JOURNAL OF DECISION MAKING AND HEALTHCARE
Volume 2 (2025), No. 1, 43–63
https://doi.org/10.69829/jdmh-025-0201-ta03
Tulipa Opera Scholarum

ANALYSIS AND RESEARCH ON FRAUD DETECTION AND RECOGNITION PROBLEM MODELING BASED ON CONDITIONAL GENERATIVE ADVERSARIAL NETWORK

LUPING ZHI[1] AND WANMIN WANG[2,*]

[1] *Business School, the University of Shanghai for Science & Technology, Shanghai, China*
[2] *School of Cyber Science and Engineering, Southeast University, Nanjing, China*

Abstract. To address problems such as multimodal non-Gaussian values of continuous variables, modeling of discrete and continuous variables, and extreme imbalance of data distribution in tabular data, this study combined anomaly detection and deep learning techniques, first in generators and discriminators, to add conditional variables to construct Conditional Generative Adversarial Nets, making the Generative Adversarial Network model easier to control and solving the uncertainty of the original Generative Adversarial Network model. Then, under the condition of satisfying the categorical labels, the generative model in the conditional generative adversarial network is used to generate artificial transaction data with the same distribution as the real data to achieve data augmentation for LightGBM for classification prediction. Finally, principal component analysis and t-SNE were used to reduce dimensional techniques to visualize synthetic large-scale and high-dimensional data. Experiments show that the model only needs to extract a small amount of transaction data with categorical labels from the training set for training; furthermore, the training process is easier to control, the model is not easy to collapse but has better feature extraction capabilities, and the synthetic transaction data has better coverage. It is more in line with expectations, can better overcome the defects of traditional fraud-detection models for misclassifying most samples, and helps improve the industry's efficiency in identifying transaction fraud to meet the needs of enterprises.

Keywords. Fraud detection, Deep learning, Conditional generative adversarial networks, LightGBM.

1. Introduction

Data anomalies are results or values that are not as expected and are divided into three main categories: data point-based anomalies, context-based anomalies, and pattern-based anomalies. Anomaly detection identifies anomalous data points, events, or observations using advanced algorithms and is widely used in fields such as cybersecurity, finance, manufacturing, healthcare, and banking. In areas such as finance and insurance, anomaly detection can be used to detect risks such as fraudulent transactions and claims [1], which can be roughly divided into two stages: fraud detection methods based on machine learning and fraud detection methods based on deep learning. With the development of technologies such as cloud computing, the Internet of Things, and artificial intelligence, large-scale datasets containing complete transaction information have become accessible, enabling the application of machine learning in fraud detection. Classical machine learning-based fraud detection methods can be broadly categorized into two types: clustering (an unsupervised learning technique for anomaly detection) and classification (a supervised learning approach that trains a model on historical transaction data to classify new transactions). The latter, which learns to distinguish between normal and fraudulent transactions, is the most widely used approach for fraud detection [2]. Taha and Malebary [3] introduced the OLightGBM for credit card fraud detection, while Zheng et al. [4] proposed an improved TrAdaBoost algorithm to address the concept drift problem in fraud detection. Additionally,

*Corresponding author.
E-mail address: lpzhi@usst.edu.cn (L. Zhi), 230249452@seu.edu.cn (W. Wang).
Accepted: March 03, 2025.

Ileberi et al. [5] presented a fraud detection engine based on machine learning, incorporating a genetic algorithm for feature selection and various classifiers, including decision trees, random forests, and neural networks.

However, fraud detection is challenged by highly imbalanced datasets, where fraudulent transactions constitute a small fraction of the total. This imbalance often leads to classifier bias towards the majority class, resulting in poor performance for detecting fraud. Several methods have been proposed to address this issue. Awoyemi et al. [6] applied a hybrid undersampling and oversampling technique, showing that k-nearest neighbors outperformed Bayesian and logistic regression classifiers. Yang et al. [7] developed a fraud detection framework, FFD, that utilized federated learning and oversampling to tackle imbalanced datasets. Varmedja et al. [8] demonstrated that the SMOTE technique and random forests improved classification in credit card fraud detection. Despite these advancements, challenges remain, particularly with generalization and overfitting in models trained on imbalanced datasets.

The growing availability of computing power, particularly through Graphics Processing Units (GPUs), has shifted the focus from traditional models to deep learning techniques. Deep learning has surpassed shallow models in areas like computer vision and network security and is increasingly applied to fraud detection. These techniques can automatically learn feature representations from data, enabling better scalability and adaptability to large datasets. Chouiekh and Haj [9] utilized Deep Convolutional Neural Networks (DCNN) to classify fraud in a mobile operator dataset, outperforming traditional machine learning models in terms of accuracy and training time. Wang et al. [10] proposed a semi-supervised attention graph neural network, which incorporated a hierarchical attention mechanism to enhance model interpretability and fraud detection performance. Branco et al. [11] employed recurrent neural networks (RNNs) to detect fraud in real-time by modeling payment histories as interleaved sequences. Liu and Jiang [12] introduced a novel loss function to improve model stability by optimizing feature representation. Zhang et al. [13] integrated deep learning with a feature engineering framework for fraud detection, further enhancing model performance. Lastly, Forough and Momtazi [14] proposed a voting mechanism based on artificial neural networks, which improved real-time fraud detection accuracy.

Traditional deep learning models and statistical methods for density estimation primarily focus on identifying suitable probability distributions and developing sampling algorithms. However, these approaches are often constrained by the limited flexibility of the underlying statistical models, which can hinder their ability to model complex data distributions. In contrast, generative adversarial networks (GANs) offer a more powerful and flexible approach by learning a latent feature space that captures the distribution of the given data. This flexibility allows GANs to generate high-quality synthetic data, making them a valuable tool for data augmentation and model improvement. For example, Sethia, Patel, and Raut [15] employed several GAN variants, including vanilla GANs, least squares GANs, Wasserstein GANs, marginal adaptive GANs, and Relaxed Wasserstein GANs, to generate pseudo-data, thereby improving model performance and achieving a 12.86% increase in recall.

Despite their effectiveness, GANs face significant training challenges, including convergence failure, mode collapse, and difficulty generating meaningful data when the underlying distribution is complex. To address these issues, several GAN-based approaches have been proposed. For example, Ishfaq et al. [16] introduced the Triplet-based Variational Autoencoder (TVAE), which enhances data generation flexibility and scalability, especially when data includes both categorical and continuous features. Fiore et al. [17] used GANs to generate fraudulent transactions and incorporated them into training datasets, improving classifier performance. Ba [18] demonstrated that Wasserstein GANs (WGANs) are easier to train and generate data more closely aligned with the original fraud distribution, providing significant advantages for fraud detection. Xu et al. [19] introduced CTGAN, which addresses mode collapse and captures complex feature interactions, particularly in tabular data. Wang and Yao [20] developed an unfolding GAN-based oversampling method, outperforming traditional techniques on a credit card dataset and achieving an F1 score of 85.59%. Additionally, transformer-based models for fraud detection,

such as those proposed by Yu et al. [21] and Tian et al. [22], have significantly improved the capture of temporal patterns in transactional data, further enhancing fraud detection performance.

A promising approach to stabilize GAN training and mitigate mode collapse is the use of Conditional GANs (CGANs), which incorporate conditional variables into both the generator and discriminator. This allows the model to generate data based on specified conditions, enhancing stability and the quality of synthetic data. Mirza and Osindero [23] introduced CGANs to address challenges like mode collapse in traditional GANs. By incorporating conditions such as category labels in the MNIST dataset or word vectors in the MIR Flickr25000 dataset, CGANs generate more relevant and specific data. While CGANs have been widely applied in image and speech processing, their use in tabular data, with its diverse continuous variables and skewed feature distributions, remains less explored. However, CGANs hold significant potential for generating structured tabular data, particularly in fraud detection tasks where data distributions are often imbalanced and complex. In conclusion, while traditional data generation models lack flexibility, CGANs provide a more powerful and adaptable solution for tackling complex data challenges in fraud detection. Integrating CGANs into fraud detection systems could enhance both data generation and model performance, offering a promising avenue for future research in imbalanced settings.

In this study, a fraud-detection model based on conditional generative adversarial networks was designed for multimodal non-Gaussian values of continuous variables in tabular data, modeling of discrete and continuous variables, and extreme imbalance in category columns. The model uses conditional generative adversarial networks to model probability distributions in tabular data and adds categorical labels of transaction data to the generator and discriminator, such that the artificial transaction data generated by the generator are consistent with the real data distribution under the condition that the categorical labels are satisfied. This makes the adversarial network model more controllable, resolves the uncertainty of the original generative adversarial network model, and generates transaction data that are more consistent with expectations, which in turn facilitates classification prediction using LightGBM (LGB) [24]. Finally, using principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction techniques to visualize large-scale and high-dimensional data, we verified that the model has good feature extraction ability and that the synthesized data has good coverage.

2. Fraud Detection Methods

2.1. **Problem description.** The fraud detection problem studied herein is theoretically an anomaly detection problem under conditions of extreme imbalance in the distribution of positive and negative class data, which is essentially a binary classification of a class of serial data. This study constructs a binary detection model by processing historical credit card transaction data, and uses the model to analyze the current spending behavior of cardholders to identify whether the behavior is credit card fraud. These problems include large data sample sizes, high computational complexity, highly skewed data distributions, and sequential relationships between the data.

Let the feature set be X_i and the sample class be Y_i, where $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}, i = 1, 2, \ldots, n$, $Y_i = \{y_i\} \in \{0, 1\}$ and n are the numbers of samples and m is the number of features. When $y_i = 0$, the i-th transaction is normal, and when $y_i = 1$, the i-th transaction is fraudulent. The credit card's historical transaction data $f(x)$ and the credit card holder's i-th transaction data $f(x_i)$, so it can be determined whether $f(x_i)$ is a fraudulent transaction based on $f(x)$, that is,

$$f(x) = \{X_i, Y_i\} = \begin{bmatrix} x_{11}, x_{12}, \ldots, x_{1m}, y_1 \\ x_{21}, x_{22}, \ldots, x_{2m}, y_2 \\ \vdots \\ x_{n1}, x_{n2}, \ldots, x_{nm}, y_n \end{bmatrix}, \tag{2.1}$$

$$\text{and } f(x_i) = \{x_{i1}, x_{i2}, \ldots, x_{im}\}. \tag{2.2}$$

Currently, the most commonly used methods for addressing the category imbalance problem are oversampling, undersampling, and improvements to various algorithms. In this study, we attempt to apply deep learning algorithms to the credit card fraud detection problem, use credit card historical transaction data to train conditional generation adversarial networks, replace the sampler with an arbitrary algorithm with microscopic parameters, and then adjust the generator and discriminator accordingly according to the results of discrimination so that the discriminator cannot distinguish between data sources until Nash equilibrium is reached. Finally, we obtain an efficient generator model and discriminator model for generating a few categories of fraud samples to maximize the quality of the generated samples.

2.2. **A fraud detection model based on conditional generative adversarial networks.** To overcome the shortcomings of traditional fraud-detection methods in dealing with unbalanced datasets with high misclassification rates for most classes of samples, this study constructed a fraud-detection model based on conditional generative adversarial networks. The overall framework of the model is shown in Figure 1 and is divided into four parts: data preprocessing, a conditional generative adversarial network model, a LGB classification model, and model testing and evaluation. The training of a conditional generative adversarial network uses the idea of a zero-sum game in game theory, and the network consists of a generator and discriminator that learn through the mutual confrontation of the generator and discriminator. Unlike the GAN, the conditional generative adversarial network incorporates conditional variables in the modeling of both generative and discriminative models to guide the data generation process. The conditional variable can be categorical labels or data from different modalities. In this study, category label Y was used as a condition variable.

The network structure of the data generator in the conditional generative adversarial network is shown in Figure 2. Randomly generated noisy data Z conforming to the Gaussian distribution were combined with categorical labels Y and input into the generator model, and the input layer was connected to the hidden layer using a fully connected network. In neural networks, the output of each layer is a linear combination of the inputs of the previous layer, and a linear model exhibits poor expressiveness. Introducing an activation function between the layers to add nonlinear factors can improve the fitting ability of deep neural networks. Because the sigmoid activation function tends to cause problems such as gradient disappearance when the input is very large or very small, this study adds a LeakyRELU activation function with a slope of 0.01 between the input and hidden layers in the generator model to correct the data distribution while retaining some of the negative axis values so that the negative axis information is not lost.

The network structure diagram of the data discriminator in the conditional generative adversarial network is shown in Figure 3. The input of the data discriminator consists of (X, Y) formed by combining the real credit card historical transaction dataset X and the category label Y, that is, the data samples in the training set, and $(G(Z|Y), Y)$ formed by combining the synthetic data $G(Z|Y)$ generated by the data generator with the category label Y. Similar to the data generator model, the discriminator model uses a six-layer fully connected neural network and a LeakyRELU activation function with a slope of 0.01; however, a dropout layer is added to the models' training process such that the nodes in the hidden layer are temporarily discarded from the network at each iteration (including

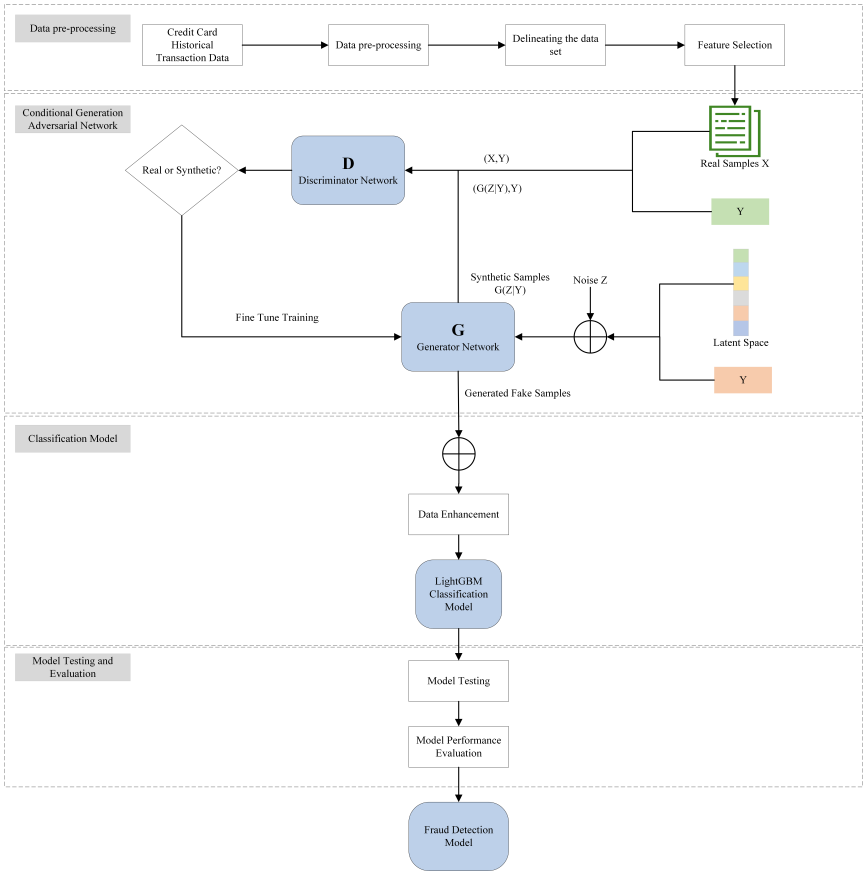

FIGURE 1. Architecture of the fraud detection model based on conditional generative adversarial network

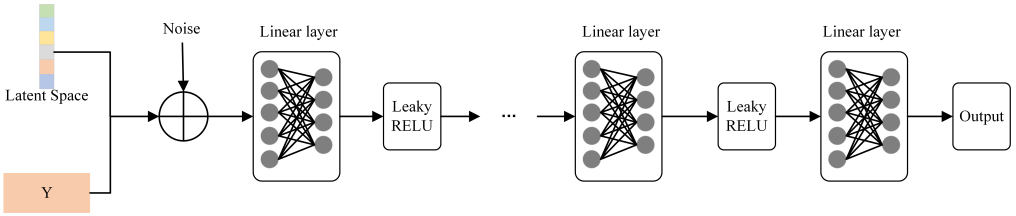

FIGURE 2. Network structure of the data generator in the conditional generation network

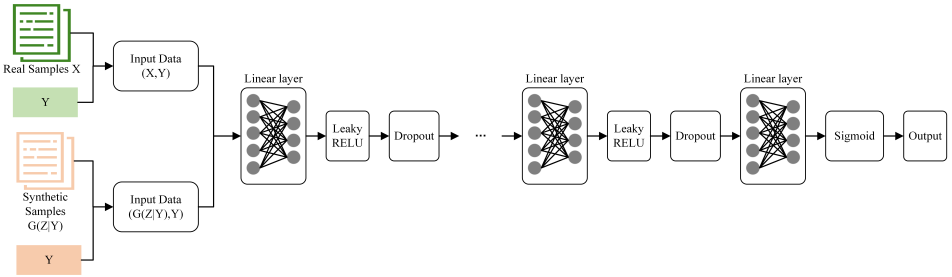

FIGURE 3. Network structure of data discriminator in conditional generative adversarial network

forward and backward propagation) with a certain probability of reducing overfitting. In this study, we set the deactivation probability, that is, each neuron has a 0.1 probability of not being activated. The output of the data discriminator is a probability value that determines whether the input data are real credit card transaction data or credit card transaction data synthesized by the generator. If the input data are real credit card transaction data, the output of the data discriminator model is close to 1; if the input data are credit card transaction data synthesized by the generator, it is close to 0. Therefore, a sigmoid activation function is used in the last layer of the deep neural network to transform the output into a probability representation between zero and one for binary classification. All weights in the generator and discriminator are fine-tuned according to adversarial training. After several training iterations until the output value of the discriminator is close to approximately 0.5, the Nash equilibrium is reached, the discriminator cannot discriminate the input source of the data, and the conditional generative adversarial network training is completed.

The specific steps of the CGAN-LGB-based fraud-detection model are as follows:

a) Preprocessing of historical transaction data: data filtering, missing value processing, and data coding. Effective data preprocessing can improve the model's effectiveness and reduce the time required for the actual modeling process. To improve the ability of the learning algorithm to generalize, and the readability and interpretation of the results, data filtering and missing value processing reduce the amount of data while maintaining the original data structure and meaning. Data coding transforms the classification features into classification values to adapt the data to the algorithm and library for further learning.

b) In the historical credit card transaction dataset, fraudulent transactions account for 0.172% of all transactions, and the data distribution is extremely unbalanced. The random assignment cannot guarantee the ratio of normal and fraudulent transactions in the training and test sets. Therefore, under the premise that both the training and test sets contain fraudulent transactions, this study allocates the credit card historical transaction dataset according to the ratio in the categorical labels, with 70% being the training set and 30% being the test set. To avoid data leakage, the training process of all models is conducted only on the training set.

c) To build a fraud detection system, feature selection is essential to improve the performance and accuracy of the classification model by reducing redundant and irrelevant features and finding the optimal subset of features from the feature set using an appropriate search strategy to effectively reduce runtime and improve model accuracy. The optimal feature subset is screened using a random forest-based sequence-forward search strategy approach to reduce the runtime of the conditional generation adversarial network and to reduce storage costs and overfitting risks.

d) In the conditional generative adversarial network's training process, the randomly generated noisy data in the potential space conforming to the Gaussian distribution and the conditional variable Y are combined and used to train the generator, which generates synthetic data $G(X|Y)$ with the same distribution as the real data when the condition Y is satisfied. Then, the real training data set X and the sample category label Y are combined to form (X, Y), and the synthetic data $G(X|Y)$ generated by the generator is combined with the credit card transaction data category label Y to form $(G(X|Y), Y)$, and these two parts of data are mixed and disordered, and the one-to-one correspondence is maintained, and input to the discriminator D as a whole, and the discriminator D determines whether the input data is the real credit card transaction data or the credit card transaction data synthesized by the generator. Finally, adjustments are made following the discriminator's discriminatory outcomes, and numerous iterations are carried out until the Nash equilibrium is reached. The discriminator is unable to discriminate the data's input source, at which point the conditional generative adversarial network has been trained.

e) The optimal feature subset is used to train the base classifier, and the hyperparameters in the model are determined by "grid search + 5-fold hierarchical cross-validation," and the LGB strong classifier

with optimal generalization performance is constructed based on the selected hyperparameters. Credit card fraud transaction data are then generated using the conditional generative adversarial network's data generator. Fraudulent transactions are combined with the initial training dataset to achieve data augmentation, which avoids the overfitting problem caused by unbalanced data.

f) Finally, the entire model is tested using test data from historical credit card transaction data, and the accuracy, precision, recall, F1-score, and AUC values are used as evaluation metrics to assess the generalization performance of the CGAN-LGB-based fraud-detection model and the output test results.

2.3. **Fraud detection algorithms.** The proposed framework for fraud detection integrates a CGAN with LGB, formalized through a systematic procedure as illustrated in Algorithm 1. The CGAN is an extension of the original GAN that feeds the categorical label Y as part of the input layer to both the discriminator and the generator. In the generator, the prior input noise Z and the conditional information Y jointly form a subjoint hidden layer representation. The objective function of the CGAN is a two-player minimax game with conditional probabilities:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))], \tag{2.3}$$

where $p_{\text{data}}(x)$ denotes the distribution of real credit card historical transaction data, $p_z(x)$ denotes the distribution of credit card transaction data synthesized by the generative model, and y denotes the category label. During the model's training process, the generator and discriminator are trained simultaneously; this fixes the discriminator (respectively, generator) by adjusting the parameters (respectively, D) to minimize (respectively, maximize) the objective function, thereby forming a confrontation.

During the generator training process, the generator is based on the discriminative results of the discriminator, and it is necessary to ensure that the discriminator does not change significantly. The task of the generator is to build samples that can generate the distribution of the real credit card historical transaction data so that the discriminator cannot identify the input source of the data. Thus, the loss function of the generator is given by:

$$V(G) = \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))]. \tag{2.4}$$

During the discriminator training process, the discriminator is based on the training results of the generator, and it is necessary to ensure that the generator does not change significantly. The task of the discriminator is to maximize the discrimination between the input data, whether it is the real credit card transaction data or the credit card transaction data synthesized by the generator. So the loss function of the discriminator is shown in:

$$V(D) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x|y)]. \tag{2.5}$$

In the LGB classifier training process, the histogram optimization and depth-first search strategies are used to integrate the base classifier (decision tree) training to find the best branching point and reduce time complexity. A second-order Taylor expansion is used, and a regular term is added as the objective function to prevent problems such as overfitting, as shown in:

$$\mathcal{L}_N = \sum_{i=1}^{N} l(y_i, y_i^{N-1} + F_N(x_i)) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} W_j^2, \tag{2.6}$$

where $\mathcal{L}_N$ is the objective function after the N iterations, l is the original objective function, y_i is the category label of the i-th sample, x_i is the i-th sample, F_N is the model of the Nth iteration, T is the number of leaf nodes, and W_j is the output of the j-th node.

Algorithm CGAN-LGB Fraud Detection Framework

Input: Historical transaction dataset $\mathcal{D}_{\text{train}} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{N}$ where $\mathbf{Y}_i \in \{0, 1\}$ denotes class labels
Output: Trained LGB classifier with enhanced fraud detection capability

Initialize generator G_θ and discriminator D_ϕ with the normal initialization
Define latent space dimension d and conditional embedding dimension c
for each training iteration $k = 1 : K$ **do**
    Sample latent vectors $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_d)$
    Concatenate with class labels: $\tilde{\mathbf{z}} = [\mathbf{z} \oplus \mathbf{y}]$
    Generate synthetic samples: $\mathbf{X}_{\text{synth}} = G_\theta(\tilde{\mathbf{z}})$
    Compute generator loss: $\mathcal{L}_G = \mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D_\phi(G_\theta(\tilde{\mathbf{z}}))]$
    Update parameters: $\theta \leftarrow \theta - \eta_G \nabla_\theta \mathcal{L}_G$
end for
for each training iteration $m = 1 : M$ **do**
    Sample real data batch: $\mathbf{X}_{\text{real}} \sim \mathcal{D}_{\text{train}}$
    Generate synthetic batch: $\mathbf{X}_{\text{synth}} = G_\theta([\mathbf{z} \oplus \mathbf{y}])$
    Compute discriminator loss:

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{X}\sim p_{\text{data}}}[\log D_\phi(\mathbf{X})] - \mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D_\phi(G_\theta(\tilde{\mathbf{z}})))]$$

    Update parameters: $\phi \leftarrow \phi + \eta_D \nabla_\phi \mathcal{L}_D$
end for
while not converged **do**
    Alternate between Phase 2 and 3 until Nash equilibrium:

$$\min_\theta \max_\phi \mathbb{E}[\log D_\phi(\mathbf{X}|\mathbf{y})] + \mathbb{E}[\log(1 - D_\phi(G_\theta(\mathbf{z}|\mathbf{y})))]$$

    Monitor Jensen-Shannon divergence $\mathcal{D}_{\text{JS}}\left(p_{\text{data}} \,\|\, p_g\right)$
end while
Generate balanced dataset:

$$\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{train}} \cup \{\mathbf{X}_{\text{synth}}^{(i)}, y = 1\}_{i=1}^{N_{\text{fraud}}}$$

where N_{fraud} balances class distribution
Train LGB model on $\mathcal{D}_{\text{aug}}$ with focal loss:

$$\mathcal{L}_{\text{focal}} = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

Optimize hyperparameters via Bayesian optimization
Evaluate the model on the test set $\mathcal{D}_{\text{test}}$ using the metrics: precision, recall, AUC, and F1.

3. Data Pre-Processing and Evaluation Indicators

3.1. **Environment and dataset.** This experiment is based on the Pytorch framework for model construction, and the detailed experimental configuration is shown in Table 1. To evaluate the anomaly detection performance of the proposed algorithm in this paper, a credit card fraud detection dataset is used for evaluation. The dataset was collected by Worldline and ULB's machine learning group during a collaboration aimed at improving the existing fraud detection process through techniques in existing data-driven strategies to improve fraud detection accuracy and better explain fraud patterns as well as prevent fraud. There were 492 fraudulent transactions out of 284,807 transactions, with fraudulent transactions accounting for 0.172% of all transactions, an extreme imbalance in the data distribution [25].

TABLE 1. Experimental Environment Configuration

Project	Configuration Description	Project	Configuration Description
Operating System	Windows 10	Compilation Environment	PyCharm 2022.3.1
Python Version	3.7	GPU	NVIDIA GeForce GTX 1050 Ti
CUDA Version	12.0	RAM	128 GB
Torch Version	1.13.1	Scikit-learn Version	1.0.2

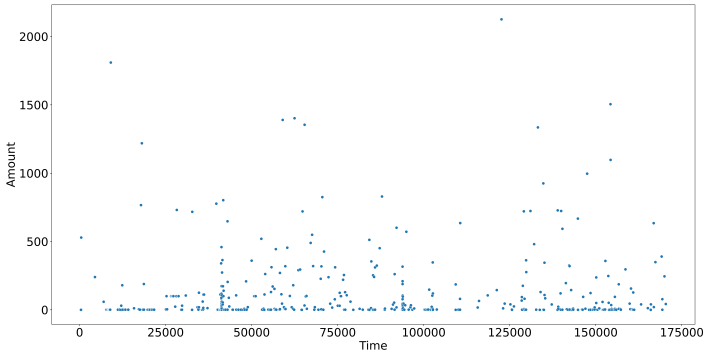

FIGURE 4. Feature "Time" scatter plot

3.2. **Data pre-processing.** Due to issues such as the confidentiality of credit card issuers and the information security of credit card holders, the original dataset contains only numerical input variables as a result of PCA conversion without providing original features and more background information about the data, and the overall quality is good as there are no missing values for each feature as well as labels in the data set, so operations such as missing value processing and data coding are not required. Except for the two features of "Time" and "Amount", PCA feature extraction and conversion are not used. After the other features are extracted and converted, the feature attribute names and related background information are hidden and converted into features V1, V2, ... V28, which basically obey the normal distribution. It can be seen from Figure 4 that there is no cluster mode in a certain time interval. Therefore, we assume that fraud occurs randomly.

The overall quality of the dataset is good, and there are no missing values in the dataset for any of the features or labels. As shown in Figure 5, there is no correlation between features V1 to V28, and there is no correlation between time and transaction amount. Since the two features "Time" and "Amount" are not transformed by PCA, they have obvious differences in the range of values compared with other features, so they need to be standardized. In this empirical study, this paper uses the StandardScaler function in the Scikit-Learn library to perform Z-core standardization on the two characteristics of "Time" and "Amount" and obtains standard data sets, which include 70% of the standardized training set and 30% of the standardized testing set.

In building a credit card fraud detection system, feature selection reduces running time and improves model accuracy by reducing redundant and irrelevant features, using appropriate search strategies to

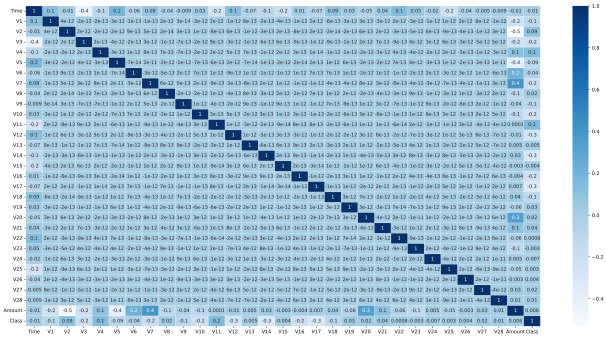

FIGURE 5. Data association diagram

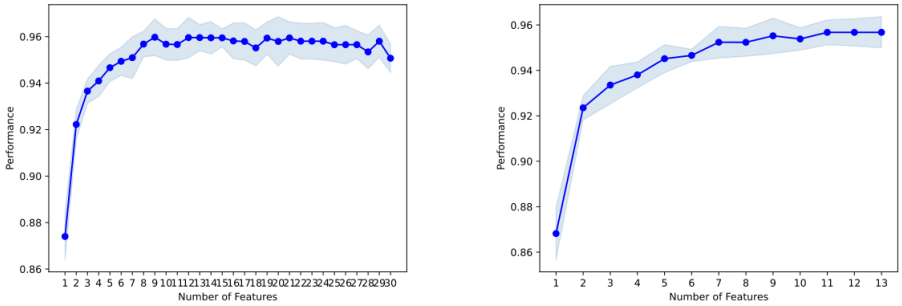

FIGURE 6. Iteration diagram of random forest-based sequential feature selection method. (a) Iterative graph with feature number 30. (b) Iteration diagram with feature number 13

find the optimal feature subset, and reducing feature dimensionality. According to the relationship between the method itself and the constructed model, feature selection is mainly classified into the filter, embedded, and wrapper methods. To improve the training speed and model prediction ability of the conditional adversarial network, this paper selects random forest as the base processor for sequential feature selection and uses 5-fold cross-validation and the AUC value as the evaluation index for the model to select the optimal number of features autonomously. As shown in Figure 6, the model prediction effect is optimal when the number of features is 13, and the features are “V1”, “V2”, “V4”, “V8”, “V14”, “V16”, “V17”, “V18”, “V19”, “V25”, “V27”, “V28”, and “Amount”.

3.3. **Evaluation indicators.** To validate the effectiveness of machine learning algorithms and deep learning algorithms, it is not sufficient to objectively evaluate the model based on its performance on the training set alone; it is also necessary to evaluate the validity of the model on the test set, that is, to evaluate the effectiveness of the generalization performance of the model. The common dichotomous classification problem is usually based on accuracy as a criterion, but the accuracy becomes somewhat one-sided owing to the class imbalance ratio; therefore, the confusion matrix and its derived metrics are currently mainly used for the effective evaluation of imbalanced data classification problems. The confusion matrix divides the samples into confusion matrices to judge the reliability of the experimental results according to their true categories and the predicted categories of the classifier, and the confusion matrix of the two classification problems is presented in Table 2.

TABLE 2. Confusion matrix of classification results

Predicted results	**Real situation**	
	True	False
The prediction is true	TP	FP
The prediction is false	FN	TN

Using the values in Table 2, the values of the derived metrics of the confusion matrix, accuracy, precision, recall, and F1-score can be derived to better understand the performance of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN},$$
$$\text{Precision} = \frac{TP}{TP + FP},$$
$$\text{Recall} = \frac{TP}{TP + FN} \text{ and}$$
$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Based on the notation in Table 2, the true positive rate (TPR) and false positive rate (FPR) are defined as

$$TPR = \frac{TP}{TP + FN} \text{ and } FPR = \frac{FP}{TN + FP}.$$

The samples are ranked according to the prediction results of the LGB classifier, and the samples are predicted as positive examples one by one in this order, and the ROC curves are plotted to evaluate the generalization performance of the binary classifier. Although the ROC curves are robust in objectively identifying better classifiers even when the category distribution is significantly changed, However, in the problem of category imbalance, the large number of negative examples causes the FPR to grow insignificantly, resulting in the ROC curve presenting an overly optimistic estimate of the effect. The PR curve can be plotted with recall on the horizontal axis and precision on the vertical axis, and both metrics focus on positive cases; therefore, the PR curve is widely considered superior to the ROC curve in this case.

In summary, the accuracy, precision, recall, F1-score, and AUC values were selected as evaluation indices, and the ROC and PR curves were used to examine the generalization performance of the model. The model's capacity to generalize is enhanced by the selection of its hyperparameters using the "grid search + 5-fold cross-validation" method.

4. Experiments and Results Analysiss

In this section, the proposed method is compared with three classical machine learning methods and three popular integrated learning methods using a real credit card fraud dataset. The comparison results verify the superior performance of the conditional generative adversarial network-based fraud detection method and further analyze and demonstrate the effectiveness of the anomaly detection method from three perspectives: data enhancement effectiveness analysis, generative adversarial network model comparison analysis, and ablation analysis.

Six imbalanced dataset processing methods were compared in data augmentation experiments, including four oversampling algorithms and two integrated sampling algorithms, to verify the effectiveness of fraudulent credit card transaction data generated by conditional generation adversarial networks. The problems of poor classifier accuracy and generalization performance caused by the extreme imbalance between normal and fraudulent credit card transaction data can be solved using data

Table 3. Hyperparameters of Conditional Generative Adversarial Networks

CGAN Parameter	Value	CGAN Parameter	Value
Latent Size	13	Number of Features	13
Number of Classes	2	Embedding Size	2
Batch Size	1024	Learning Rate	0.001
Epochs	10000		

augmentation. Compared with other generative adversarial models, including the GAN and WGAN, the validation conditional generative adversarial network can quickly reach the convergence condition during the training process and has the advantages of less model collapse and easier control of the training process, which can effectively improve the accuracy of credit card fraud classification.

4.1. **Experimental setup.** In this experiment, the training of a fraud detection model based on CGAN consisted of a generator model, a discriminator model, and a LGB classification model. During the model training phase, since fraudulent transactions in the dataset account for only 0.172% of all transactions, resulting in an extremely unbalanced data distribution, the training process was designed to avoid neglecting the minority fraudulent class. Therefore, we selected 10,000 randomly chosen normal transactions from the training set along with a mixture of fraudulent transactions as inputs for the CGAN model. The potential space sample batch training size was set to 1024, with the binary cross-entropy loss function used to measure the difference between predicted and actual classes. Both the generator and discriminator models were optimized using the ADAM optimizer with an initial learning rate of 0.001, and the model was trained for 10,000 epochs. Given the highly imbalanced dataset, the evaluation metric for the grid search was the AUC to better reflect the model's performance in distinguishing between fraudulent and non-fraudulent transactions.

For hyperparameter tuning, the choices of latent size, batch size, and other parameters were based on a series of experiments to achieve optimal performance. The latent size was set to 13, which was empirically found to offer a good balance between model complexity and ability to generate meaningful synthetic samples for the minority class. The batch size of 1024 was chosen to ensure that the model had enough data to learn from at each training step without overburdening the system memory. A learning rate of 0.001 was selected as it is a common starting point for ADAM optimizer, and after testing, it demonstrated stable convergence. The 10,000 epochs were chosen based on previous studies showing that enough iterations were required to achieve convergence given the complexity of the data. The hyperparameters of the conditional generative adversarial network are listed in Table 3.

4.2. **Analysis of comparative results.**

4.2.1. *Fraud detection algorithm performance comparison.* In this study, PyCharm2022.3.1 (Community Edition) was selected as the compilation environment, and the accuracy, precision, recall, F1-score, and AUC values were selected as evaluation metrics. The original dataset was divided into a 70% training set and 30% test set using the Train_test_split function in the Scikit-Learn machine learning library. The functions in the Scikit-Learn library were then called to train three classical fraud-detection models and three popular integrated learning models using a normalized training set. To avoid data leakage, only the hyperparameters in the models are selected using "grid search + 5-fold cross-validation" on the normalized training dataset. The generalization performance of each base classification model was evaluated using a standardized test set, and the results obtained for each classification model are listed uniformly for comparison purposes, as presented in Table 4.

TABLE 4. Comparison of classifier test results

Model	Accuracy(%)	Recall(%)	Precision(%)	F1-Score(%)	AUC(%)
LR	99.91	59.46	86.27	70.4	95.63
SVM	99.94	**75.68**	89.60	82.05	93.64
KNN	99.94	72.97	90.00	80.60	91.87
XGBoost	99.92	66.22	85.96	74.81	95.98
AdaBoost	99.90	55.41	83.67	66.67	96.15
LGB	**99.95**	74.32	**94.83**	**83.33**	**97.06**

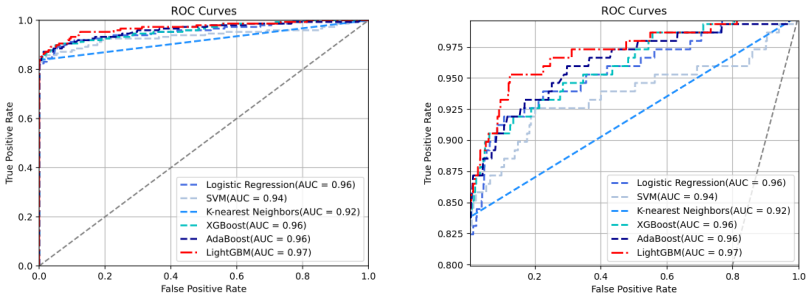

FIGURE 7. ROC curve. (a) Original ROC curve. (b) ROC curve with TPR greater than 0.80

The comparison tests revealed that LGB has high accuracy, precision, F1-score, and AUC values. The base classifiers LR, XGBoost, and AdaBoost have higher AUC values, but the recall of XGBoost and AdaBoost is lower, and the computation of LR on large-scale data consumes a large amount of machine memory and computing time; so LR, these methods are not suitable for handling credit card fraud detection. For the dataset of historical credit card transactions, the AUC values of the base classifiers KNN and SVM were significantly lower than those of the other base classifiers, and the classification results were poor.

Figure 7(a) shows the ROC curves of each base classifier, and to improve the visualization, a part of the TPR greater than 0.80 is enlarged to obtain Figure 7(b). As shown in Figure 7(b), according to the area under the ROC curve, i.e., the AUC value, it can be seen that the AUC value of LGB is 0.97, slightly higher than that of XGBoost and AdaBoost. Although ROC plots are widely used to evaluate the classification performance and generalization ability of classifiers, the ROC curves are overly optimistic in the case of class imbalance. The PR curves are relatively more informative, and the closer the PR curve is to the upper right, the better the model performance. From Figure 8, the PR curve of the base classifier LGB completely "wraps" the rest of the base classifiers, so it is asserted that the performance of LGB is better compared with the other classifiers. Although the AUC areas under the ROC curves of LR, AdaBoost, and XGBoost were all 0.96, the AP areas under the PR curves were 0.68, 0.68, and 0.78, respectively, and the PR curves were relatively far from the upper right and had considerable room for improvement. It can be seen that the ROC curve in the category imbalance problem provides a more optimistic estimate, whereas the PR curve constantly reveals the influence of FP because of its precision.

Combining the results in Table 4, Figure 7, and Figure 8, LGB can effectively improve generalization performance and has a good ability to handle samples with high dimensionality. For historical credit

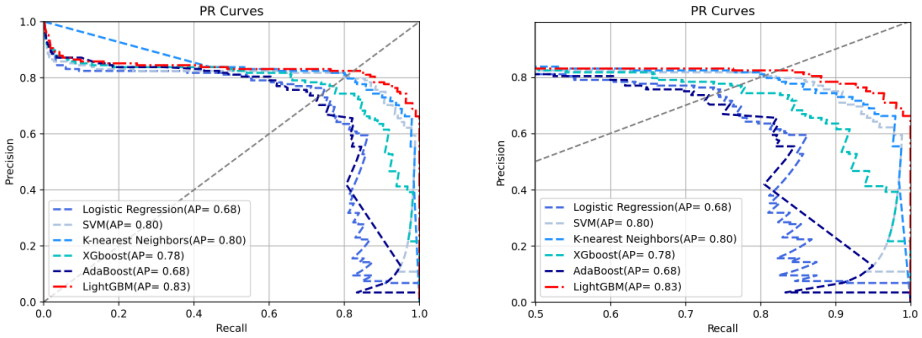

FIGURE 8. PR curve. (a) Original PR curve. (b) Recall greater than 0.5 PR curve

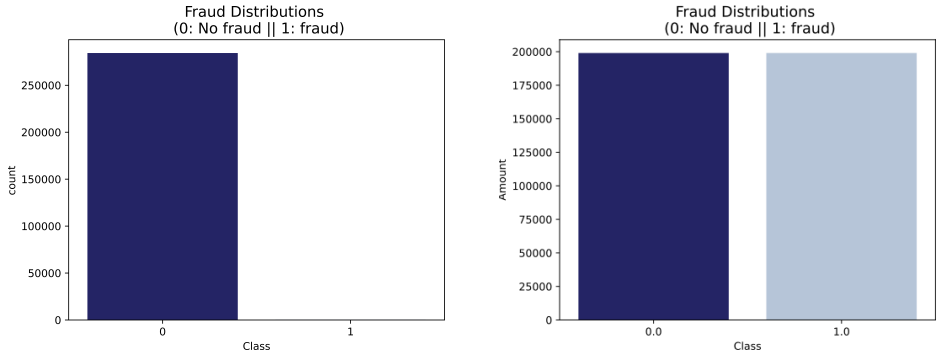

FIGURE 9. Histogram comparing data distribution before and after CGAN generated model for data enhancement. (a) Original data distribution. (b) Distribution of CGAN data after data enhancement

card transaction fraud detection, one prefers to find more fraudulent transactions while not wanting to have a high false positive rate. Although KNN and SVM have powerful classification abilities, the training time is too long as the number of samples increases, which is not suitable for large data processing. LGB combines the unilateral gradient sampling algorithm and the mutually exclusive feature bundling algorithm and transforms the traversal samples into a traversal histogram and depth-first splitting strategy using the histogram optimization strategy, which greatly reduces the time complexity and has the advantages of easy parameter setting for the model, higher prediction accuracy for classification problems, and support for parallel computing, among many other machine learning methods. Therefore, more research directions are placed in LGB, and LGB is chosen as the base classifier of the model.

4.2.2. *Data Enhancement Validity Analysis.* To avoid problems such as classifier failure or overfitting due to unbalanced datasets, the generative model in the conditional generative adversarial network is used to generate fraudulent transaction data. The synthetic fraudulent data were combined with the original training data to construct a new training set to solve the overfitting problem caused by unbalanced data. A comparison of the data distribution before and after data enhancement is shown in Figure 9.

To verify that the effectiveness of the fraudulent credit card transaction data synthesized by the generator in the conditional generative adversarial network and the shortcomings of misclassification of

most class samples by traditional methods can be effectively overcome by data enhancement, the experimental results of this study were compared with six imbalanced dataset processing methods combined with LGB, including four oversampling algorithms (SMOTE, SvmSMOTE, BorderlineSMOTE, and ADASYN) and two integrated sampling algorithms (SMOTEEN and SMOTETomek). The parameters of LGB remained the same as before, and the detection results are listed in Table 5.

The validity of the design of the key components in the model was verified by ablation experiments, and the results of the comparison of the model with and without CGAN for each index are shown in Figure 10. However, we observed that the recall rate (78.38%) remains suboptimal compared to other metrics, such as precision (95.08%). This discrepancy can be attributed to the highly imbalanced nature of the dataset, where fraudulent transactions make up a very small fraction of the total transactions. As a result, the model may focus more on precision, achieving higher accuracy in predicting non-fraudulent transactions, while struggling to identify the minority fraudulent class.

In future work, several strategies could be explored to improve recall and balance it with precision. One approach is class weighting, where higher weights are assigned to fraudulent transactions during training to prioritize the detection of the minority class. Another strategy is resampling, either through over-sampling fraudulent transactions or under-sampling non-fraudulent ones, to address class imbalance. Additionally, hybrid loss functions such as the F1 score could be used to combine precision and recall, guiding the model to better balance these metrics. Lastly, model enhancements, including advanced CGAN architectures or regularization techniques, could be investigated to improve the model's ability to generalize and detect the minority class. These strategies can help mitigate the issue of suboptimal recall and contribute to a more balanced performance in future iterations of the model.

Table 5. Comparison of detection results of unbalanced oversampling algorithms

Model	Accuracy(%)	Recall(%)	Precision(%)	F1-Score(%)	AUC(%)
LGB	99.95	74.32	94.83	83.33	97.06
SMOTE+LGB	99.94	**81.76**	81.21	81.48	**97.21**
SvmSMOTE+LGB	99.95	**81.76**	88.97	85.21	96.67
BorderlineSMOTE+LGB	99.94	79.73	87.41	83.39	97.12
ADASYN+LGB	99.93	80.41	77.27	78.81	97.18
SMOTEENN+LGB	99.88	**81.76**	62.37	70.76	97.05
SMOTETomek+LGB	99.94	**81.76**	81.21	81.48	**97.21**
CGAN-LGB	**99.96**	78.38	**95.08**	**85.93**	95.97

In this study, the samples are divided into confusion matrices to judge the reliability of the experimental results according to their real categories and the predicted categories of the classifier, and the values of the underlying indicators in the confusion matrix of each model are shown in Table 6. Meanwhile, the confusion matrices of each model are plotted, taking the SMOTE+LGB model, the BorderlineSMOTE+LGB model, the SMOTEENN+LGB model, and the CGAN-LGB fraud detection model as examples, as shown in Figures 11, 12, 13, and 14.

A comparison of Figures 11, 12, 13, and 14 reveals that the direct use of the oversampling algorithm improves the accuracy for a few classes of samples but leads to serious misclassification cases for most classes of samples, and the false positive rate is significantly increased. From Table 6, the false positive rate in the LGB model is 0.007% and the number of FPs is 6, but the error rate in this model for the prediction of minority class samples is as high as 25.6757% and the number of FNs is 38; in the SMOTE+LGB model, BorderlineSMOTE+LGB model and ADASYN+LGB model, the false positive rates are 0.0328%, 0.0199%, and 0.041%, and the number of FPs are 28, 17 and 35, respectively; in the SMOTEENN+LGB model, the false positive rate is as high as 0.0856%, and the number of FPs is 73; as can be

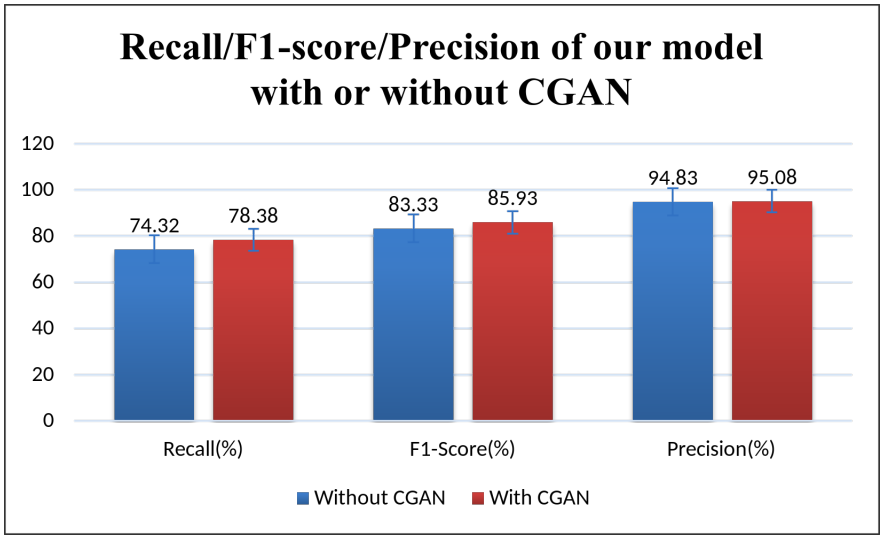

FIGURE 10. Ablation analysis diagram

TABLE 6. Base indicators of the confusion matrix

Model	FP(pcs)	FN(pcs)	False positive rate (%)	False negative rate (%)
LGB	6	38	0.0070	25.6757
SMOTE+LGB	28	27	0.0328	18.2432
SvmSMOTE+LGB	15	27	0.0176	18.2432
BorderlineSMOTE+LGB	17	30	0.0199	20.2703
ADASYN+LGB	35	29	0.0410	19.5946
SMOTEENN+LGB	73	27	0.0856	18.2432
SMOTETomek+LGB	28	27	0.0328	18.2432
CGAN-LGB	6	32	0.0070	21.6216

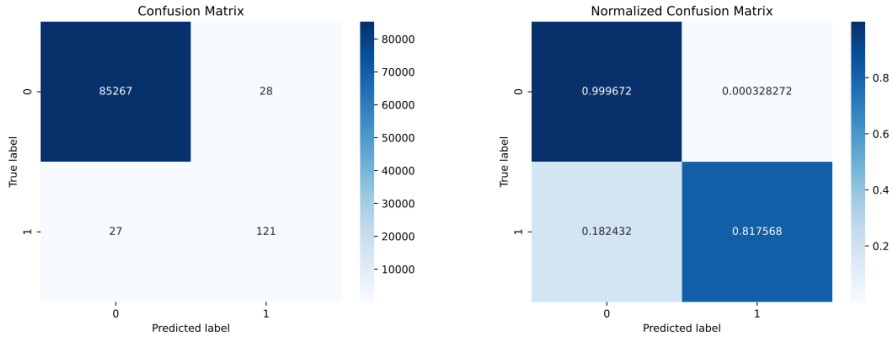

FIGURE 11. Schematic of confusion matrix of SMOTE+LGB model. (a) Confusion Matrix. (b) Normalized Confusion Matrix

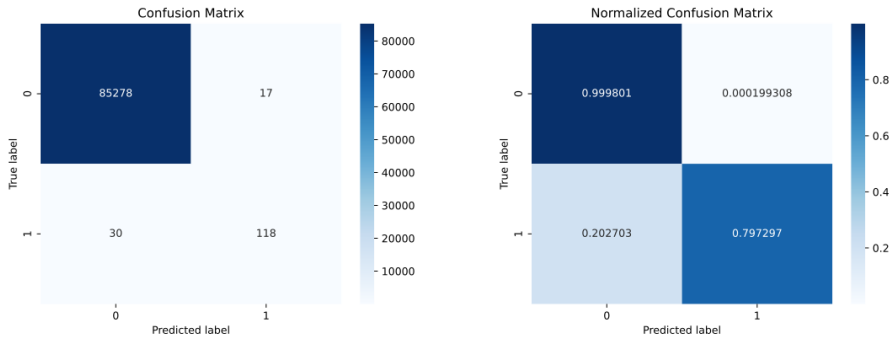

FIGURE 12. Schematic of the BorderlineSMOTE+LGB model confusion matrix. (a) Confusion Matrix. (b) Normalized Confusion Matrix

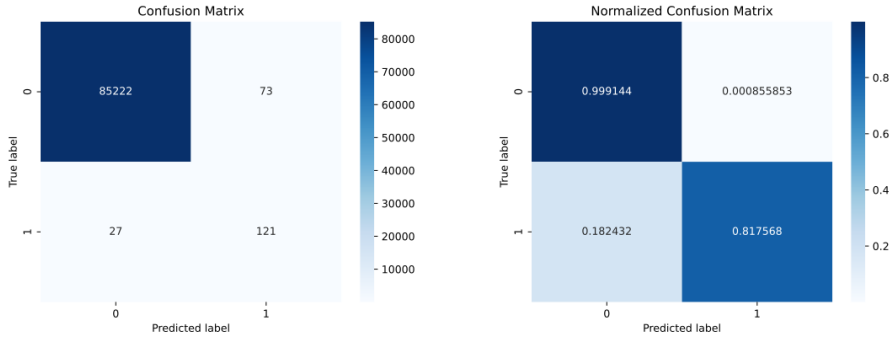

FIGURE 13. Schematic of the SMOTEENN+LGB model confusion matrix. (a) Confusion Matrix. (b) Normalized Confusion Matrix

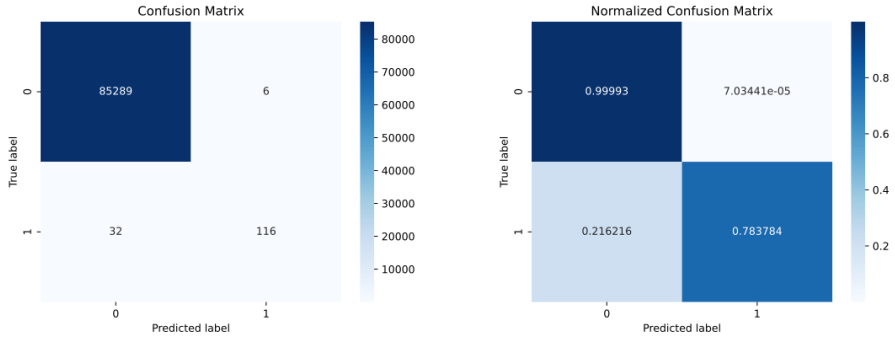

FIGURE 14. Schematic of the CGAN-LGB model confusion matrix. (a) Confusion Matrix. (b) Normalized Confusion Matrix

seen from Figure 14, the false positive rate based on the CGAN-LGB fraud detection model is 0.007%, the number of FPs is 6, and the prediction for a few classes of samples The error rate is 21.6216% and the number of FNs is 32.

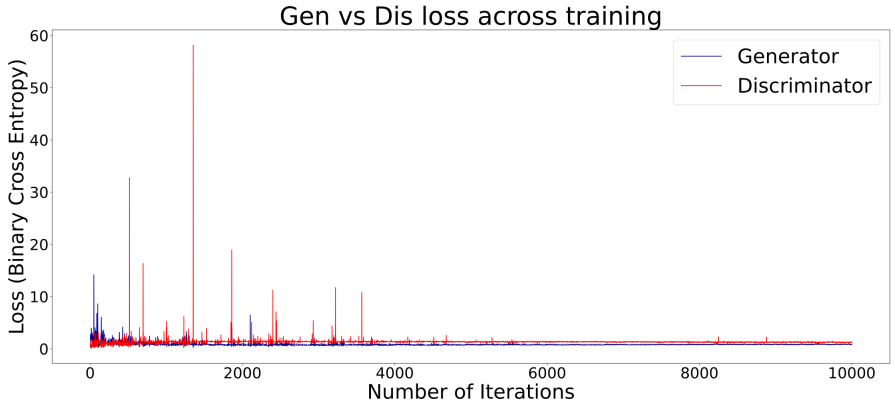

FIGURE 15. Loss diagram of data generation model and discriminator in the conditional generative adversarial network

4.2.3. *Generating Adversarial Network Models for Comparative Analysis.* The loss function plots of the generative and discriminators in the conditional generative adversarial network, as illustrated in Figure 15, were used to guide the hyperparameter settings throughout the conditional generative adversarial training process. As can be observed, after 10,000 training rounds, the conditional generative adversarial model gradually converges, reaches the Nash equilibrium, and finishes training. The loss values of the generator and discriminator no longer fluctuated significantly after reaching a specific value. The conditional generative adversarial network may now produce data samples that satisfy the condition variables and are consistent with the distribution of the original sample data.

To evaluate the authenticity of the synthesized data, this study synthesizes 6000 transaction data using a generator trained in a conditional generative adversarial network and then visualizes them with the real 6000 transaction data samples by PCA and Tsne dimensionality reduction techniques, where PCA maps n-dimensional features to k-dimensions and constructs k-dimensional features again based on the original n-dimensional features in an attempt to preserve the global structure of the data, whereas t-SNE converts the similarity between data points into joint probabilities and tries to preserve the local structure by minimizing the KL scatter between the joint probabilities of the low-dimensional embedded data and high-dimensional data.

In this study, we plotted the distribution of the original and synthesized data after dimensionality reduction in PCA and TSNE using the decomposition and manifold packages in Scikit-Learn. Figure 16 shows that the data synthesized by conditional generative adversarial networks in the credit card historical transaction dataset have better coverage, and the model has better feature extraction capability. The results of the data synthesized using the different generative adversarial networks for credit card fraud detection are listed in Table 7. Compared with the traditional GAN and Unrolled GAN, the CGAN-LGB-based fraud-detection model improved the recall by 8.15% and F1-score by 4.87% and 0.34%, respectively. Compared with WGAN+LR and WGAN+LR, the model has a substantial improvement in all Compared with WGAN+LR and WGAN+LR, the model has a significant improvement in all indicators; compared with WGAN+ANN and RWGAN+ANN, the F1-score of the model has a significant improvement.

5. CONCLUSION AND FUTURE WORK

In this study, we propose a new fraud-detection model based on a conditional generative adversarial network that only requires the extraction of a small amount of transaction data with categorical

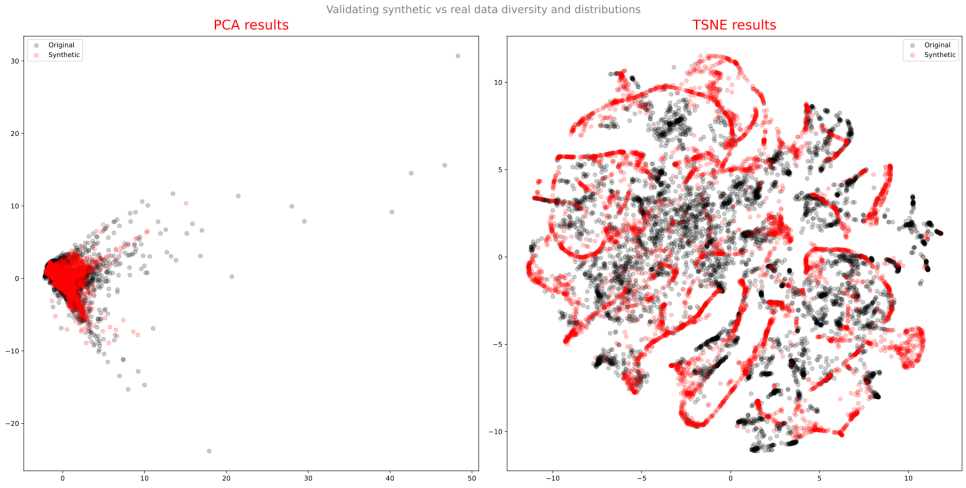

FIGURE 16. Distribution of original data and synthetic data after dimensionality reduction of PCA and TSNE

TABLE 7. Comparison results of different generative adversarial network models

Model	Accuracy(%)	Recall(%)	Precision (%)	F1-Score(%)	AUC(%)
WGAN+ANN(Sethia, A)	99.96	91.2	-	78.52	-
RWGAN+ANN(Sethia, A)	99.96	**94.35**	-	80.45	-
GAN(Fiore, U)	99.96	70.23	**95.83**	81.06	-
WGAN+LR(Ba, H)	-	80.30	50.00	58.30	94.20
WCGAN+LR(Ba, H)	-	64.20	85.20	71.00	94.80
Unrolled GAN(Mirza, M)	-	-	-	85.59	-
CGAN-LGB	**99.96**	78.38	95.08	**85.93**	**95.97**

labels from the training set and then generates a large number of fraudulent transactions for data enhancement. Because the conditional generative adversarial network adds conditional variables in the modeling of both generative and discriminators, it makes the training process easier to control, the model is less likely to collapse, and it can quickly reach the convergence condition and be used to guide the data generation process so that the synthetic fraud-class data can handle problems, such as extreme data imbalance. The empirical analysis of the publicly available historical credit card transaction dataset showed that the model achieved 99.96% accuracy, 95.08% precision, and a 95.97% AUC value. Compared to other fraud detection methods, the model can overcome the defects of traditional methods in misclassifying most classes of samples and significantly improve the efficiency of enterprises in identifying transaction fraud, which has a significant early warning effect.

In the data enhancement validity comparison experiments with four oversampling algorithms and two integrated sampling algorithms, the results show that the fraud-detection model based on conditional generative adversarial networks has a false positive rate of 0.007%, six FPs of 6, an error rate of 21.6216% for minority class sample prediction, and 32 FNs, which has a significant advantage in both accuracy rate and F1-score, but is slightly lower than other imbalance algorithms in terms of recall rate. The effectiveness of the design of key components in the model is verified through ablation experiments, which corroborate that the conditional generative adversarial network has better feature extraction ability and better coverage of the synthesized data. In the generative adversarial network model comparison experiments, compared with traditional GAN and Unrolled GAN, the CGAN-LGB-based fraud

detection model improves recall by 8.15% and the F1-score by 4.87% and 0.34%, respectively; compared with WGAN+LR and WGAN+LR, the model has substantial improvements in all metrics; compared with WGAN +ANN and RWGAN +ANN, the F1-score of this model is significantly improved.

The proposed model applies deep learning techniques to the binary classification problem of credit card fraud detection through cyclic organic structured fusion. It overcomes the shortcomings of traditional methods that often misclassify most classes of samples. By broadening the application of conditional generative adversarial networks (CGANs), the model enhances the diversity of methods for classifying imbalanced samples and demonstrates superior fraud detection performance. This approach provides both a theoretical foundation and practical guidance for financial institutions seeking to apply deep learning techniques in fraud detection. However, since the proposed method falls under supervised learning, it has certain limitations, such as the challenges of transfer learning and high memory usage during the training of conditional GANs, which require substantial computational power. In future work, further research is needed to explore the theoretical foundations of neural networks and GANs to improve model interpretability. This will help decision-makers in the empirical analysis of the detection results produced by these algorithms. Additionally, further studies could investigate the application of semi-supervised and unsupervised learning methods to enhance the model's performance and versatility.

Statements and Declarations

This work was supported by the Shanghai Philosophy and Social Science Planning Project under Grant No. 2024BGL001.

Acknowledgments

We would like to thank Worldline and the ULB (Université Libre de Bruxelles) machine learning group for collecting and sharing the credit card fraud detection dataset, which made it possible to conduct the relevant empirical study in this paper. We also would like to thank Editage (www.editage.cn) for English language editing.

References

[1] S. Alla and S. K. Adari. *Beginning Anomaly Detection Using Python-Based Deep Learning*. Apress, New Jersey, 2019.

[2] K. Choi, J. Yi, C. Park, and S. Yoon. Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. *IEEE Access*, 9:120043–120065, 2021.

[3] A. A. Taha and S. J. Malebary. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. *IEEE Access*, 8:25579–25587, 2020.

[4] L. Zheng, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li. Improved TrAdaBoost and its application to transaction fraud detection. *IEEE Transactions on Computational Social Systems*, 7(5):1304–1316, 2020.

[5] E. Ileberi, Y. Sun, and Z. Wang. A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, 9(1):1–17, 2022.

[6] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pages 1–9. Lagos, Nigeria, 2017, IEEE.

[7] W. Yang, Y. Zhang, K. Ye, L. Li, and C. Z. Xu. Ffd: A federated learning based method for credit card fraud detection. In K. Chen, S. Seshadri, L.-J. Zhang, editors, *Big Data–BigData 2019: 8th International Congress, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019*, Proceedings 8, pages 18–32. Springer, 2019.

[8] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla. Credit card fraud detection-machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. East Sarajevo, Bosnia and Herzegovina, 2019, IEEE.

[9] A. Chouiekh and E. H. I. E. Haj. Convnets for fraud detection analysis. *Procedia Computer Science*, 127:133–138, 2018.

[10] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, and Y. Qi. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 598–607, Beijing, China, 2019, IEEE.

[11] B. Branco, P. Abreu, A. S. Gomes, M. S. Almeida, J. T. Ascensão, and P. Bizarro. Interleaved sequence rnns for fraud detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3101–3109, August 2020.

[12] Z. Li, G. Liu, and C. Jiang. Deep representation learning with full center loss for credit card fraud detection. *IEEE Transactions on Computational Social Systems*, 7(2):569–579, 2020.

[13] X. Zhang, Y. Han, W. Xu, and Q. Wang. HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences*, 557:302–316, 2021.

[14] J. Forough and S. Momtazi. Ensemble of deep sequential models for credit card fraud detection. *Applied Soft Computing*, 99:106883, 2021.

[15] A. Sethia, R. Patel, and P. Raut. Data augmentation using generative models for credit card fraud detection. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6, Greater Noida, India, 2018, IEEE.

[16] H. Ishfaq, A. Hoogi, and D. Rubin. TVAE: Triplet-based variational autoencoder using metric learning. *arXiv preprint arXiv:1802.04403*, 2018.

[17] 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. *Information Sciences*, 479:448–455, 2019.

[18] H. Ba. Improving detection of credit card fraudulent transactions using generative adversarial networks. *arXiv preprint arXiv:1907.03355*, 2019.

[19] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling Tabular data using Conditional GAN. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7335 - 7345, 2019, Curran Associates Inc, New York, United States.

[20] J. Wang and L. Yao. Unrolled gan-based oversampling of credit card dataset for fraud detection. In *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 858–861, Dalian, China, 2022, IEEE.

[21] C. Yu, Y. Xu, J. Cao, Y. Zhang, Y. Jin, and M. Zhu. Credit card fraud detection using advanced transformer model. *In 2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*, pages 343–350, Hong Kong, China, 2024.

[22] Y. Tian and G. Liu. Spatial-Temporal-Aware Graph Transformer for Transaction Fraud Detection. *IEEE Transactions on Industrial Informatics*, 20(11):12659–12668, 2024.

[23] M. Mirza and S. Osindero. Conditional generative adversarial nets. *ArXiv Preprint*, arXiv:1411.1784, 2014.

[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, and T. Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3149-3157, Long Beach, 2017.

[25] Kaggle.com. Credit Card Fraud Detection. [online] Available at: https://www.kaggle.com/mlg-ulb/creditcardfraud [Accessed December 9, 2023].