Hindawi
Mobile Information Systems
Volume 2022, Article ID 9103437, 14 pages
https://doi.org/10.1155/2022/9103437

Research Article

A ResNet-LSTM Based Credit Scoring Approach for Imbalanced Data

Anqin Zhang,[1] Baicheng Peng,[1] Jingjing Chen,[2] Qingfu Liu,[2] Shibo Jiang,[3] and Youmei Zhou[4]

[1]*College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China*
[2]*School of Economics, Fudan University, Shanghai, China*
[3]*School of Hotel and Tourism Management, The Hong Kong Polytechnic University, Hong Kong SAR, China*
[4]*College of Architecture and Urban Planning, Tongji University, Shanghai, China*

Correspondence should be addressed to Youmei Zhou; 20310231@tongji.edu.cn

Received 23 February 2022; Revised 21 March 2022; Accepted 29 March 2022; Published 26 April 2022

Academic Editor: Yan Huang

Detecting potential defaults or bad debt with limited information has become a huge challenge. The main difficulties faced by the credit scoring are sample imbalance and poor classification performance. For this reason, we first proposed the auxiliary conditional tabular generative adversarial network (ACTGAN) to generate sufficient default transaction samples from the original data, then we designed a model based on ResNet-LSTM used for feature extraction, which includes two submodels of ResNet and LSTM to extract static local features and dynamic temporal features from the original data, respectively. After that, a spatiotemporal attention module is added to calculate the importance of the two submodel's output in order to extract more critical information. Finally, we applied the focus loss function into the XGBoost classifier to improve the probability output of the credit default risk. We verified the designed credit scoring model in two real-world datasets. The experimental results showed that ACTGAN can effectively solve the problem of data imbalance. The ResNet-LSTM+XGBoost model for classification is better than other traditional algorithms in F1 value, AUC, and KS value, which proves the effectiveness and portability of this model in the field of credit scoring.

1. Introduction

With the advancement of computer science and technology, online financial service is booming worldwide. While the industry is booming, the amount of data in the financial industry is showing explosive growth, and the increase in consumer credit demand has also brought huge amount of financial fraud incidents, which will seriously damage consumers and financial platforms. Therefore, in order to minimize the losses of platforms and consumers, many researchers have conducted studies, and proposed a large number of models to predict the credit risk level of online loan customers and avoid the occurrence of default or bad debt.

Credit scoring can be summarized as a binary classification problem, which predicts the default probability of loan applicants and divides a loan into default or nondefault [1], thereby helping financial institutions make appropriate decisions. Meantime, credit scoring is also an imbalanced classification problem. The number of default samples is significantly lower than the normal nondefault samples. However, the identification of default samples is what we should focus on [2]. The misclassification of fraudulent transactions as normal is much more damaging than detecting a normal transaction as fraud in credit scenario [3]. Therefore, researchers are trying to improve the accuracy rate of default sample classification in recent years. The current mainstream methods for dealing with imbalanced datasets can be divided into three ways [4]: the first is based on data preprocessing, such as undersampling and oversampling methods, by eliminating samples from the majority class or increasing samples from the minority class to change

the proportion of the original imbalanced data. Although these methods alleviate the problem of data imbalance to some extent, deleting the majority class samples inevitably causes information loss and makes the model unable to use the existing information. On the other hand, the method of adding samples to the minority class lacks data diversity, cause overfitting of the model to a certain extent.

The second way pays attention to feature extraction utilizes machine learning techniques to mine hidden features of data. Models such as multilayer perceptron [5], convolutional neural network [6], and deep belief network [7] have been widely used in data mining in the field of credit scoring. However, an obvious drawback is that existing models exclusively regard static features and dynamic features as a whole feature space as the input of neural network, and ignored time dependencies of user behavior data. For this reason, our proposed model incorporates both static features and dynamic time-based data into model input, which captures critical information from multi-source heterogenous credit data.

The third way aims to improve the performance of classification algorithm, among which cost-sensitive learning and ensemble learning methods are particularly prominent. The cost-sensitive learning reduces the biased error towards the negative class and improves the recall rate of the positive class. Ensemble learning is a machine learning approach where multiple learners are trained to solve the classification problem. The central concept is to combine several "weak leaners" into a "strong learner", thereby eventually boosting the performance of classifiers. However, the improvement at the algorithm level only assigns more classification weights to the minority class samples, which prone to overfitting, and does not solve the problem of the scarcity of positive samples.

Since it is difficult for a single method to satisfy the requirements of different imbalanced datasets, the applicability is generally not strong. At the same time, the combined model can take advantages of each single credit scoring method, thereby our research considers all three levels of imbalanced credit scoring. We firstly propose an auxiliary conditional tabular generative adversarial network (ACTG AN) to alleviate the class imbalance problem in credit scoring task. Specifically, our ACTGAN which uses the Wasserstain distance to define the difference between the real data and the generated data. An auxiliary classifier is added to discriminator to stabilize generator's output. Gradient penalty is introduced to optimize the loss function. Crosslayer is added to the network to calculate high-dimensional feature interactions and generating a sufficient number of positive samples to form an enhanced dataset. Results show that ACTGAN can significantly improve the classification performance of the credit default prediction model.

Secondly, a hybrid deep learning feature extraction algorithm is designed, which divided data features into static local features and dynamic temporal features. Static financial data is input into a convolutional neural network with residual module, and dynamic feature data is input to LSTM to capture key information on its time series, and attention module is used to calculate the importance of static feature vectors and dynamic feature sequences. The fully connected layer is then applied to fuse two kinds of feature embeddings into a unified latent feature space. Finally, at the algorithm level, the Focal Loss for the imbalanced data classification is used to improve XGBoost and obtain the final output.

The main contributions of our research are as follows. First, a new generative adversarial network ACTGAN is proposed to generate more minority class samples to overcome the class imbalance problem. Second, we employ a feature extraction module to integrate multi-source heterogeneous data into the latent feature space and use attention layer to calculate the importance of information. Third, at the classification algorithm level, we leverage a focal loss function to improve the XGBoost classifier.

The remainder of the paper is structured as follows: Section 2 details related previous literatures on credit scoring. Section 3 describes the proposed framework. Section 4 presents the statistical information of datasets, the evaluation experiments and the result summarizations. The last section draws conclusions and future research directions.

2. Theory and Methods

2.1. GAN for Data Generation. Generative adversarial network (GAN) is an excellent generative model network proposed by Goodfellow et al. [8]. A typical GAN consists of two components: a generator G learns the probability distribution of real data and transforms the input noise z into new synthetic data, a discriminator D tries to distinguish the real samples from generated ones. The competition between G and D can be formalized as a minimax game:

$$\begin{aligned} Score &= \min_{G} \max_{D} V(D, G) \\ &= E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log (1 - D(G(z)))] \end{aligned} \tag{1}$$

In formula (1), p_{data} represents the distribution of real data. p_z represents the distribution of input random noise, $G(z)$ represents generated data, $V(D, G)$ represents the output value of D and G. We train the D to maximize $V(D, G)$. We train the G to minimize $V(D, G)$. Current studies [9] [10] adopt GAN as the over-sampling approach for the minority class to solve the class imbalance problem caused by credit data. They first train the GAN on a specific dataset and then use the well-trained Generator G to produce new synthetic samples. Finally, the original real samples are mixed with the generated ones and used to train the classifier for credit scoring. Although these methods have made remarkable progress, GANs with the vanilla GAN loss are difficult to train. Since the input of the GAN samples from random noise, the generated data is also random and chaotic, and it is impossible to control what category the generated image or data belongs to, which causes model collapsing and nonconvergence. Hence, We extend GAN to solve the class imbalance and mode collapsing in credit

scoring by adding conditional restrictions and gradient penalty.

2.2. Deep Learning for Feature Extraction. Inspired by the success of deep learning in a series of fields, several models started to employ neural networks for credit scoring. Based on the multilayered perceptron (MLP) approach, Blanco et al. [11] built several nonparametric credit scoring models and showed demonstrated excellent performance against traditional models. Metawa et al. [12] utilize a deep belief network (DBN) to model rich and complicated information for credit scoring. Deng et al. [13] adopted convolutional neural network (CNN) to capture the relations among the chosen attributes and output the default probabilities. With the application of deep learning technology, many cross domain research methods are gradually introduced into the field of credit evaluation, such as natural language processing to mine the correlation between lending companies [14], graph neural network to mine the relationship between entities in the social network of credit users [15–17], the autoencoder [18, 19] uses encoding-decoding technology to achieve data dimensionality reduction and other operations, which can achieve deeper mining of data patterns.

2.3. Imbalance Classification. The current imbalanced data processing methods based on ensemble learning mainly combine ensemble learning with other imbalanced data classification processing methods to comprehensively improve the classification effect. For example, Iranmehr et al. [20] proposed a cost-sensitive learning-based structured SVM ensemble classification algorithm, which increased the weight of minority samples and improved the classification accuracy of unbalanced data. Paleologo et al. [21] extended the advantage of the bagging approach, where the training subsets are formed by random sampling to address a class of imbalanced problems. Luo [22] compared the performance of the bagging approach using DT, SVM, K-nearest Neighbor (KNN), and MLP based on an imbalanced and large dataset. They obtained that bagging KNN was more sui than other methods for large and imbalanced datasets in credit scoring. Wang et al. [23] proposed a classification algorithm that combines the undersampling method and cost sensitivity, which improves the classification performance on unbalanced data. Tsai et al. [24] conducted a comprehensive study comparing classifiers ensemble methods for three public credit scoring datasets.

3. Methodology

3.1. ACTGAN. In order to solve the problem of data imbalance, based on the conditional generative adversarial network [25], we proposed a new generative adversarial network structure (ACTGAN) to generate positive samples, as shown in Figure 1. First, same as the original CGAN, the input of generator contains random noise z and a conditional vector c. Since the tabular data has high-dimensional features, a crosslayer is used to calculate the correlation between features before discriminator [26]. Stacking l-cross-layers can better model the relationship between variables and increase the variation of noise.

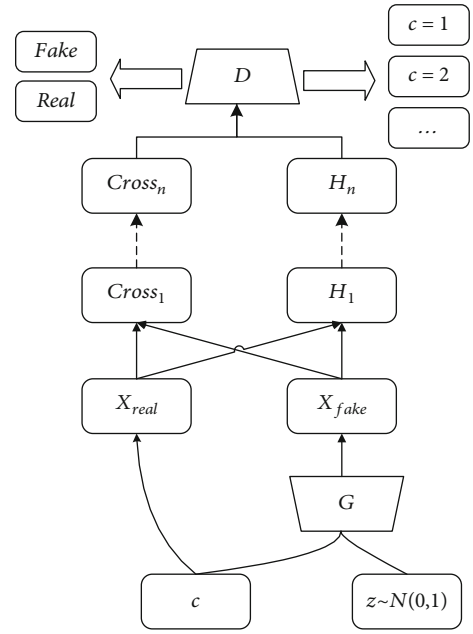


Figure 1: Structure of auxiliary conditional tabular GAN (ACTGAN).

In addition, an auxiliary classifier AC (Auxiliary Classifier) [27] is added to the discriminator D, so that the discriminator outputs both the true-or-false results of data and the classification result $\hat{c}$. The cross-entropy loss of AC is added to the discriminator loss. The output label $\hat{c}$ is compared with the generator input condition vector c to optimize the generator to generate samples that can identify a specific class. The formula is shown in (2), which calculates the binary cross-entropy loss between the true input label c and the predicted label $\hat{c}$, N represents the number of input samples.

$$L_{AC} = E_{c\sim p_{data}}\left[-\frac{1}{N}\sum_{i=1}^{N} c_i \bullet \log(\hat{c}_i) + ((1-c_i)\log(1-\hat{c}_i)\right] \tag{2}$$

In addition, since GAN has always been difficult to train, we use Wasserstein distance to replace the JS divergence in the original generative adversarial network [28–31]. Different from the gradient disappearance phenomenon of JS divergence and KL divergence, W distance has smooth characteristics and can be maximized by using a parameter value range, which can effectively narrow the generated distribution p_g and the real distribution p_{data}. The W distance formula is as follows:

$$W\left(p_g, p_{data}\right) = \frac{1}{K}\sup_{||f||\le K} E_{x\sim p_{data}(x)}(f(x)) - E_{\hat{x}\sim p_{\hat{x}}}(f(\hat{x})) \tag{3}$$

In formula (3), sup represents the least upper bound, and $||f|| \le K$ represents that the function f satisfies the

1-Lipschitz continuity, where the function f can be fitted by a neural network, so K is able to exist by restricting all parameters in the network not to exceed a certain range. Although weight clipping can make the network satisfy the 1-Lipschitz condition, the gradient will disappear due to the restricted weights and improper setting of weight clipping. Therefore, a gradient penalty (GP) is established for the above problem as shown in formula (4). The gradient penalty is a soft constraint, which can control the gradient around 1, alleviate the problem of gradient disappearance, and make the model stable.

$$L_{GP} = \lambda_{GP} E_{\hat{x}\sim p_{\hat{x}}} \left[\left(\left|\left| \nabla_{\hat{x}} D(\hat{x}) \right|\right|_2 - 1 \right)^2 \right] \tag{4}$$

We show the end-to-end training process of our data generation model ACTGAN in Algorithm 1:

(a) First, we manually specify network parameters $[\theta_g, \theta_d]$ and initialize generator G and discriminator D

(b) Sample real data x, condition vector c from real data; sample z from Gaussian-distributed noise data

(c) Generator G generates synthetic data X_{fake}

(d) Fuse real data with fake data as crosslayers' input, the crosslayer and the hidden layer calculate the high-dimensional feature interaction. After the neural network computing of the l_{th} layer, the embedding is obtained by concatenating hidden layers and crosslayers

(e) The embedding is input into the discriminator D to obtain the predicted true-or-false probability $\hat{y}$ and label value $\hat{c}$

(f) Set G constant, we optimize D through devised loss function L_D, in which $E_{x\sim p_{data}(x)}[D(x)] - E_{z\sim p_z(z)}[D(G(z|c))]$ calculating the binary cross entropy between real data and generated data. Auxiliary classifier loss L_{AC} is added so as to stabilize model training and make generated data closer to original ones. Moreover, L_{GP} is used to control the gradient from disappearing

We add The parameters of generator and discriminator are optimized by Adam optimizer, with learning rate $\eta = 0.0001$. The number of crosslayers l is set to 3.

3.2. ResNet-LSTM for Feature Extraction. The credit scoring data is usually a mixture of structured and semi-structured data, which are called multi-source heterogeneous data. We can divide the credit scoring data into two types: user profile data (i.e., gender, occupation and education) and time-based user behavior data (i.e., credit card transactions and previous loans data). Most researches only focus on a single type of data but fail to fuse these two types of data to extract high-level hidden feature. Some researches [32] treat all kinds of data equally and fail to capture the dynamics of user payment behavior over time, while others [33] only focus on user behavior data, not on user profile data that are critical to the credit scoring task. These conventional methods cannot mine and fuse the rich latent information from such multi-source heterogeneous credit data and thus fail to extract high-level hidden feature for credit scoring. In this context, the integration of multi-sources heterogeneous data has been considered as one of the crucial research points for credit scoring.

According to the above, after adding more positive samples to form an enhanced dataset, we propose a hybrid ResNet-LSTM model for the different structural features of the dataset, including two feature extractors, in which ResNet is used to extract static features, and the static features mainly include demographic characteristics (age, gender, height and weight, etc.), financial characteristics (income, property, etc.), dynamic characteristics include the user's operation sequence over a period of time (the number of applications per month, the number of repayments, etc.) and use LSTM to extract the entire business cycle. In order to capture the important information of the feature extraction object and suppress the irrelevant details, the spatial attention (SAM) module and the temporal attention (TAM) module are used to calculate the importance of static feature and dynamic feature vector, respectively, and A series of attention weight parameters are assigned to improve the network performance, and finally a fully connected layer is used for fusion and output to XGBoost to calculate the classification results. The framework flow chart is shown in Figure 2.

3.2.1. ResNet-LSTM. Static features contain a large amount of low-dimensional tabular data, which can be regarded as single-channel images. Convolutional neural networks have certain advantages in processing static image data. Operations such as convolution and pooling can reduce the amount of parameter calculation. Tabular data can be divided into multiple 1D vectors as the input of CNN, and the calculation amount of using 1-dimensional convolution in forward propagation is far less than the traditional CNN that uses images as input.

However, experiments show that the performance of traditional convolutional neural network will begin to degrade with the increase of network layers, which indicates that when the network becomes very deep, the depth network becomes difficult to train. This is because the ReLU function in the neural network only performs nonlinear transformation on the input data. If the low-dimensional data is redundant, huge amount of information will cause the mapping value of the neurons to flow through to be 0, that is, the gradient disappears. The residual network [34, 35] (ResNet) can solve the problem of performance declining. By short-circuiting the identity block and the ReLU layer through shortcuts can flexibly save useful information in the neural network and reduce the information redundancy in the data. Therefore the static features are trained with the residual shortcut structure.

At the same time, the consumer transaction data contains a large number of operation behavior sequences, which can be regarded as time series, and a sequence processor is

Require: The gradient penalty coefficient λ_{GP}, the number of iterations n_{critic}, the batch size m, Adam hyperparameters η
Create G, initialize θ_g, create D, initialize θ_d.
For $t = 1, \cdots, n_{critic}$ **do**
  For $i = 1, \cdots, m$ **do**
    Sample real data $x \sim p_{data}$, random noise $z \sim p(z)$, condition vector $c \sim p_{data}$.

$$X_{fake} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_m) = G(z|c)$$

$$Cross_1, H_1 = [X_{real}, X_{fake}]$$

$$\cdots$$

$$Cross_l = x_0 x_{l-1}^T \omega_{l-1} + b_{l-1} + x_{l-1}$$

$$H_l = W_l x_{l-1} + b_l$$

$$Embedding = \sigma([Cross_l^T, H_l^T] W_{logits})$$

$$\hat{y}, \hat{c} = D(Embedding)$$

$$L_D = E_{x \sim p_{data}(x)}[D(x)] - E_{z \sim p_z(z)}[D(G(z|c))] - L_{GP} + L_{AC}$$

  End for
  Update parameter $\theta_d \longleftarrow \theta_d + \eta \nabla L_D(\theta_d)$
End for

$$L_G = L_{AC} - E_{z \sim p_z(z)}[D(G(z|c))]$$

Update parameter $\theta_g \longleftarrow \theta_g + \eta \nabla L_G(\theta_g)$

Algorithm 1: ACTGAN training algorithm.

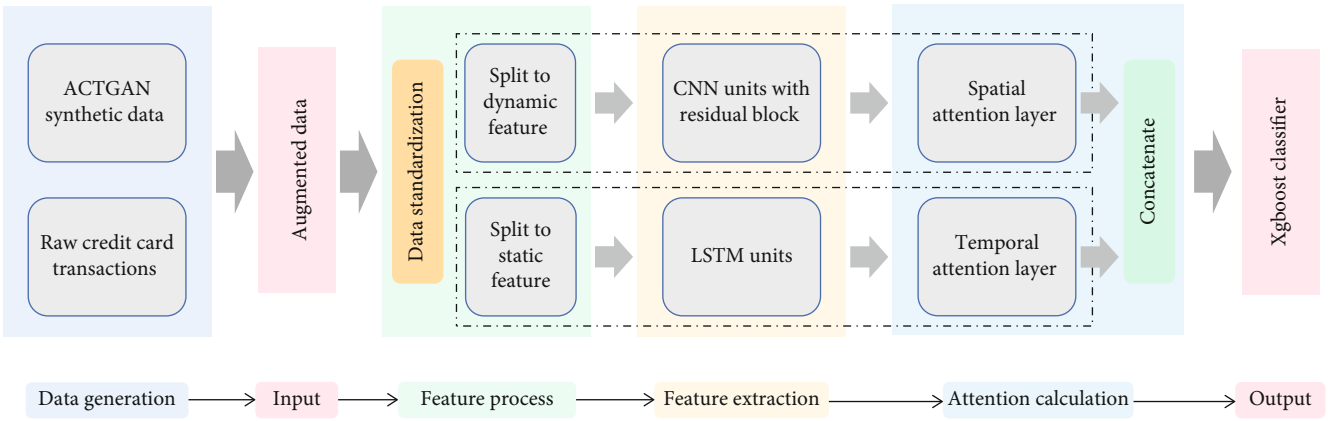


Figure 2: Credit scoring framework.

required to obtain the time embedding. Therefore, BiLSTM is used for modeling dynamic features. The two feature learners output the corresponding feature vectors, which are added to the attention mechanism module to improve model performance. The structure of ResNet-LSTM is shown in Figure 3.

3.2.2. Attention Module. Attention mechanism in the deep learning model mainly consists of two parts: first, determine which part of the entire input needs to be paid attention to; second, extract features from the key parts to obtain important information and sort the output. Attention mechanism can help the model assign different weights to each part of the input sequence, extract more key information and enable the model to make accurate judgments without bringing more computing and storage to the model. In the field of credit scoring literature [36, 37], the bidirectional LSTM network is used to add a temporal attention layer to predict credit card data, which has played a certain role in improving the classification effect of the model. However, these studies only used the temporal attention mechanism and did not consider the impact of nontime series data.

Moreover, different types of attention have different effects on the data. Since we input tabular data and some time series, the hybrid attention mechanism is used to capture importance information for different types of data. Tabular data can be regarded as a single-channel image and introduced into the image. Spatial attention is used to extract features, and time series uses the temporal attention mechanism to output the importance ranking of time points.

Spatial Attention Mechanism is proposed by the image field [38]. The process is shown in Figure 4. First, a global maximum pooling and global average pooling based on feature map X are performed to obtain average pooling feature map $AvgPool(X)$ and max pooling feature map $MaxPool(X)$, respectively, and then the two feature maps are concatenated to generate a valid feature representation vector. At last, through a one-dimensional convolution

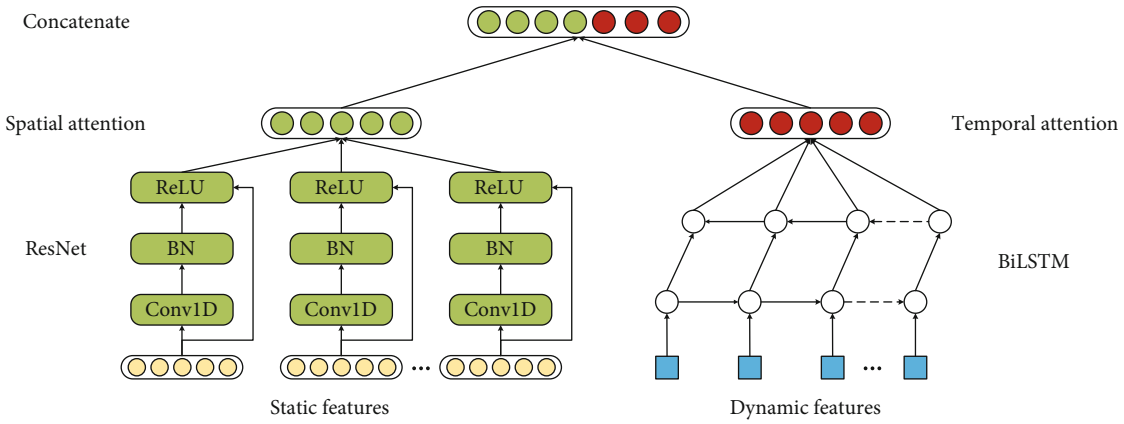


Figure 3: Structure of ResNet-LSTM.

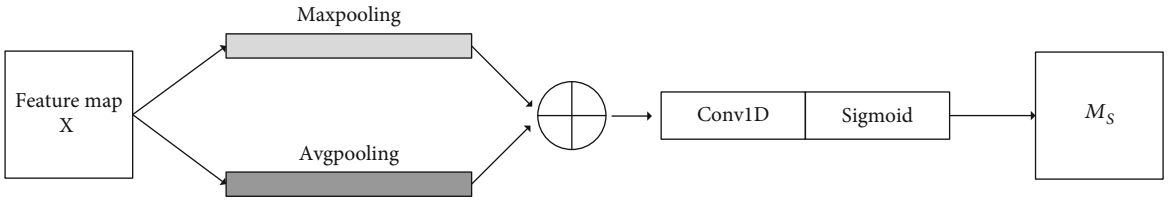


Figure 4: Spatial attention module.

operation f and a sigmoid activate function σ to generate a spatial attention feature vector, that is, M_s. The spatial attention formula is as follows:

$$M_s(X) = \sigma(f([AvgPool(X); MaxPool(X)])) \tag{5}$$

The input of the temporal Attention Mechanism is the hidden state of the BiLSTM at each moment. First, the input vector is encoded as a context vector by the tanh function, and the internal relationship of the time series data is learned by calculating the weight coefficient a, and the attention is allocated to the corresponding time to make the model focus on the more important subsequences. The temporal attention mechanism formula is as follows:

$$\begin{aligned} u_i &= \tanh\ (Wh_i + b) \\ a_i &= \text{softmax}(u_i) \\ M_D &= \sum_{i=1}^{t} a_i h_i \end{aligned} \tag{6}$$

Where h_i represents the output of the i time point of LSTM; t represents the length of the input sequence; a_i represents the weight of the output of the i time point; M_D refers to the weighted total of the LSTM output at each time point.

3.2.3. Focal Loss-XGBoost for Classification. Finally, we linearly fuse the two features learned by the neural network, and use the XGBoost classifier to output the final classification result $output_i$. Default customers are represented by 1, and normal customers are represented by 0. The formula is as follows:

$$\begin{aligned} X &= M_s \bigoplus M_D \\ output_i &= F(X) \end{aligned} \tag{7}$$

We use the Focal Loss function, which is the target detection algorithm, to optimize classification performance. The loss function specially designed for imbalanced classification is mainly added by the cross-entropy function with α coefficient and γ coefficient, so that the loss function is more inclined to focus on the minority class samples, so as to avoid the performance degradation caused by the easy-to-classify samples during the model training process. The formula is shown in (8), $\hat{y}$ represents the output probability of the classifier. When y =1, α is greater than 0.5 and less than 1 can increase the loss of misclassification, and the focusing coefficient γ can adjust the weight of easy-to-classify samples and hard-to-classify samples (γ >0). When y =1, the closer $\hat{y}$ is to 1, the smaller $(1-\hat{y})^{\gamma}$ is, it means that the sample is easier to be classified, the smaller the loss weight of the easy-to-classify sample is, thus make the classifier pay more attention to the hard-to-classify samples.

$$L_{FL} = \begin{cases} -\alpha(1-\hat{y})^{\gamma} \log \hat{y}, y = 1 \\ -(1-\alpha)\hat{y}^{\gamma} \log\ (1-\hat{y}), y = 0 \end{cases} \tag{8}$$

Table 1: Details of datasets used for evaluation.

Dataset	Normal samples	Default samples	Static features	Dynamic features	Imbalance rate
Bank	13307	2009	11	17	0.15
UCI	23364	6636	5	18	0.28

4. Experiments and Result Analysis

4.1. Dataset Description. We employ two real-world customer loan applicant datasets to implement and evaluate the proposed model, the first dataset is user loan data from an anonymous Chinese commercial bank, which contains about 15,000 consumer loan application records, including asset status, personal information, city of residence, equipment used, number of applications and other characteristics, nearly 15% of applicants are default users.

The second dataset, taken from the UCI Machine Learning Repository, is related to 30,000 applicants and transaction payments. It contains customer behavior data for the past 6 to 12 months (e.g. monthly/quarter/year application volume, billing amount, and default history), along with their financial and demographic information such as gender, city of work, age, property, and conditions. Nearly 22% of applicants are default users.

4.2. Metrics and Implementation Details. We divide the data into static features and dynamic features, and use the 0-1 label to indicate whether the customer defaults in the future. The details of the data set are shown in Table 1. We randomly split 70% of the data for training and 30% for testing. When training data, 20% of the data was randomly selected as the validation set. We adopt several metrics commonly used in credit scoring to evaluate the performance of the proposed model: AUC (Area under curve), recall, F1 value, and KS value.

TP (True Positive) indicates the actual default sample and the prediction is also a default, TN (True Negative) indicates that the actual nondefault sample is also predicted to be a nondefault, FP (False Positive) indicates that the actual default sample is predicted to be a nondefault, FN (False Negative) means that samples that are not in default are predicted to default. We calculate the true positive rate (TPR), false positive rate (FPR), F1 values and Recall values, the formula is as follows:

$$\begin{aligned} TPR &= \frac{TP}{TP + FN} \\ FPR &= \frac{FP}{TN + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1 &= \frac{2TP}{2TP + FP + FN} \end{aligned} \tag{9}$$

KS (Kolmogorov-Smirnov) is an evaluation index used to distinguish the degree of separation of positive and negative samples, and is often used in credit scoring models. The predicted outcome for each sample is a probability value in the range 0 to 1. The predicted probability values of positive and negative samples are arranged from smallest to largest, and the KS value is the absolute value of the largest difference between the two distributions. Generally speaking, the larger the KS value, the better the discrimination between positive and negative samples. The formula for the KS value is as follows:

$$KS = \max |TPR - FPR| \tag{10}$$

Table 2: Parameter setting of ResNet-LSTM.

Parameters	Value
batch_size	24/48
Convolution kernel size	16/32/64/128
LSTM units	128
Dropout	0.5
Activation function	ReLU
Optimizer/learning rate	Adam/0.0001
Epoch	100

The specific parameters of the neural network model are shown in Table 2. The convolution kernel is set to 4 layers of 16, 32, 64, and 128 units. In order to prevent overfitting, the drop out value is set to 0.5 to randomly inactivate 50% of the neurons. ReLU activation function and Adam optimizer to speed up convergence, learning rate is set to 0.0001.

4.3. Experimental Results

4.3.1. Imbalance Credit Scoring Result and Analysis. To verify the performance of ACTGAN, we train the model for 10000 iterations and compare with the vanilla conditional generative adversarial network (CGAN). As shown in Figure 5, the generator loss of the ACTGAN generator is lower than that of CGAN from the beginning, indicating that adding the crosslayer can well learn the interactive features of high-dimensional data, improve the ability of network feature extraction and enhance the performance of the generator. For the discriminator, the loss of ACTG AN after 8000 iterations is significantly lower than that of CGAN, which verifies that adding gradient penalty and auxiliary classifiers greatly improves the stability of model training. For W distance, CGAN stabilizes around 1 after 8000 iterations, compared with our proposed model, the Wasserstein distance of ACTGAN stabilizes around 0.6 and fluctuates slightly, indicating that ACTGAN has better convergence stability, and the synthetic data generated by

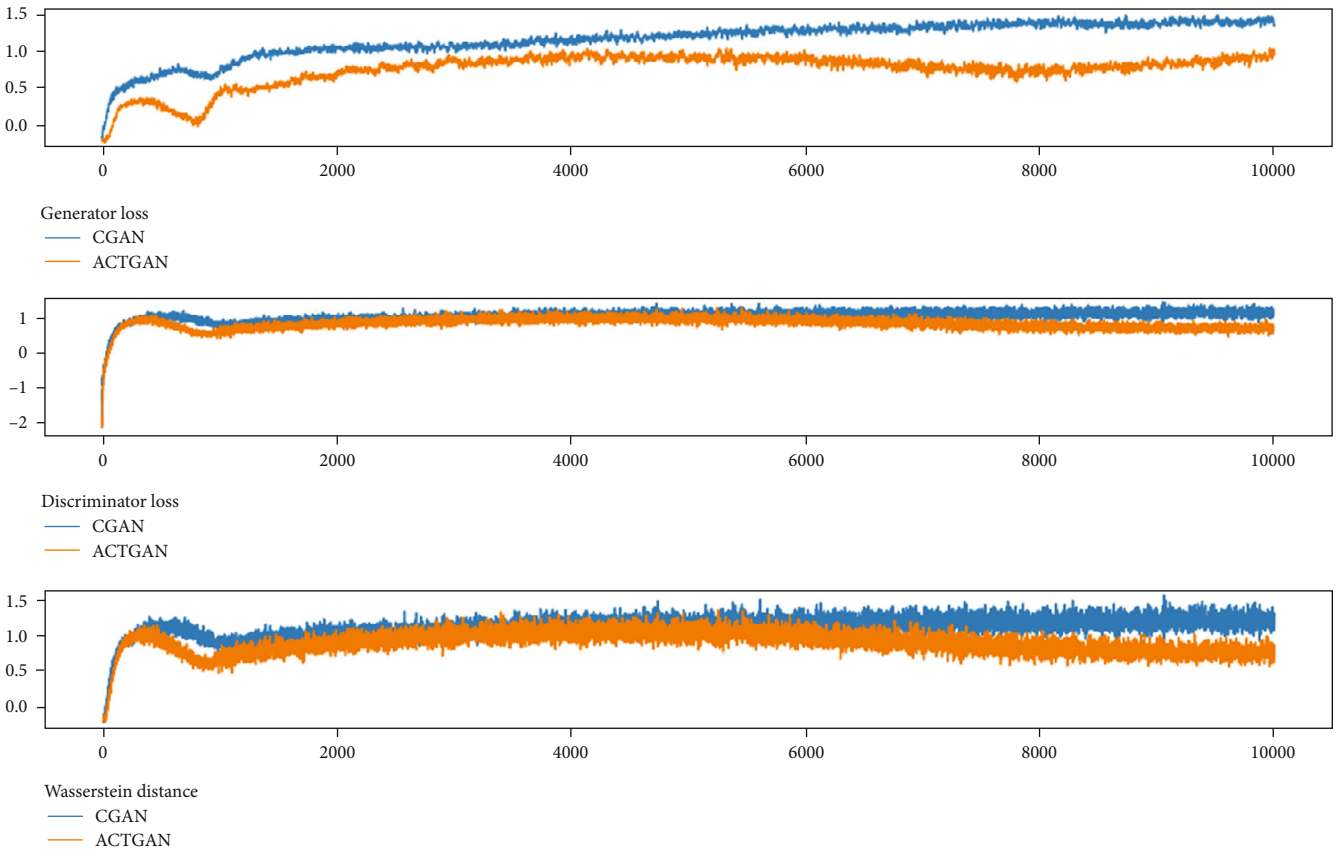


Figure 5: Comparison of training results.

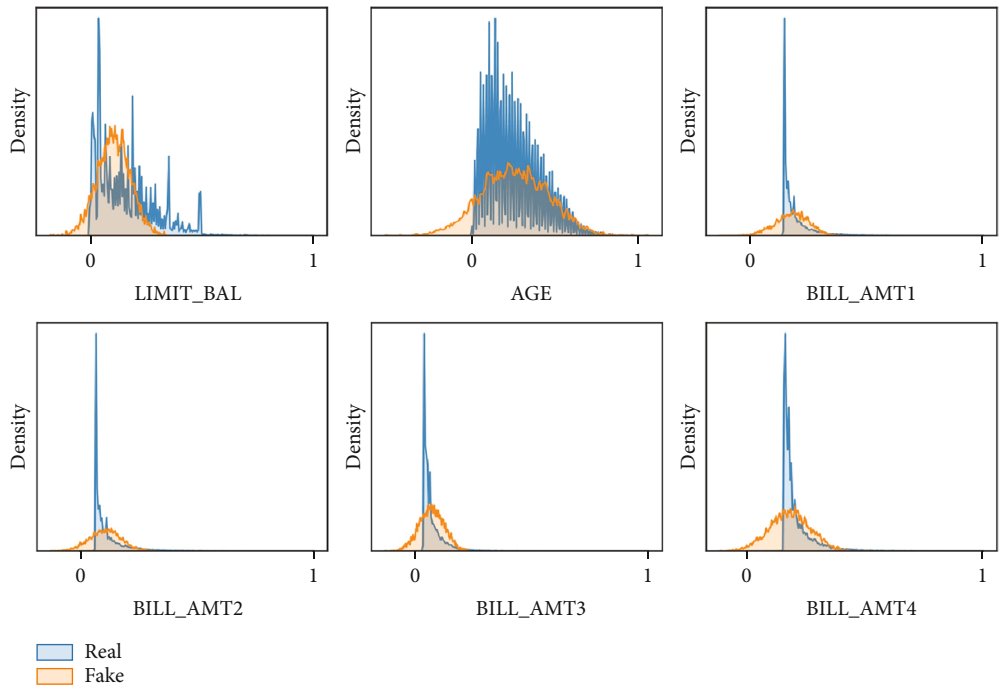
ACTGAN is distributed with a higher similarity to the real data.

Taking the six characteristics of age AGE, LIMIT_BAL, and BILL_AMT1-4 as examples, draw the distribution frequency map of real data and generated data. Since the data is preprocessed before being brought into the training model, the characteristics at this time can only reflect information, and does not represent the real attribute value. As shown in Figure 6, as the number of iterations increases, the data distribution generated by the generator G is getting closer and closer to the real data distribution, showing model's excellent learning ability.

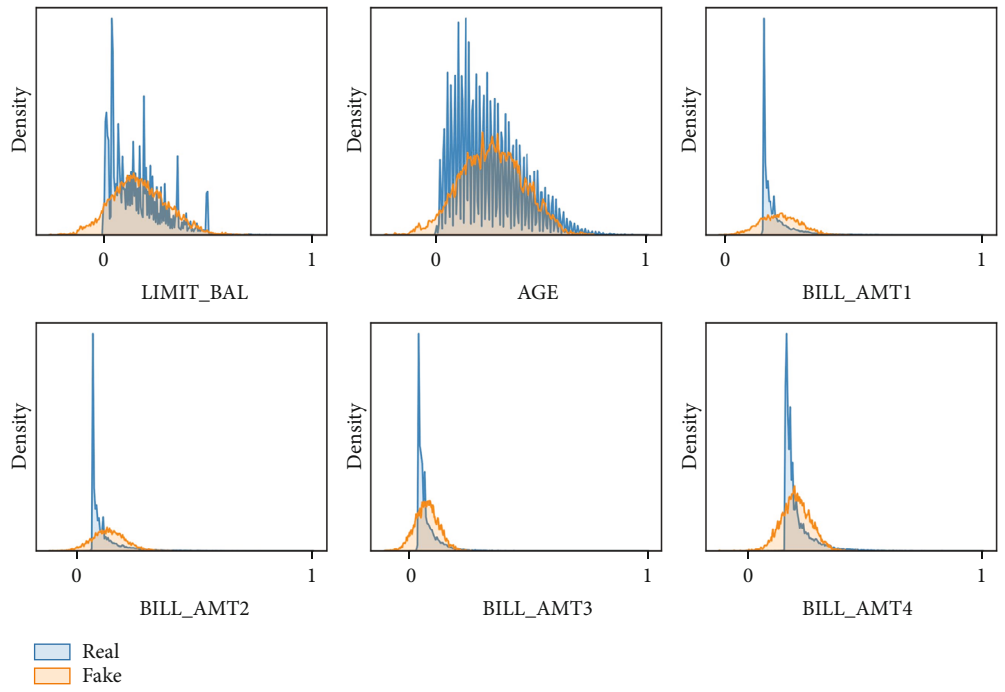
In order to reflect the enhancement effect of the data generation algorithm in this paper and the impact of data enhancement on the recognition performance of the classification algorithm, top data enhancement algorithms such as ADASYN, SMOTE, BorderlineSMOTE, and CGAN are selected to enhance the imbalanced data set in Table 1. Each dataset generates sufficient samples to increase the imbalance ratio to 0.4, and eventually forms an enhanced dataset; four machine learning algorithms: support vector machine, logistic regression, decision tree, and K-nearest neighbors are selected as classifiers, and the classifiers are compared in all performance metrics on the dataset.

In Tables 3 and 4, the bold font in each row represents the highest value of the row, from which it can be seen that the ACTGAN method is significantly better than other methods when using decision tree, support vector machine and k-nearest neighbor classifier for classification. Although the ACTGAN method does not show obvious advantages under some evaluation criteria, it may be related to the structural characteristics of some datasets. On the other hand, due to the training time of the ACTGAN model, the selection of the number of hidden layer nodes in the generator network and the discriminant network in the experiment is not very sufficient, and the number of training times of the model is not enough. The choice of hyperparameters are very dataset-dependent; but overall, the ACTG AN method still significantly outperforms several other resampling methods.

4.3.2. Ablation Study. Since we added an attention layer after the feature extraction layer, we performed ablation experiments on temporal attention and spatial attention. The results are shown in Table 5. It can be seen that the addition of the attention layer model has different degrees of improvement in the four evaluation indicators. In the UCI dataset, the effect of adding a single spatial attention SAM is better than that of adding a single TAM, and in the banking dataset, the effect of a single TAM is better than a single SAM, which may be due to the differences between the datasets and feature importance. Therefore, the combination of temporal and spatial attention can more comprehensively extract data feature information. The experimental results also show that the dual attention mechanism can effectively make up for the performance shortcomings of the single

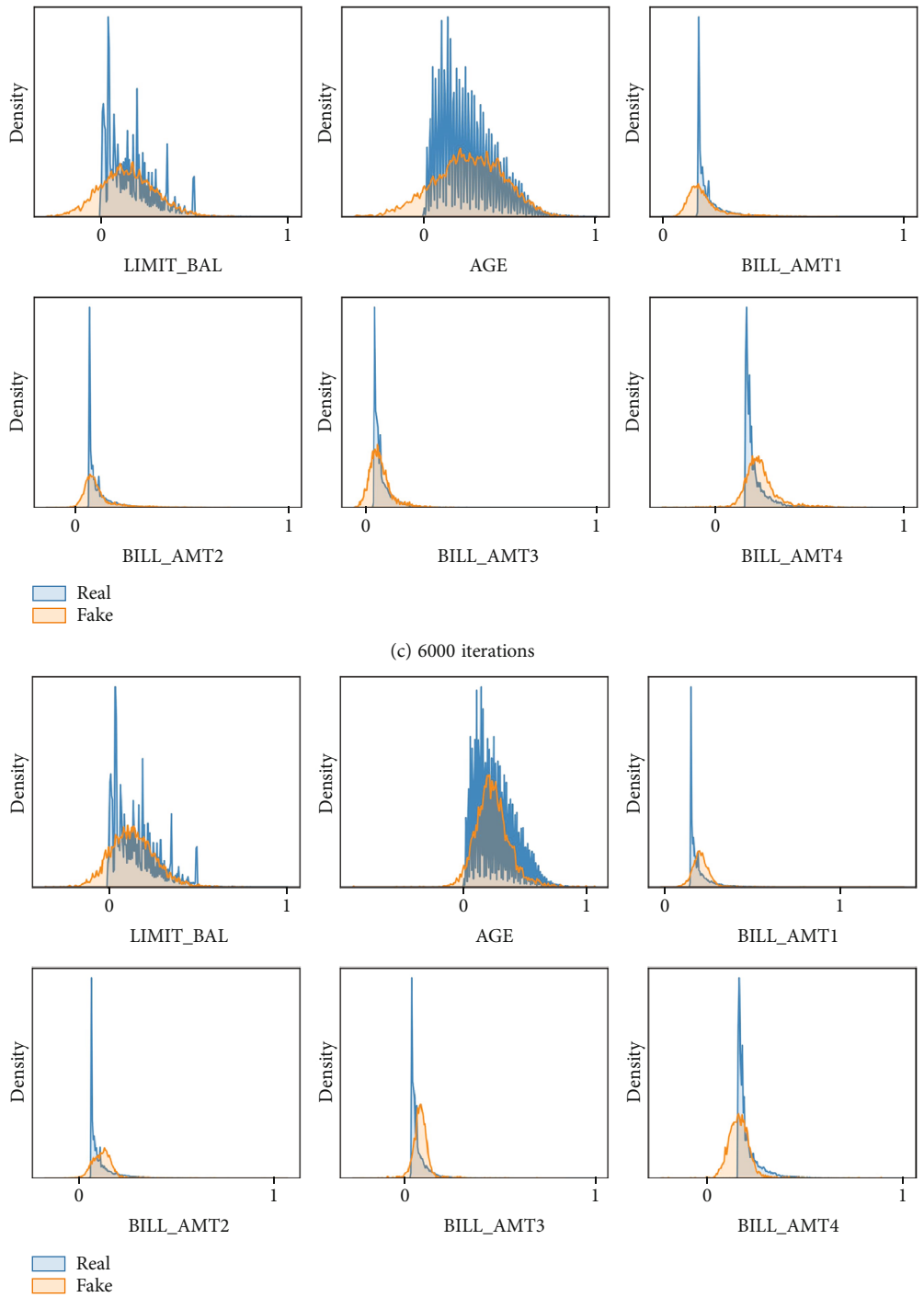


(a) 1000 iterations



(b) 3000 iterations

Figure 6: Continued.



(c) 6000 iterations

(d) 9000 iterations

Figure 6: Data distribution of real data and fake data.

attention layer. The UCI data set applies the spatiotemporal attention module in the F1 value, AUC value is better than other methods. The banking dataset outperforms other methods in recall rate, F1 value, AUC value, and KS value, and the overall performance is the best.

4.3.3. Imbalance Classification Results. To demonstrate the advantages of the ResNet-LSTM+XGBoost algorithm, several baseline methods for imbalance classification are adopted for comparison. All hyperparameters are tuned on validation set with grid search.

RNN-RF [39]: RNN is used for dynamic temporal feature modeling, static features are extracted by feedforward neural network. Then the dynamic features and static features are fused and output to the random forest algorithm to predict the final default probability.

Table 3: Comparison of different data generation algorithm (Bank dataset).

	Critics	None	ADASYN	SMOTE	BorderlineSMOTE	CGAN	ACTGAN
Support vector machine	Recall	97.53	99.18	99.07	**99.25**	99.13	99.02
	F1	77.01	**90.66**	89.66	89.75	89.37	89.27
	AUC	94.52	95.38	95.11	95.21	95.56	**96.02**
	KS	88.41	84.59	82.90	83.25	83.48	**89.01**
Logistic regression	Recall	79.43	97.95	**98.75**	98.68	89.54	92.60
	F1	72.28	**90.31**	89.68	89.79	80.92	81.84
	AUC	86.50	94.91	95.05	**95.13**	90.37	92.40
	KS	70.01	82.98	82.46	82.72	84.65	**85.54**
Decision tree	Recall	62.99	86.59	85.12	86.11	69.24	**89.24**
	F1	63.20	**87.23**	85.17	85.42	71.05	81.05
	AUC	78.73	90.74	89.58	89.93	81.65	**91.34**
	KS	53.53	81.04	77.71	78.93	61.57	**81.69**
K nearest neighbors	Recall	24.70	90.22	88.98	89.48	94.46	**95.76**
	F1	13.45	**88.26**	87.11	87.33	84.95	84.05
	AUC	16.06	92.06	91.40	91.63	**94.42**	94.09
	KS	10.04	80.12	82.78	82.69	86.17	**87.46**

Table 4: Comparison of different data generation algorithm (UCI dataset).

	Critics	None	ADASYN	SMOTE	BorderlineSMOTE	CGAN	ACTGAN
Support vector machine	Recall	22.70	23.54	40.99	42.13	46.07	**47.32**
	F1	34.15	35.77	50.92	51.96	55.72	**56.98**
	AUC	59.92	60.36	66.47	67.04	69.16	**69.79**
	KS	19.70	20.09	32.12	33.37	37.75	**39.16**
Logistic regression	Recall	25.05	27.49	33.84	35.83	39.67	**43.13**
	F1	36.41	39.71	46.10	47.95	52.05	**55.17**
	AUC	60.78	61.83	64.31	65.18	67.27	**68.84**
	KS	21.52	23.67	26.99	28.83	32.57	**36.18**
Decision tree	Recall	41.68	51.73	56.17	55.95	55.63	**56.56**
	F1	39.92	49.75	55.19	54.93	**55.28**	55.19
	AUC	61.50	66.04	68.56	68.38	**68.64**	68.31
	KS	22.05	31.25	36.52	**36.95**	36.60	36.04
K nearest neighbors	Recall	34.08	41.14	46.46	**48.84**	44.65	46.49
	F1	41.79	47.99	53.83	**56.01**	54.64	55.84
	AUC	63.00	65.28	67.97	**69.28**	68.55	69.12
	KS	24.00	29.74	34.12	36.46	36.51	**37.89**

Table 5: Ablation study.

Method	Bank				UCI			
	Recall	F1	AUC	KS	Recall	F1	AUC	KS
None	90.33	81.23	91.45	85.41	55.24	65.52	74.86	52.85
SAM	89.35	80.31	90.84	84.38	57.60	66.11	75.37	52.70
TAM	94.24	82.22	93.07	84.87	54.51	65.24	74.66	53.58
SAM + TAM	**96.20**	**82.52**	**93.82**	**85.54**	57.32	**66.45**	**75.53**	53.49

Table 6: Classification method results of model.

Method	Bank				UCI			
	Recall	F1	AUC	KS	Recall	F1	AUC	KS
RNN-RF	82.07	78.44	87.89	70.64	25.77	35.60	60.24	20.28
Adaboost	**98.73**	78.89	95.25	87.25	31.89	43.55	63.92	27.66
SMOTEBoost	81.33	73.69	87.52	72.39	39.08	46.88	65.69	29.16
CUSBoost	74.05	70.22	84.09	64.50	36.33	46.11	65.21	27.59
RUSBoost	96.68	78.84	94.47	88.91	**61.43**	51.72	70.12	40.18
CNN	80.70	73.38	87.22	88.54	39.39	47.13	65.83	40.11
LSTM	95.09	79.03	93.91	89.46	28.01	38.10	61.36	34.05
Propose	98.41	**82.52**	**96.93**	**89.54**	57.34	**67.91**	**75.40**	**53.29**

Adaptive boosting tree (AdaBoost): iteratively optimize multiple weak classifiers and make them a strong classifier by adjusting the weights of misclassified data during each iteration.

SMOTEBoost [40]: Adaboost combined with random oversampling methods. SMOTE uses k-nearest neighbors to create synthetic examples of the minority class, then injects the SMOTE method on each boost iteration.

RUSBoost [41, 42]: compared to SMOTEBoost, RUSBoost achieves the same goal by performing random undersampling (RUS) at each boost iteration instead of SMOTE.

CUSBoost [43]: the majority class data is divided into K classes using the K-means algorithm (K is determined by hyperparameter optimization). Then within each cluster, randomly selected 50% of the data. Use the selected data together with the minority class data to form new balanced data.

CNN: The CNN-based architecture with a sliding window approach on behavioral data to overcome the class imbalance problem [44]. It has two convolutional layers which have filters of length 8 and 4. The number of the filters is set to be 32 and 64, and set the hidden layer dimension as 32.

LSTM: The LSTM model with hidden layer size of 128 for feature extraction.

The experimental results are shown in Table 6. The results show that the ResNet-LSTM +XGBoost model on the two datasets has obvious advantages over other algorithms in terms of F1, AUC value, and KS value. The recall rate of Adaboost on Bank dataset and the recall rate of RUSBoost on UCI dataset has exceeded our proposed method, this is due to differences in the distribution of minority class samples between datasets. Though the recall rate has not reach the highest score, the value of our proposed method is not far from the highest one. Meanwhile, in the other indicators our method has certain advantage, indicating that the ResNet-LSTM+XGBoost algorithm has excellent feature learning ability, and can extract information embedded in data nodes from different angles. Besides, improving Focal-XGBoost can also improve the classification performance of unbalanced learning.

5. Conclusion and Future Work

In order to solve the problem of data imbalance in the field of credit scoring, a novel data generation method ACTGAN for imbalanced data is proposed. Comparing results with other imbalanced data sample generation algorithms shows that the model training convergence speed and classification effect are improved. On this basis, a fusion deep neural network credit scoring framework based on ResNet-LSTM is proposed. In the framework, ResNet is used to extract static features from static financial data, and LSTM is used to extract dynamic features to detect the time dependence of user behavior data. And the spatio-temporal attention module is added into the framework to assign different weights to each unit when processing tabular data and time series to obtain detailed information of the focused target area. In the end, the Focal Loss function is introduced to improve the XGBoost classifier, thus the supervised learning ability of neural network is improved. We compare the performance of our hybrid deep learning model with other unbalanced classifiers proposed in related fields through experiments on the UCI dataset and private banking dataset, and the performance of our method has been significantly improved on F1 value, AUC, KS value.

For applications, this research provides compelling data and methodological support for more accurate solutions to credit scoring problems in the future, guiding the design of future methods for the same type of problems, proposing faster and more accurate solutions compared to previous solutions, and providing more applicability advantages in a future where data volumes and data dimensions are increasing. For the research object and domain, the study adds new insights into the optimization and application of ResNet-LSTM methods, which can be deepened in multiple contexts in the future, for example, in real-world problems such as geospatial assessment of inter-temporal evolution where data imbalances also exist. This research can be further improved in future study in several ways. First, in order to mine more information in the dimension of feature extraction and make full use of the user data provided by the credit platform, our future work will focus on social network data for entity relationship extraction, and try graph convolution networks to further extract data embedded in data nodes Information. Also, due to the small amount of dataset used in this study, we plan to incorporate more imbalance datasets to verify the reliability of our credit scoring model.

Data Availability

The data is available by sending email to corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Research and Development Project of the Ministry of Housing and Urban-Rural Development under Grant No. 2021-K-148, the Educational Commission of Zhejiang Province of China (No. Y202147553), and Zhejiang Provincial Natural Science Foundation of China (No. LGF20C050001).

References

[1] K. Bastani, E. Asgari, and H. Namavari, "Wide and deep learning for peer-to-peer lending," *Expert Systems with Applications*, vol. 134, pp. 209–224, 2019.

[2] M. O. Zan, G. A. I. Yanrong, and F. A. N. Guanlong, "Credit card fraud classification based on GAN-Ada boost-DT imbalanced classification algorithm," *Journal of Computer Applications*, vol. 39, no. 2, p. 618, 2019.

[3] Z. Z. SamanehSorournejad, R. E. Atani, and A. H. Monadjemi, *A survey of credit card fraud detection techniques: Data and Technique Oriented Perspective*, 2016.

[4] L. L. Song, S. H. Wang, C. Yang, and X. Sheng, "Application research of improved XGBoost in imbalanced data processing," *Computer Science*, vol. 47, no. 6, pp. 98–103, 2020.

[5] V. E. Neagoe, A. D. Ciotec, and G. S. Cucu, "Deep convolutional neural networks versus multilayer perceptron for financial prediction," in *2018 International Conference on Communications (COMM)*, pp. 201–206, Bucharest, Romania, 2018.

[6] L. Yu, R. Zhou, L. Tang, and R. Chen, "A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data," *Applied Soft Computing*, vol. 69, pp. 192–202, 2018.

[7] B. Zhu, W. Yang, H. Wang, and Y. Yuan, "A hybrid deep learning model for consumer credit scoring," in *2018 international conference on artificial intelligence and big data (ICAIBD)*, pp. 205–208, Chengdu, China, 2018.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[9] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with Applications*, vol. 91, pp. 464–471, 2018.

[10] 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*, vol. 479, pp. 448–455, 2019.

[11] A. Blanco, R. Pino-Mejías, J. Lara, and S. Rayo, "Credit scoring models for the microfinance industry using neural networks: Evidence from Peru," *Expert Systems with Applications*, vol. 40, no. 1, pp. 356–364, 2013.

[12] N. Metawa, I. V. Pustokhina, D. A. Pustokhin, K. Shankar, and M. Elhoseny, "Computational intelligence-based financial crisis prediction model using feature subset selection with optimal deep belief network," *Big Data*, vol. 9, no. 2, pp. 100–115, 2021.

[13] S. Deng, R. Li, Y. Jin, and H. He, "Cnn-based feature cross and classifier for loan default prediction," in *2020 International Conference on image, video processing and artificial intelligence*, International Society for Optics and Photonics, 2020.

[14] C. Yan, X. Fu, W. Wu, S. Lu, and J. Wu, "Neural network based relation extraction of enterprises in credit risk management," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–6, Kyoto, Japan, 2019.

[15] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1569–1572, Virtual Event China, 2020.

[16] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, and Y. Qi, "Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism," *AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 946–953, 2019.

[17] Q. Zhong, Y. Liu, X. Ao et al., "Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network," in *Proceedings of The Web Conference 2020*, pp. 785–795, Taipei, Taiwan, 2020.

[18] M. A. Al-Shabi, "Credit card fraud detection using autoencoder model in unbalanced datasets," *Journal of Advances in Mathematics and Computer Science*, vol. 33, no. 5, pp. 1–16, 2019.

[19] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[20] A. Iranmehr, H. Masnadi-Shirazi, and N. Vasconcelos, "Cost-sensitive support vector machines," *Neurocomputing*, vol. 343, pp. 50–64, 2019.

[21] G. Paleologo, A. Elisseeff, and G. Antonini, "Subagging for credit scoring models," *European Journal of Operational Research*, vol. 201, no. 2, pp. 490–499, 2010.

[22] C. Luo, "A comparison analysis for credit scoring using bagging ensembles," *Expert Systems*, vol. 39, no. 2, article e12297, 2022.

[23] J. H. Wang and J. R. Yan, "Unbalanced data classification algorithm based on under-sampling and cost-sensitive," *Computer Applications*, vol. 41, no. 1, pp. 48–52, 2021.

[24] C. F. Tsai, Y. F. Hsu, and D. C. Yen, "A comparative study of classifier ensembles for bankruptcy prediction," *Applied Soft Computing*, vol. 24, pp. 977–984, 2014.

[25] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, https://arxiv.org/abs/1411.1784.

[26] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, pp. 1–7, New York, NY, USA, 2017.

[27] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*, pp. 2642–2651, Sydney, NSW Australia, 2017.

[28] M. Zheng, T. Li, R. Zhu et al., "Conditional Wasserstein generative adversarial network-gradient penalty-based approach to

alleviating imbalanced data classification," *Information Sciences*, vol. 512, pp. 1009–1023, 2020.

[29] Y. Liu, Y. Zhou, X. Liu, F. Dong, C. Wang, and Z. Wang, "Wasserstein GAN-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology," *Engineering*, vol. 5, no. 1, pp. 156–163, 2019.

[30] I. Haloui, J. S. Gupta, and V. Feuillard, "Anomaly detection with Wasserstein GAN," 2018, https://arxiv.org/abs/1812.02463.

[31] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-Learning-Enhanced multitarget detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.

[32] P. R. Vardhani, Y. I. Priyadarshini, and Y. Narasimhulu, "CNN data mining algorithm for detecting credit card fraud," in *Soft Computing and Medical Bioinformatics*, pp. 85–93, Springer, Singapore, 2019.

[33] Y. Zhang, D. Wang, Y. Chen, H. Shang, and Q. Tian, "Credit risk assessment based on long short-term memory model," in *International Conference on Intelligent Computing*, pp. 700–712, Springer, Cham, 2017.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV USA, 2016.

[35] W. Zheng, L. Yan, C. Gou, and F. Wang, "Federated meta-learning for fraudulent credit card detection," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 4654–4660, Yokohama, Japan, 2021.

[36] C. Wang, D. Han, Q. Liu, and S. Luo, "A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism LSTM," *IEEE Access*, vol. 7, pp. 2161–2168, 2018.

[37] T. Liang, G. Zeng, Q. Zhong et al., "Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets," in *14th ACM international conference on web search and data mining*, pp. 229–237, 2021.

[38] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, Cham, 2018.

[39] T. C. Hsu, S. T. Liou, Y. P. Wang, and Y. S. Huang, "Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1572–1576, Brighton, UK, 2019.

[40] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 107–119, Springer, Berlin, Heidelberg, 2003.

[41] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection for digital Twin in smart manufacturing With industrial Cyber-Physical Systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.

[42] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, p. 185, 2009.

[43] F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid, "Cusboost: cluster-based under-sampling with boosting for imbalanced classification," in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pp. 1–5, Bengaluru, India, 2017.

[44] H. Kvamme, N. Sellereite, K. Aas, and S. Sjursen, "Predicting mortgage default using convolutional neural networks," *Expert Systems with Applications*, vol. 102, pp. 207–217, 2018.