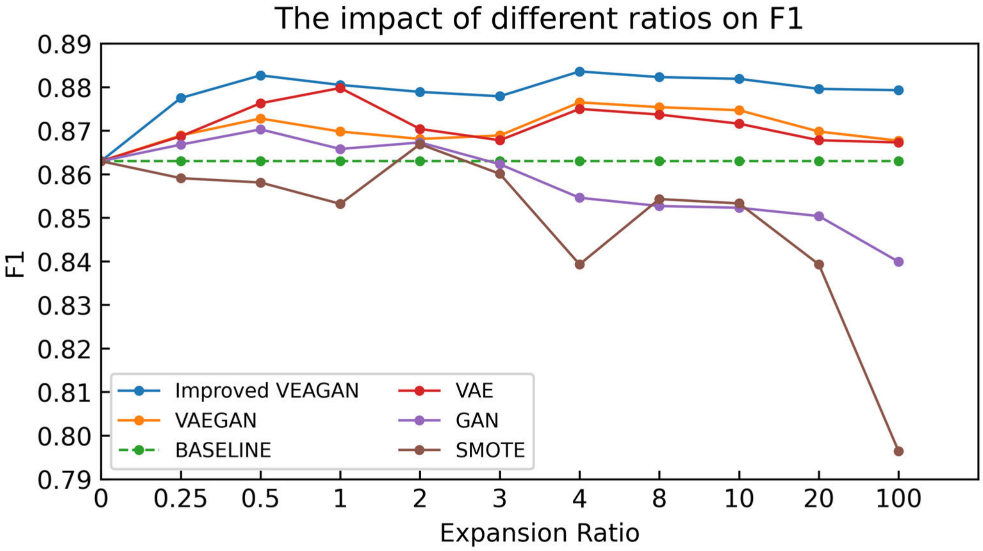


FIGURE 11. Oversampling model and baseline model classification F1_score under different training set expansion ratios.

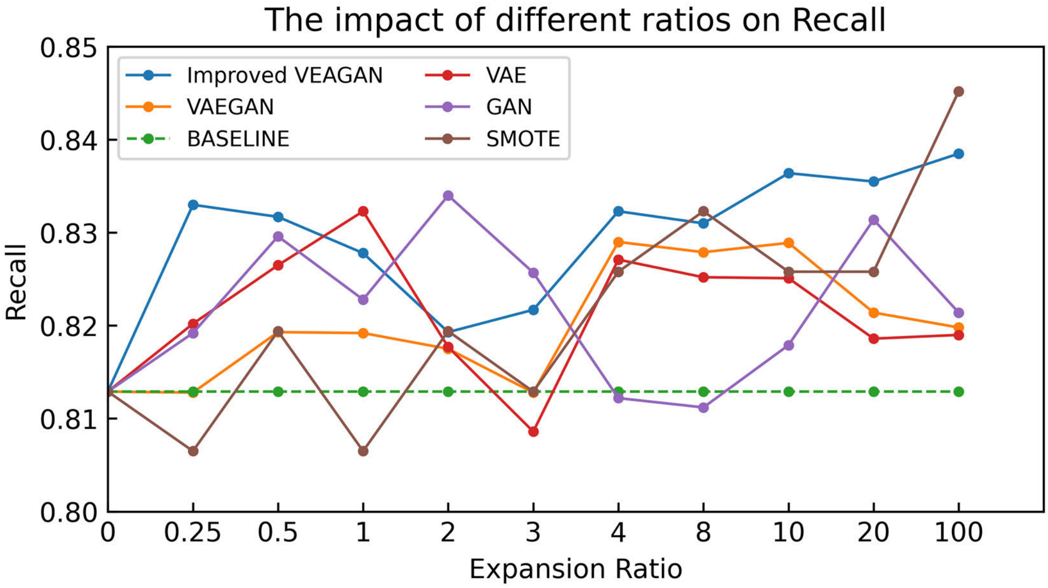


FIGURE 12. Oversampling model and baseline model classification recall under different training set expansion ratios.

The DNN was initially trained on an imbalanced (smaller) dataset, but it is well-known that DNN can perform well when there is a larger amount of data. To further validate that the improved VAEGAN model can achieve better results, we ran enhanced data on the DNN model. The experimental results are shown in Tables 8, 9 and 10.

The Recall of DNN on the raw data is 0.8065, while the Recall of XGBoost on the raw data is 0.8129. The Precision of DNN on the raw data is 0.8562, while the Precision of XGBoost on the raw data is 0.9197. The F1 score of DNN on the raw data is 0.8306, while the F1 score of XGBoost on the raw data is 0.8630. XGBoost outperforms the DNN model significantly on all three classification metrics mentioned above.

The DNN model is used as a classification algorithm. Through comparative experiments, we found that using the improved VAEGAN model for data augmentation still achieved better results. The maximum improvement in Recall for classification was 0.0276. The Precision for classification was significantly improved, with the highest increase being 0.0221. Across all augmentation ratios, the F1 score for classification was greatly improved, with the highest increase being 0.0235. However, the performance of the DNN model on augmented data is still not as good as that

TABLE 6. Recall and Specificity as the number Ng of generated examples vary.

N_g	Recall					Specificity				
	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN
0	0.8129	0.8129	0.8129	0.8129	0.8129	0.99987	0.99987	0.99987	0.99987	0.99987
84	0.8065	0.8192	0.8202	0.8128	0.8330	0.99987	0.99987	0.99988	0.99988	0.99988
168	0.8194	0.8296	0.8265	0.8193	0.8317	0.99984	0.99987	0.99987	0.99990	0.99987
337	0.8065	0.8228	0.8323	0.8192	0.8278	0.99985	0.99988	0.99988	0.99988	0.99991
674	0.8194	0.8340	0.8177	0.8175	0.8193	0.99987	0.99986	0.99988	0.99988	0.99991
1011	0.8129	0.8257	0.8086	0.8128	0.8217	0.99986	0.99984	0.99987	0.99990	0.99992
1348	0.8258	0.8122	0.8271	0.8290	0.8323	0.99974	0.99986	0.99987	0.99987	0.99991
2696	0.8320	0.8112	0.8252	0.8279	0.8310	0.99979	0.99984	0.99988	0.99988	0.99989
3370	0.8258	0.8179	0.8251	0.8289	0.8364	0.99980	0.99978	0.99988	0.99988	0.99991
6740	0.8258	0.8314	0.8186	0.8214	0.8355	0.99974	0.99978	0.99987	0.99990	0.99987
33700	0.8452	0.8214	0.8190	0.8198	0.8385	0.99950	0.99974	0.99987	0.99988	0.99991

TABLE 7. AUC as the number Ng of generated examples is varied.

N_g	AUC				
	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN
0	0.98461	0.98461	0.98461	0.98461	0.98461
84	0.98510	0.98512	0.98526	0.98537	0.98549
168	0.98584	0.98534	0.98635	0.98683	0.98846
337	0.98446	0.98441	0.98487	0.98472	0.98506
674	0.98549	0.98531	0.98532	0.98499	0.98576
1011	0.98822	0.98586	0.98667	0.98683	0.98733
1348	0.98884	0.98236	0.98431	0.98317	0.98399
2696	0.98983	0.98363	0.98563	0.98731	0.98746
3370	0.98900	0.98391	0.98683	0.98654	0.98723
6740	0.99053	0.98328	0.98587	0.98499	0.98672
33700	0.98555	0.98186	0.98443	0.98431	0.98476

TABLE 8. Recall as the number Ng of generated examples is varied.

N_g	Recall				
	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN
0	0.8065	0.8065	0.8065	0.8065	0.8065
84	0.7935	0.8145	0.8224	0.8214	0.8276
168	0.8000	0.8178	0.8237	0.8145	0.8194
337	0.8000	0.8105	0.8189	0.8247	0.8312
674	0.8194	0.8232	0.8167	0.8219	0.8249
1011	0.7742	0.8163	0.821	0.8225	0.8225
1348	0.8129	0.8204	0.8202	0.8164	0.8203
2696	0.8194	0.8301	0.8157	0.8149	0.8249
3370	0.8065	0.8264	0.8232	0.8214	0.8264
6740	0.8000	0.8265	0.8143	0.8174	0.8341
33700	0.8000	0.8184	0.8189	0.8152	0.8302

of the XGBoost model. XGBoost is more suitable for the classification of imbalanced data. The augmented data is still imbalanced, although the degree of imbalance has been reduced.

TABLE 9. Precision as the number Ng of generated examples is varied.

N_g	Precision				
	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN
0	0.8562	0.8562	0.8562	0.8562	0.8562
84	0.8255	0.8573	0.8642	0.8624	0.8635
168	0.8435	0.8554	0.8704	0.8654	0.8681
337	0.8732	0.8547	0.8664	0.8678	0.8783
674	0.8194	0.8466	0.8668	0.8713	0.8734
1011	0.8000	0.8503	0.8623	0.8658	0.8672
1348	0.8630	0.8468	0.8609	0.8623	0.8674
2696	0.8301	0.8345	0.8603	0.8614	0.8653
3370	0.8503	0.8327	0.8583	0.8600	0.8667
6740	0.8493	0.8237	0.8586	0.8589	0.8644
33700	0.8378	0.8184	0.8557	0.8564	0.8630

TABLE 10. F1_score as the number Ng of generated examples is varied.

N_g	F1_score				
	SMOTE	GAN	VAE	VAEGAN	Improved VAEGAN
0	0.8306	0.8306	0.8306	0.8306	0.8306
84	0.8092	0.8354	0.8428	0.8414	0.8452
168	0.8212	0.8362	0.8464	0.8392	0.8430
337	0.835	0.8320	0.8420	0.8457	0.8541
674	0.8194	0.8347	0.8410	0.8459	0.8485
1011	0.7869	0.8330	0.8411	0.8436	0.8443
1348	0.8372	0.8334	0.8401	0.8387	0.8432
2696	0.8247	0.8323	0.8374	0.8375	0.8446
3370	0.8278	0.8295	0.8404	0.8403	0.8461
6740	0.8239	0.8251	0.8359	0.8376	0.8490
33700	0.8185	0.8184	0.8369	0.8353	0.8463

VI. CONCLUSION AND FUTURE WORK

This paper proposes a new credit card fraud detection method that combines the improved VAEGAN oversampling method with the XGBoost classification algorithm. The improved VAEGAN oversampling model is trained using the minority class samples in the original training set, and then a large amount of minority class data is generated. Although our model is proposed in the context of credit card fraud detection, it can be easily extended to other application domains involving class imbalance. The experimental results suggest that the XGBoost algorithm, as the baseline model for credit card fraud detection, has achieved better classification results than Logistic, decision tree, random forest, and neural network. This reveals that ensemble methods may be more effective when dealing with class-imbalanced classification problems. Oversampling methods are also an effective way to improve the performance of imbalanced classification problems.

Overall, the improved VAEGAN method achieved an excellent precision and F1 score, but the improvement in recall and AUC at certain expansion ratios were not significant compared to the GAN and VAE methods. Compared with the VAEGAN model, the complexity has increased. In the future, we will study how to stabilize further the improvement of Recall and AUC based on steadily improving the precision and F1 score.

REFERENCES

[1] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.

[2] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Exp. Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[3] H. Patel, D. S. Rajput, G. T. Reddy, C. Iwendi, A. K. Bashir, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 4, Apr. 2020, Art. no. 155014772091640.

[4] C. Jian, J. Gao, and Y. Ao, "A new sampling method for classifying imbalanced data based on support vector machine ensemble," *Neurocomputing*, vol. 193, pp. 115–122, Jun. 2016.

[5] Y. Liu, Y. Wang, X. Ren, H. Zhou, and X. Diao, "A classification method based on feature selection for imbalanced data," *IEEE Access*, vol. 7, pp. 81794–81807, 2019.

[6] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Exp. Syst. Appl.*, vol. 51, pp. 134–142, Jun. 2016.

[7] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 129–134.

[8] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Inf. Sci.*, vol. 557, pp. 317–331, May 2021.

[9] A. Salazar, G. Safont, and L. Vergara, "Semi-supervised learning for imbalanced classification of credit card transaction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.

[10] N. R. Dzakiyullah, A. Pramuntadi, and A. K. Fauziyyah, "Semi-supervised classification on credit card fraud detection using autoencoders," *J. Appl. Data Sci.*, vol. 2, no. 1, pp. 1–7, 2021.

[11] L. Ni, J. Li, H. Xu, X. Wang, and J. Zhang, "Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, pp. 1–16, 2023.

[12] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, "GMM-based undersampling and its application for credit card fraud detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[13] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Mar. 2018, pp. 1–6.

[14] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018.

[15] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022.

[16] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, Oct. 2017, pp. 1–9.

[17] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. A. Khalid, "Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset," in *Proc. IEEE 16th Int. Conf. Control Autom. (ICCA)*, Oct. 2020, pp. 803–808.

[18] A. K. Gangwar and V. Ravi, "WIP: Generative adversarial network for oversampling data in credit card fraud detection," *Proc. 15th Int. Conf. (ICISS)*. Hyderabad, India: Springer, Dec. 2019, pp. 123–134.

[19] Y.-J. Lee, Y.-R. Yeh, and Y. F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1460–1470, Jul. 2013.

[20] B. Prasetiyo, Alamsyah, M. A. Muslim, and N. Baroroh, "Evaluation performance recall and F2 score of credit card fraud detection unbalanced dataset using SMOTE oversampling technique," *J. Physics: Conf. Ser.*, vol. 1918, no. 4, Jun. 2021, Art. no. 042002.

[21] H. Zhu, M. Zhou, G. Liu, Y. Xie, S. Liu, and C. Guo, "NUS: Noisy-sample-removed undersampling scheme for imbalanced classification and application to credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, early access, Mar. 7, 2023, doi: 10.1109/TCSS.2023.3243925.

[22] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decis. Support Syst.*, vol. 95, pp. 91–101, Mar. 2017.

[23] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019.

[24] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," *Global Transitions Proc.*, vol. 2, no. 1, pp. 35–41, Jun. 2021.

[25] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Exp. Syst. Appl.*, vol. 100, pp. 234–245, Jun. 2018.

[26] A. Singh, R. K. Ranjan, and A. Tiwari, "Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms," *J. Experim. Theor. Artif. Intell.*, vol. 34, no. 4, pp. 571–598, Jul. 2022.

[27] N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, p. 662, Feb. 2022.

[28] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[30] H. Al Majzoub, I. Elgedawy, Ö. Akaydın, and M. Köse Ulukök, "HCAB-SMOTE: A hybrid clustered affinitive borderline SMOTE approach for imbalanced data binary classification," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 3205–3222, Apr. 2020.

[31] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Inf. Sci.*, vol. 479, pp. 448–455, Apr. 2019.

[32] H. Tingfei, C. Guangquan, and H. Kuihua, "Using variational auto encoding in credit card fraud detection," *IEEE Access*, vol. 8, pp. 149841–149853, 2020.

YUANMING DING received the Ph.D. degree from Keio University, Japan, in 2004. From November 2004 to November 2016, he was a Postdoctoral Fellow with JSPS. Since 2009, he has been a Professor with the Information Engineering College, Dalian University, China. His research interests include communication signal processing, network technologies, machine learning, and information security.

WEI KANG received the B.S. degree in network engineering from Chuzhou University, Chuzhou, China, in 2021. He is currently pursuing the master's degree with the Communication and Network Key Laboratory, Dalian University. His current research interests include machine learning and information security.

JIANXIN FENG received the Ph.D. degree from Northeastern University, China, in 2005. From 1999 to 2012, she was a Teacher with the Institute of Information Science and Engineering, Northeastern University. From 2018 to 2019, she was a Visiting Scholar with the Department of Computer Science, Liverpool John Moores University. She is currently an Associate Professor with the College of Information Engineering, Dalian University, China. Her current research interests include network protocol, wireless communication, machine learning, and information security.

BO PENG received the B.Sc. degree from the Jiangxi University of Science and Technology, in 2021. He is currently pursuing the master's degree with the Communication and Network Key Laboratory, Dalian University. His current research interests include neural networks and machine learning.

ANNA YANG received the B.S. degree in network engineering from Chuzhou University, Chuzhou, China, in 2021. She is currently pursuing the master's degree with the Communication and Network Key Laboratory, Dalian University. Her current research interests include machine learning and image processing.