IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 4, 2020, accepted August 4, 2020, date of publication August 10, 2020, date of current version August 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015600

Using Variational Auto Encoding in Credit Card Fraud Detection

HUANG TINGFEI, CHENG GUANGQUAN, AND HUANG KUIHUA
System Engineering College, National University of Defense Technology, Changsha 410073, China
Corresponding author: Cheng Guangquan (cgq299@nudt.edu.cn)

ABSTRACT Machine learning approaches are widely used to analyze and detect the increasingly serious problem of credit card fraud. However, typical credit card datasets present imbalanced classification situations because of severely skewed class distributions. Although researchers have proposed some strategies to deal with these imbalances, disadvantages remain. We propose an oversampling method based on variational automatic coding (VAE), combined with classic deep learning techniques, to solve this problem. The VAE method is used to generate a large amount of diverse cases from minority groups in an imbalanced dataset, which are then used to train the classification network. The proposed method is tested on an open credit card fraud dataset, which contains transactions conducted by European cardholders over two days in September 2013. Experimental results show that the VAE method performs better than synthetic minority oversampling techniques and traditional deep neural network methods. In addition, it outperforms recent oversampling methods based on generative adversarial network (GAN) models. After submitting the extended dataset to the baseline for training, the test of the VAE model performs well on indicators including precision, F-measure, accuracy and specificity. These experimental results suggest that the VAE-based oversampling method can be effectively applied to imbalanced classification problems.

INDEX TERMS Credit card fraud, variational automatic coding, oversampling, generative adversarial network, deep learning.

I. INTRODUCTION

In recent decades, especially with the development of e-commerce, more and more people use credit cards for payment. Although credit card payments facilitate all kinds of business activities, credit card fraud is a significant problem.

Credit card fraud not only brings huge economic losses to financial institutions and banks, but also trouble and stress to the lives of individuals who are affected. Recent statistics show that, in 2018 the global economic loss caused by credit card fraud was 27.85 billion dollars, an increase of 16.2% compared with 23.97 billion dollars in 2017. If this trend continues, by 2023 the economic losses caused by credit card fraud will exceed 35 billion dollars [1].

Effective fraud monitoring and prevention can reduce the economic loss of credit card fraud to issuing and managing online trading institutions. In addition, effective fraud detection applications can increase customer confidence and reduce customer complaints. Most credit card fraud detection approaches make use of machine learning [2]. At present, machine learning has many mature methods to solve the problem of credit card fraud [3], [4], including supervised learning [5], [6], semi-supervised learning [2], [7], and unsupervised learning [7]. However, despite much research [7]–[11], a perfect and efficient solution is still needed [12].

Thanks to the rapid development of hardware technology and big data technology, the most widely used method for fraud detection is supervised learning methods. However, in credit fraud situations, the number of positive (fraudulent) cases is much smaller than the number of negative cases. This creates a problem of imbalanced classification, where one class is very much smaller than the other class [13]. After model training, the positive cases will be interpreted as noise and will be discarded, resulting in the deviation of classification results towards the negative class. Therefore, researchers have suggested ways to improve fraud classification results by reducing the class imbalance in the training data. One approach is to increase the number/proportion of positive cases (oversampling) and the other is to reduce the number/proportion of negative cases (undersampling) [14]–[16]. The undersampling method involves choosing

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

a scheme in which a negative sample and all positive cases are mixed. Although it can achieve certain results, it will lose some important hidden information from missing negative cases, which could have an unfavorable impact on the classification results. In general the oversampling method is preferred by researchers.

This paper describes an oversampling method to generate a large number of reliable positive cases. After the additional cases are mixed into the original training set the imbalance between negative and positive cases is reduced. This class balance can improve the general performance of the classifier, and ultimately improve the accuracy of the fraud detection scheme. There are strict requirements for the oversampling model, which needs to fully analyze the original positive cases and find the deep connection and hidden information between the positive cases.

The method proposed in this paper is inspired by a generative model of artificial intelligence, which is primarily used to generate image data and has achieved exciting results in that field. We propose using the deep learning method of variational encoding (VAE) to generate positive data. Coincidentally, Fiore *et al.* [17] recently studied the application of generative adversarial networks (GAN) in an oversampling method, and obtained good sensitivity performance in the imbalanced classification. Nonetheless, there are significant shortcomings to the GAN method. First, it is difficult for the discriminator and generator to reach the theoretical Nash equilibrium when the GAN is trained; that is, the model struggles to converge. Second, the lack of diversity among the generated cases will lead to duplicate cases. Finally, the most important concern is that the GAN is not suitable for processing discrete data; due to the loss function and the characteristics of text data, it cannot generate text data well.

In order to solve the problem of data set imbalance in credit card fraud detection based on supervised classification, a new training set is constructed, in which the number of cases of ''minority class'' is more than that of the original training set. This training set is a combination of virtual cases generated by VAE and cases of ''minority class'' in the original training set. First of all, we use VAE to generate a series of reliable virtual cases, which are trained by minority class cases extracted from the original training set. The trained VAE can imitate the example of the primitive minority class cases. From the view of the nature of VAE, the virtual case generated has the characteristics of diversity. After that, the synthetic case is combined with the original training data to obtain a more effective enhanced training set. Then the desired effectiveness can be achieved by using a traditional classifier. Experimental evaluation shows that the performance of the classifier trained on the extension set is much better than that trained on the original data, especially in terms of precision and F1-measure, which is a very effective solution for credit card fraud detection. Although our framework is introduced in the context of credit card fraud detection, it should be noted that the framework is quite general and can be easily extended to other application domains.

The structure of this paper is as follows. In Section 2, we systematically introduce the relevant work of credit fraud. In Section 3, we present some basic theoretical knowledge about the model. In Section 4, the model architecture used in the experiment is elaborated in detail. In Section 5, we discuss related experimental content, such as data preprocessing, comparison and analysis of experimental results. Finally, we close by summarizing our goals and findings.

II. RELATED WORK

Credit card fraud has long been the subject of research. Over the years, researchers have used a variety of methods to perform fraud detection analyses, including clustering technology [18], peer group analysis [19], genetic algorithms [20], association rules [21], Bayes analyses [22], and neural networks [10].

However, the situation has changed. The prosperity of the e-commerce industry makes it easy for credit card issuers to collect transaction-level information for credit card fraud. In academia, machine learning has become the main method to study credit card fraud, because it can benefit from these massive datasets of complete transaction information [23]–[25]. Specifically, supervised machine learning methods [5], [6] are considered the most promising to solve such problems. In supervised learning, two strategies are commonly used to improve the performance of the model. One is the use of improved classifiers; the other is to improve data preprocessing [26].

The first strategy has already yielded valuable results. Some researchers have shown that random forest models outperform machine learning algorithms such as support vector machines, logistic regression, and K-nearest neighbors [9]. At the same time, other researchers have focused on the application of neural networks in credit card fraud detection. For example, Zandian *et al.* [10] used neural networks to detect massive credit card fraud transactions. Maes *et al.* [27] used a three-layer neural network feed-forward function for fraud detection. In addition, some researchers have explored the differences between neural networks and traditional machine learning in solving credit card fraud. For example, Akbani *et al.* [28] compared neural network and Bayesian network method for credit card fraud detection. Their experimental results show that Bayesian networks are superior to neural networks in credit card fraud detection.

In the second strategy, some researchers have adopted the method of undersampling [29]. This approach reduces the number of cases for most classes in the training set, so as to reduce the differences between class sizes (the numbers of cases per class) in the training set. However, although this approach can improve some indicators, it also loses important information which makes the trained classifier defective [30]. Instead, oversampling [31] has become the current research focus for improving data pre-processing. Early approaches simply copied case information of minority classes Although this was effective, it also limited the generalization performance of the model, encouraging researchers to explore new

FIGURE 1. **(a):The pre-processed data flows into the hidden layer of the neural network, then the features of the input data are abstracted at multiple levels in the hidden layer,so that the output result supports the clearest classification by the classifier. (b):The new sample x' generated by the SMOTE method is any point on the line connecting the minority samples a and b, where sample a is randomly determined, and sample b is the nearest neighbor sample of sample a.**

methods of oversampling. Al Majzoub *et al.* [32] proposed a method of synthesizing minority oversampling (SMOTE). After sampling the minorities, a new sample is synthesized by randomly selecting cases and generating interpolation. Repeated operations of this type yield a large number of new cases from the minority sample. Variants of the SMOTE method include Farthest SMOTE [33], HCAB-SMOTE [34], etc.

Recently, Yee *et al.* [4] used GAN to oversample credit card fraud and showed that, overall, it is better than the traditional SMOTE method. While acknowledging their success, we have chosen not to focus only on the GAN method, because its optimization is very challenging. On the contrary, we adopt VAE as the oversampling module. After relevant optimization, the output samples from the model are very diverse. We also use a different benchmark model. There are not many Sigmoid and Relu functions in the model. In the experiment, only three fully connected layers are used and two Relu functions are added, a configuration that yields strong results.

III. PRELIMINARIES

Because the oversampling method performs well in imbalanced classification problems, it is widely used especially in the problem of credit card fraud. In this paper, we examine three different oversampling models used to detect credit card fraud: SMOTE, GAN, and VAE. In addition, the baseline used is not a classical machine learning method, but a deep learning method.

A. SMOTE

The core of the SMOTE algorithm [32] is to analyze the samples of minority groups, synthesize sample data according to certain rules, and then add artificially synthesized cases of minority groups to the training set. In the process of generating data, the algorithm uses K nearest neighbor (KNN) technology. First, for each case x of the minority sample (the white diamond shown in Figure 1(b)), calculate the Euclidean distance between every two cases in the minority group to obtain the k nearest neighbor of the target case. Second, randomly select a sample a, and then find n cases from the nearest neighbors of the selected cases. Then a sample b is selected from the n cases, and the new sample x' is any point on the line between the two samples a and b. This is expressed by the formula:

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

B. BASELINE, GAN AND VAE

Deep learning [35] is a new research direction of machine learning. Its structure is very flexible and can be adjusted to a variety of needs. After the structure is determined, the network parameters can be adjusted continuously to make its data output close to the label samples. In terms of mathematics, there is very little difference between deep learning and traditional machine learning methods, both of which analyze the characteristics of the data in a high-dimensional space, and then process the data efficiently.

$$out_j^l = \sigma(\sum_k w_k^l out_k^{l-1} + b_j^l) \tag{2}$$

In Formula (2), out_j^l represents the output of the j neuron in the network l layer, w_k^l represents the weight of j neuron in the network l layer, and b^l represents the offset of j neuron in the network l layer. The specific operation process of the neural network is represented in Figure 1(a). After each dataset is input into the network, the weight of each neuron in the current layer is multiplied by the data and the neuron's offset is added, and then processed by the activation function to yield the output from the current layer of the neural network. The output data is then used as the input data for the next layer

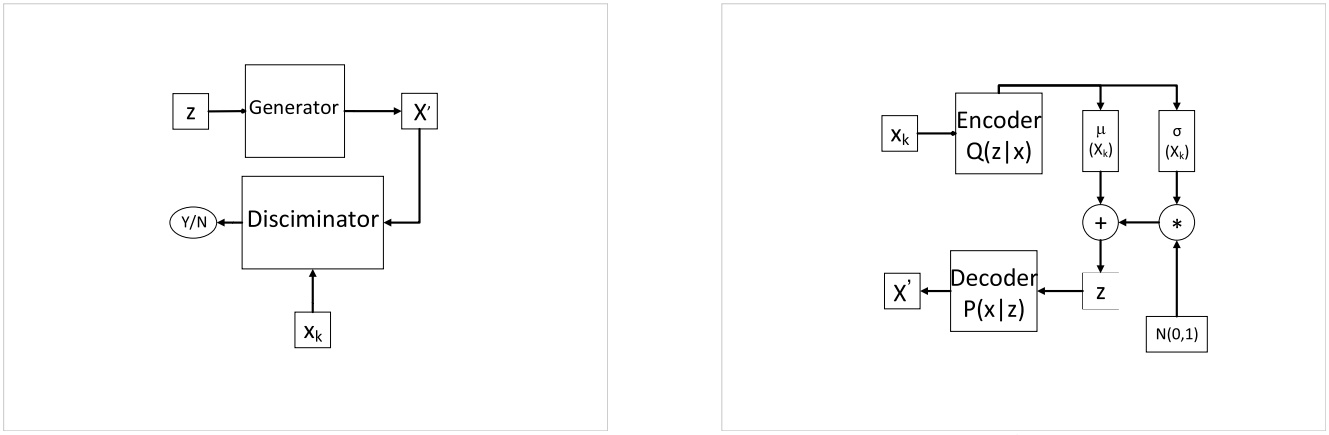


FIGURE 2. (a):The input of generator G is random noise z, and the generated output is a virtual sample that is transmitted to discriminator D. The role of discriminator D is to distinguish the sample generated by generator G from the real sample. At the same time, both sides receive negative feedback to continuously optimize themselves during the process. (b):The model first uses a neural network to fit the distribution of real samples to align with the standard normal distribution $N(0, 1)$, and then it samples from the standard normal distribution. The generator uses the sampled data to generate samples to ensure the diversity of the final output samples.

of the neural network again, and so on, until the target result is output.

GAN [36] is also essentially a type of deep learning model, and from the perspective of development, it offers the most promising direction for deep learning progress. GAN has two main modules. In practice, each module is a deep learning model, that is, a neural network. The two models are used as a generator (G) and a discriminator (D), respectively. The generator network generates simulated data, and the discriminator judges the difference between the simulated data and the target data, yielding a true or false judgment about the virtual data. In the end, the model can output higher-quality simulation data to complete the data generation task.

The process of generating data by GAN is similar to the process of Figure 2(a). Figure 2(a) shows an iteration of GAN activity. At the beginning of the process, the generator initially generates poor data that is passed to the discriminator for the first discrimination activity. The discriminator then feeds back the results of its discrimination to the generator and at the same time optimizes its own parameters. The generator can use the discriminator results to self-optimize its parameters, and generate improved virtual data. At this point, one iteration process ends. GAN will continue to iterate on this cycle until it reaches equilibrium, That is, the parameters of the generator and discriminator cannot be obviously optimized.

Consider the process using picture data as an example. The distribution $p(x)$ of the real picture data x is known. The next thing to do is to generate a virtual image data of x', which has the same distribution. Typically, researchers use an original random noise z to generate virtual data that conforms to a specific distribution; by using an original dataset with a specific distribution to generate a virtual dataset close to the target data. Kingma *et al.* [36] used a neural network as a generator. After inputting the original data, the virtual data was output based on the fitting results.

$$V(G, D) = E_{x\sim}[logD(x)] + E_{x\sim P_{x'}}[log(1 - D(x))] \quad (3)$$

Among them, P_x is the real data distribution, and $P_{x'}$ is the distribution of the generated virtual data. The advantage of formula (3) is that, after fixing G, $maxV(G, D)$, means the difference between P_x and $P_{x'}$, then one only needs to find the best G to minimize the difference between them.

VAE is obtained by adding variation on the basis of the autoencoder. It is very similar to GAN. Again, the function is to transform and fit the data distribution to generate virtual data close to the target. The working process of the model is shown in Figure 2(b).

Researchers assume that, given a batch of raw data $x_1, x_2, x_3, \ldots, x_n$, there is a specific sample x_k and k is in $[1, n]$, for which there is an exclusive distribution p_k that is independent, multivariate, and normal [6]. With this assumption, researchers can easily use the decoder to restore the data. As we all know, the normal distribution is fully described by two important parameters, mean and variance. Generally, the number of samples is the same as that of normal distribution. If all samples are obtained, it can be very challenging work. Therefore, researchers often use neural networks to fit the mean and variance of normal distribution.

VAE has another advantage, as shown in Figure 2(b). The raw data is input to the encoder, and the average and variance of the raw data are obtained. Then, according to the mean and variance output from the encoder, generate a random number that obeys the corresponding Gaussian distribution. However, in this case the obtained data distribution cannot be directly decoded. If the researchers were to decode the data distribution directly, the effect would be the same as autoencoder [38]. Therefore, the researchers made the distribution p_k close to the standard normal distribution. Further,

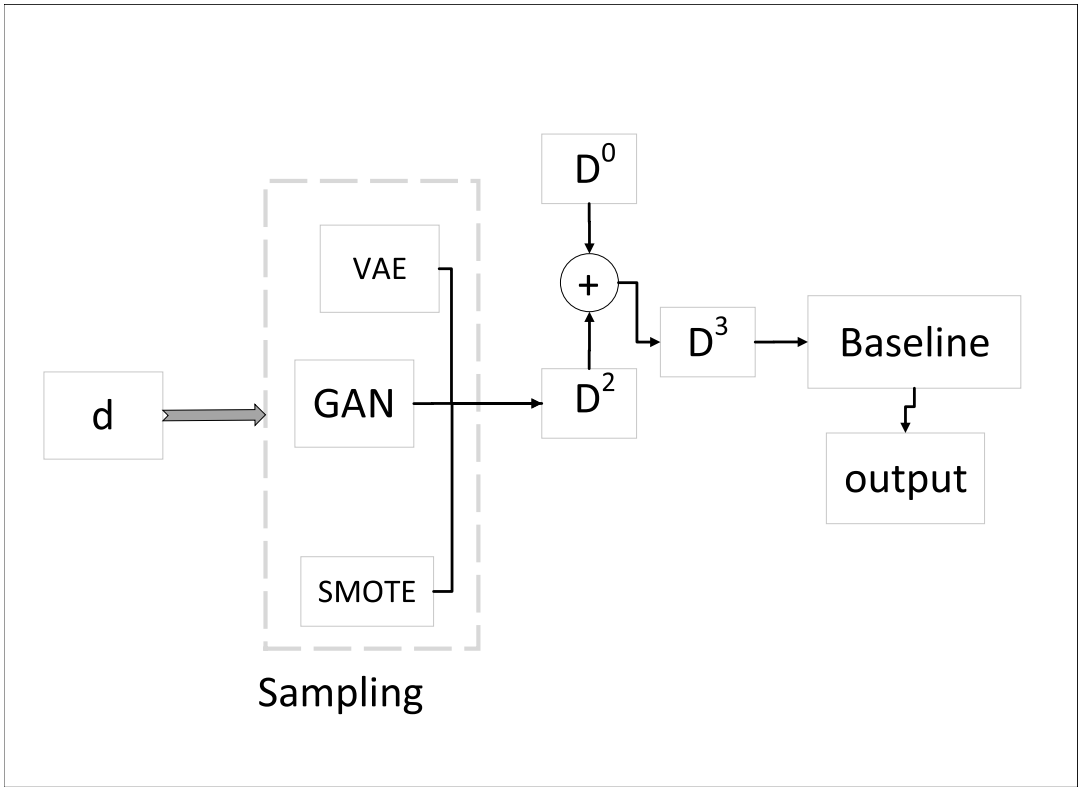


FIGURE 3. The sample d of the minority group in the training set is input into the sampling module. The input data is used as the training set d of the sampling module. The output is a set of positive samples D^2 generated by the model and the virtual fraud data is merged with the original training data D^0 into a new training set D^3. This training set is input to the benchmark model to train the classifier. Finally, the classifier outputs the classification results.

according to Equation 4,

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

Among them, $\sigma(x)$ is the real data standard deviation, and $\mu(x)$ is the mean of the real data. Only then can the encoder's distribution be decoded directly, and finally generate data. We also designed a special loss function [37] to keep all samples close to the standard normal distribution.

IV. OUR APPROACH

Credit card fraud detection problems can be expressed as common binary classification problems. Different from the common classification problem, however, the distribution of cases into the two categories (fraudulent, not fraudulent) is seriously imblanced. Among them, fraudulent transactions are only a small part of non-fraudulent, accounting for less than 1% of the total data [13]. To improve the effectiveness of classification results, a framework for detecting credit card fraud will usually try to eliminate the gap between the two categories of cases in the data set. The framework adopted in this paper is to inject positive data obtained by oversampling into the original training set to obtain a hybrid training set. The gap between the two types of samples in the mixed training set is reduced, and then the classifier is trained using the mixed training set.

A. FRAMEWORK USED IN THE EXPERIMENT

In the experiment, the baseline method uses a deep learning network. The module of the classifier consists of three layers of network, with two relu functions sandwiched in between the neural networks. During training, the divided training set D^0 passes through each layer network and relu function in turn, yielding the classification label as final output. In the training process, the supervised learning optimizes the model parameters according to the feedback results of the tag, and maintains the model parameters after the model training. When the test set data is input, the model will output the classification results. Finally, the classification results are tested against related indicators.

The second model used the SMOTE module to replace the VAE. The model corresponds to Figure 3, but substituting the SMOTE module for the VAE module (method 1). Before training, minority class samples (fraud samples d) in the training set are input into the SMOTE module, which synthesizes virtual fraud data based on the input training set data d. This virtual fraud data is then merged with the original training set D^0 into a new training set D^3. This training set is input into the reference model to train the classifier, and the next stage of the training process continues as for the baseline method.

Recently, the GAN method has been used for oversampling. Its workflow is similar to that of SMOTE. Again, the model follows Figure 3 except that the VAE module is changed to the GAN module (method 2). Before training, it is also necessary to input minority samples d from the training set into the GAN module. The input data is used as the training set d of the GAN, and the parameters of the GAN are adjusted. The GAN receives input random noise to generate the virtual fraud data. During training, the model achieves Nash equilibrium as described in Section 3. However, it is difficult for GAN to achieve the theoretical Nash equilibrium in actual training, so this paper determines whether the equilibrium is reached by observing the change in the error output by the model. When the loss function values of the discriminator and generator are not changed obviously, the model is considered to have reached an actual balance. After reaching equilibrium, the optimized model parameters are saved. The optimal parameters of GAN are adjusted to generate virtual fraud data. The virtual fraud data and the original training set D^2 are combined into a new training set D^3. This training set is input into the reference model to train the classifier, and the next stage of the training process continues as for the baseline method.

In section 1, we analyzed the limitations of the GAN method. Compared with the GAN method, first, the VAE method can generate data with more potential variables, that is, the generated data has better diversity, and second, it can directly generate data by means of encoding and decoding, which is more convenient to operate. Finally, the generation of text data has very few restrictions on VAE, which makes VAE excellent in text generation. These advantages make VAE achieve better performance than GAN in this experiment.

Figure 3 shows the model for the proposed method using the VAE. Before training, it is also necessary to input minority samples d into the VAE module. After data is input as the VAE training set d, the VAE parameters are adjusted and the VAE module receives random noise input to generate virtual fraud data. After the model training, the optimal parameters of VAE are adjusted to generate virtual fraud data. The virtual fraud data and the original training set D^0 are combined into a new training set D^3. This training set is input into the reference model to train the classifier, and the next steps of the training process continue as for the baseline method.

B. METRICS

Researchers have proposed many measurement metrics to detect the performance of imbalanced classification models. These metrics include recall rate [17], [38], specificity [17], precision [17], [38], F-measure [17], [38], and accuracy. In this paper, we focus on two indicators: precision and F-measure. In credit card fraud, these are considered the most important indicators. Precision is the ratio of the number of real positive cases and the number of predicted positive cases. In the detection of credit card fraud, the first goal is to provide the maximum truth, that is, the highest precision. In practice, a false alarm can lead to a poor customer experience, and potentially lead to the loss of customers, so the precision of the model is very important, and is considered the most important indicator of the fraud system. On the other hand, although precision can be increased by reducing the model's recall rate, it is impossible to improve the precision by reducing the recall rate without limitation. Recall rate is the ratio of the number of predicted positive cases to the number of all real positive cases. Real fraud problems have real negative economic implications to the enterprise, so the recall rate is also worthy of our attention. Another indicator is the F-measure, which takes values on the range [0,1]. The F-measure captures both the precision and recall rate, measuring improvements in both indicators simultaneously. In machine learning, this index is often used to evaluate the advantages and disadvantages of various algorithms, because it can evaluate the precision and recall rate in combination. Accuracy is a commonly used indicator, which represents the proportion between the number of correctly classified cases and the total number of cases. If the accuracy of a model is too low, it cannot be applied in practice. Specificity is the ratio of the number of predicted negative cases predicted to be negative to the number of real negative cases. In the experiment, we consider these five indicators to measure the performance of the model.

V. EXPERIMENT

Credit card data is private to enterprises and customers, which makes it difficult to obtain a dataset of credit card fraud data. Therefore, our experiment was tested on an open dataset. The dataset can be downloaded at the URL `https://www.kaggle.com/mlg-ulb/creditcardfraud`. The dataset contains information about credit card transactions conducted by cardholders in Europe over two days in September 2013. There are 284807 transactions in total, including 492 positive cases (fraudulent transactions), accounting for 0.172% of the data, which is typical of an imbalanced classification problem. It contains only numerical input variables which are the result of a PCA (principal components analysis) transformation. Unfortunately, due to confidentiality issues, we cannot get the original features and more background information about the data. The information we can obtain is shown in Table 5. Features V1, V2, ...V28 may be result of a PCA Dimensionality reduction to protect user identities and sensitive features the only features which have not been transformed with PCA are 'Time' and 'Amount'. 'Time' represents number of seconds elapsed between this transaction and the first transaction in the dataset. 'Amount' represents the transaction amount. In addition, in the 'Class', 1 represents a fraudulent transaction, otherwise 0.

The eigenvalue of 'Amount' is standardized and normalized to a new eigenvalue. The so-called standardization and normalization means that the following formula (5) is applied so that the variance of the new feature value is 1 and the mean value is 0. The correlation of 30 eigenvalues of the original

dataset is determined by pearson correlation coefficient. The results show that the two eigenvalues of time and amount are related to the other 28 eigenvalues processed by PCA. Therefore, the two eigenvalues are deleted.

$$X = (x - \mu)/\sigma \tag{5}$$

Among them, X is a new eigenvalue, and x is the eigenvalue of 'Amount'. μ is the mean of this feature of 'Amount', σ is the standard deviation of this feature of 'Amount'.

In general, researchers will delete duplicate data; however, for the imbalanced classification problem, to ensure data integrity and all important information, duplicate data was not deleted. The dataset includes 9144 duplicate cases in the data, of which 473 are positive cases (fraudulent transactions) and 8671 are negative cases.

Finally, the dataset was divided into a training set (about two-thirds) and a test set (about one-third). Therefore, there were 349 positive cases in the training set (about 0.00183%) and 143 positive cases in the test set (0.00152%).

A. BASELINE

Because the deep learning method was selected as the baseline, the hyperparameters of the deep neural network had to be tested to select the best performing model. The hyperparameters of a neural network have a great impact on the final performance of the model. The hyperparameters that most influence deep learning are the number of layers of the neural network and the initial values of neural network weights. Too few layers of neural network will hinder the extraction of data features, and too many layers will lead to model over-fitting. To select the baseline with the best performance, two layers of neural networks, three layers of neural networks, and four layers of neural networks were tested, respectively, and the classification results from the classifier were analyzed and compared. From this testing, three layers of neural networks were chosen for use. The weight initialization of the neural network has a strong effect on the speed of convergence and model performance. Effective choices of weights can solve gradient problem associated with deepening the layers of the neural network. After the number of neural network layers was determined, random seeds were selected on the integer interval of [0, 100] to initialize the weights of the neural network. It was found that most of the random seeds had a positive effect on the experimental results. After synthesizing the five indexes, one of the better ones was selected as the experimental parameter. Finally, the random number 8 was selected as the weight initialization parameter of neural network. In addition, in the experiment, the optimal learning rate, the number of iterations, the loss function and other metrics were measured step by step. The parameters of the benchmark model are shown in Table 1.

B. SMOTE APPROACH

In the SMOTE method, the parameters of the baseline module remained the same as for the baseline method (Section 5.1).

TABLE 1. Baseline model parameters.

DNN parameters	value
Learning rate	0.00003
Optimizer	Adam
Loss function	crossEntropyLoss()
Network initialization seed	8
Batchsize	64
Iteration	70
Activation function	Relu

In the SMOTE, when the virtual positive cases are synthesized, the random seed must be fixed to reproduce the results. This parameter has little impact on the classification results of the baseline model, and any value can be specified.

The specific parameters of this method are shown in Table 2. Similarly, the specific architecture of the model is shown in Figure 3 of the Section 4. In this part of the experiment, the effect of oversampling using the SMOTE model was tested in detail. Although there were 349 positive cases in the original training set, any number of virtual positive cases can be generated in the experiment. In the end, the proportion of cases generated using the SMOTE to the total number of positive cases was 0.25, 0.5, 1, 2, 3, 4, 8, 10, 20, 100, respectively. After the cases were generated, they were combined with the original training set for training.

TABLE 2. SMOTE model parameters.

DNN parameters	value
Learning rate	0.00003
Optimizer	Adam
Loss function	crossEntropyLoss()
Network/SMOTE initialization seed	8/42
Batchsize	64
Iteration	70
Activation function	Relu

In the training process, there are a total of 70 iterations. In each iteration, the test set is input in batches. The classification results are stored every 100 steps, and the classification results are stored 30 times per iteration. We use the average value of a single iteration of five indicators. After the training, the results of [40, 60] iterations are selected for analysis and comparison.

The test results of SMOTE are shown in the box plot of Figure 4. The horizontal axis in the figure represents the different training sets used in training. The vertical axis represents the proportion of the injected virtual data relative to the positive data in the training data D^0. The ordinate shows the numerical size of the relevant metrics. The value range of the metrics is [0, 1], the larger the value, the better. It can be found that with the increase of the virtual data injected into the training set, the sensitivity first increases rapidly and then goes up smoothly. It starts to increase rapidly

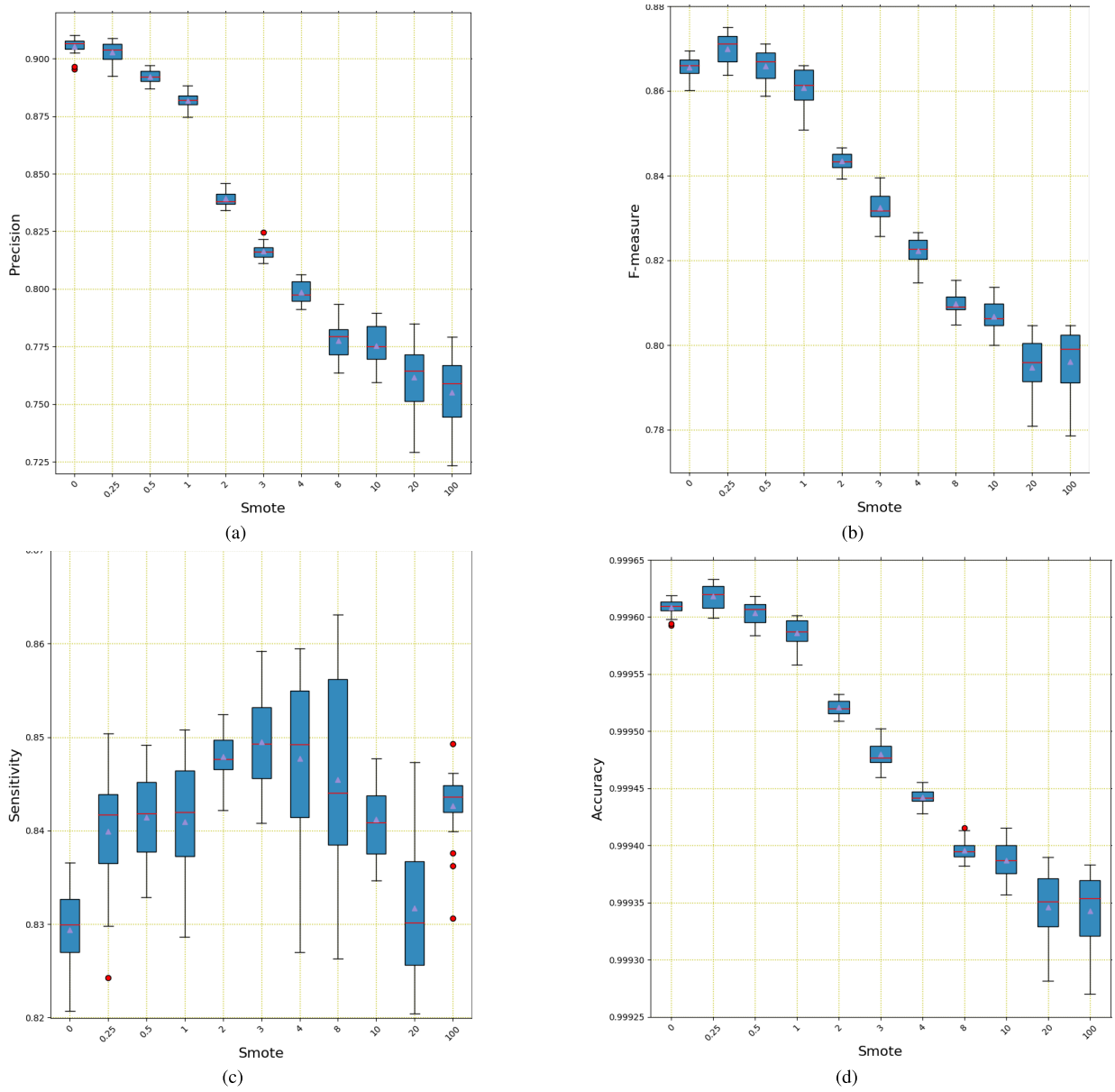


FIGURE 4. After the original training set is injected with different numbers of positive sample data generated by SMOTE, the baseline effect changes. The data is the average value of 20 iterations after the test results converge. (a)Precision. (b) F-measure. (c) Sensitivity. (d) Accuracy.

again when the data is 2 times and then goes up smoothly. It starts to keep decline greatly when the data is 8 to 20 times, and has a huge rebound when the data is 100 times. Among other metrics, the precision first drops slowly, then starts to maintain plummets hugely when the data is 2 time. It starts to maintain a slow decline speed when the data is 10 times until the end of the test. F measure and accuracy had a slightly increase at the beginning and then showed the same decline as precision. The possible reason may be that when the virtual positive data is limited, a small amount of noise will produce a benign deviation to the reference model classification.

C. GAN APPROACH

Table 3 reports the parameters of the GAN. In the experiment, the model hyperparameters were measured and determined.

TABLE 3. GAN model parameters.

GAN parameters	value
Learning rate	0.0001
Optimizer	Adam
Loss function	$log(1 - D(G(z)))$
Network initialization seed	1
Batchsize	1
Iteration	600
Activation function	Relu & Sigmoid

The network of the generative model and the discriminant model in the GAN is two layers. The GAN is an improvement from the original model, which is basically the same as the

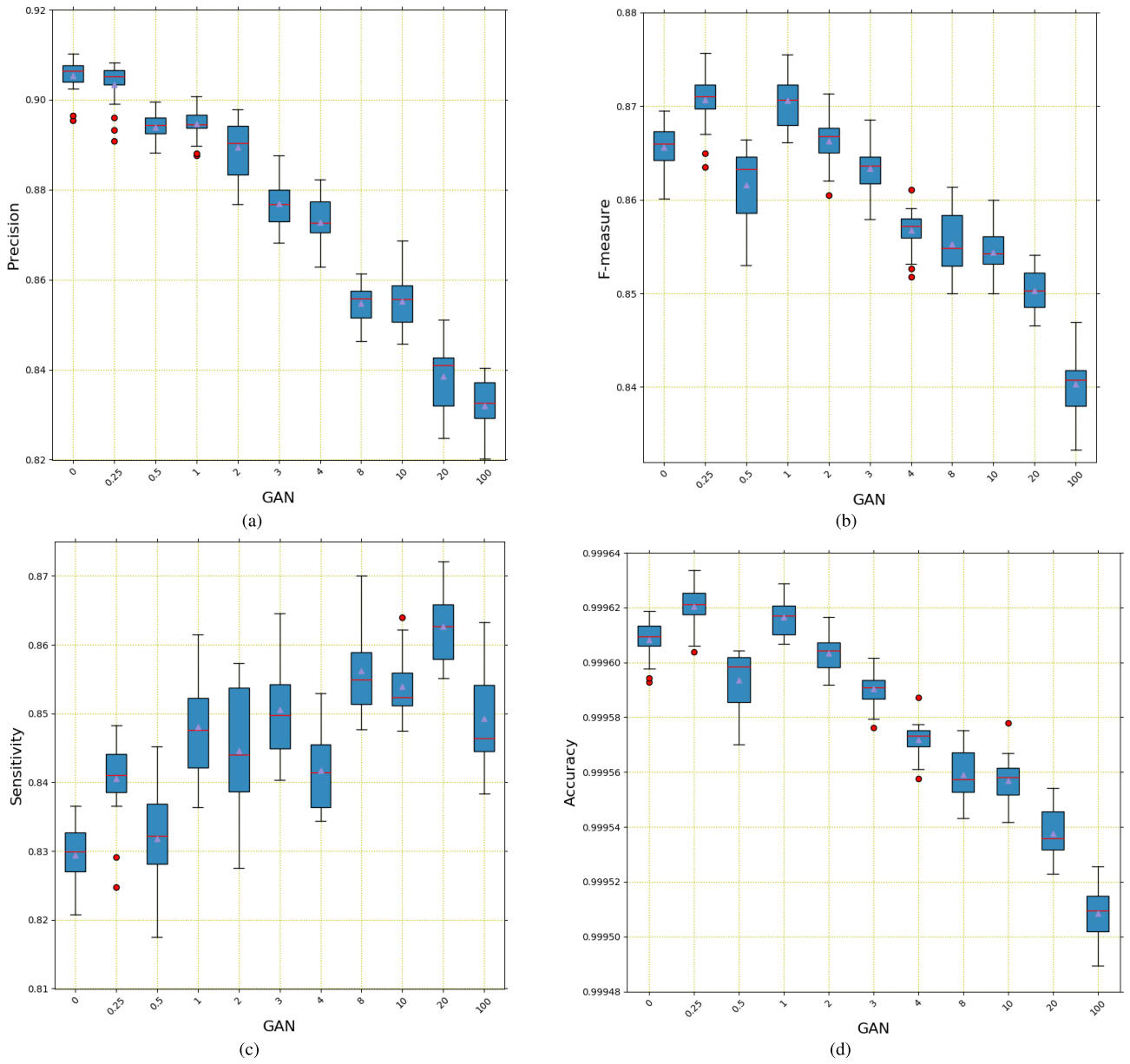


FIGURE 5. After the original training set is injected with different numbers of positive sample data generated by GAN, the baseline effect changes. The data is the average value of 20 iterations after the test results converge: (a)Precision. (b) F-measure. (c) Sensitivity. (d) Accuracy.

model adopted by Fiore *et al.* [17]. The specific architecture of the model is shown in Figure 3.

Figure 5 shows the experimental results based on the oversampling of GAN. It can be found that with the increase of the virtual data injected into the training set, the sensitivity first increases hugely and then drops, then increases again hugely and drops. Sensitivity maintain a wave-like change and present an overall rising trend. Among other metrics, the precision first drops slowly, then drops hugely. Similarly, precision maintain a wave-like change and present an overall decline trend until the end of the test. F measure and accuracy had a greatly increase at the beginning, and then drop hugely,and then increases greatly again. After that, they kept drop until the end of the test.

D. VAE APPROACH

Unlike GAN, the VAE model is not only easy to converge, but can also generate very diverse cases. The method of VAE combines the generating model with the useful algorithm elements in deep learning technology. There are three difficulties in the application of VAE in credit card fraud detection, which differ from its application to image generation. First, because the final output of the model is not image data, using a convolution network as the neural network module of VAE can't achieve better performance. Second, the positive data in the training set is very scarce and is not suitable for batch training. Finally, the number of positive training cases is small, which leads to big fluctuations in the value of the loss function in the model output, and which is used to judge

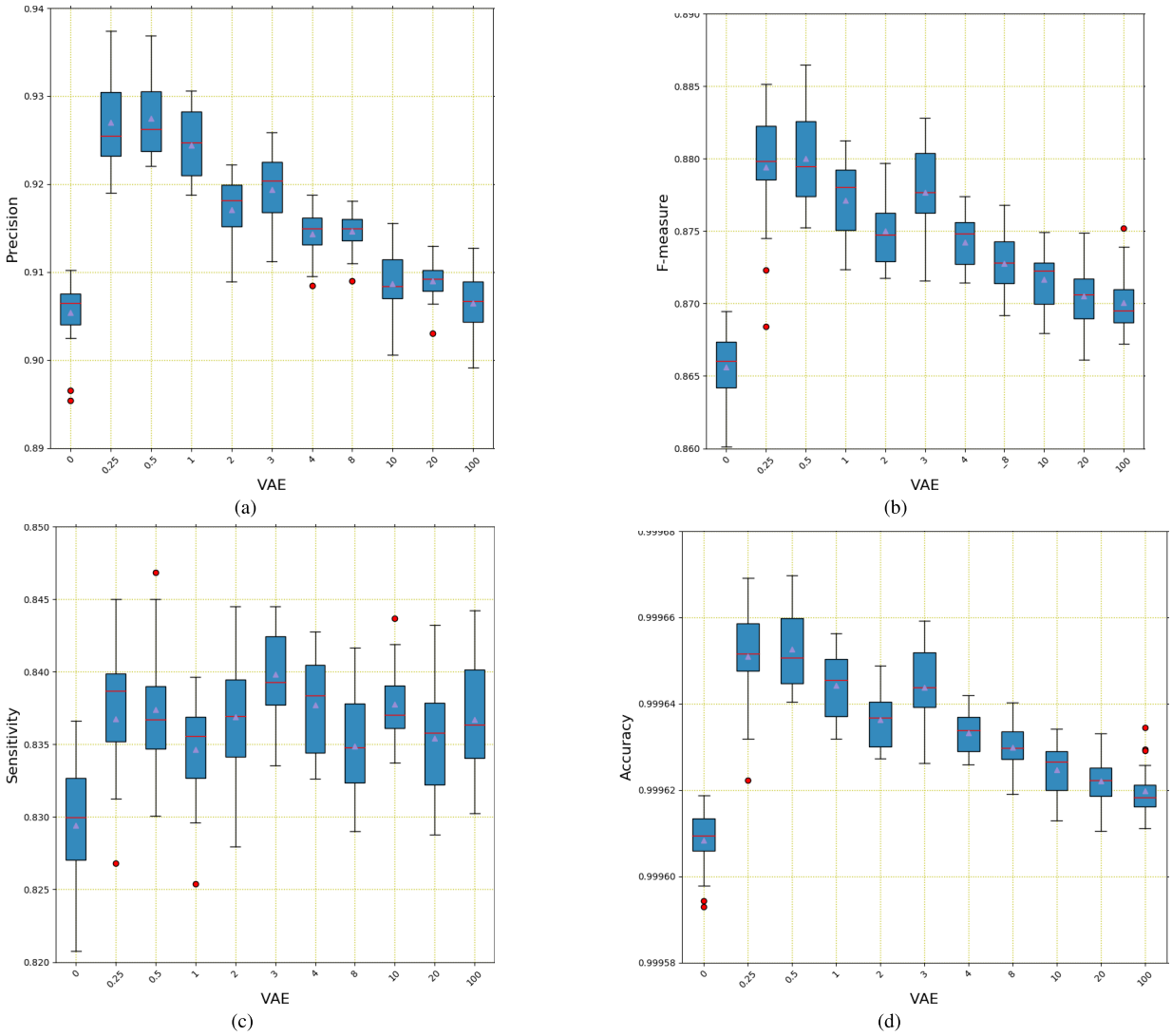


FIGURE 6. After the original training set is injected with different numbers of positive sample data generated by VAE, the baseline effect changes. The data is the average value of 20 iterations after the test results converge. (a)Precision. (b) F-measure. (c) Sensitivity. (d) Accuracy.

the convergence of the model. We address these difficulties as follows: First, we use the full connection network as the neural network module of VAE. Second, in the experiment, instead of batch processing, we use the input data one by one for training. Third, our approach uses the average value of the loss function of each iteration of the model to judge the convergence of the learning algorithm.

Our VAE model parameters are shown in Table 4. In the experiment, the decoder and encoder in VAE were a two-layer neural network. After extracting the positive cases from the original training set, we input the data one by one for training. At the same time, we observed the average value of the loss function of the model to determine the convergence result. The specific architecture of the VAE is shown in Figure 2(b).

The experimental results of the proposed method are shown in Figure 6. Compared with the benchmark model,

TABLE 4. VAE model parameters.

VAE parameters	value
Learning rate	0.000005
Optimizer	Adam
Loss function	BCE + KLD
Network initialization seed	1
Batchsize	1
Iteration	41
Activation function	Relu & Sigmoid

the performance of the VAE model in precision, F-measure, recall rate and accuracy rate are all suitable for fraud detection. The overall results show a wave-like fluctuation, first a sharp increase, then a downward trend, then a short rise, and finally a slight decline. However, even when the experimental

TABLE 5. Data Sources.

	V1	...	V28	Time	Amount	Class
1	-1.3598071336738	...	-0.0210530534538215	0	149.62	0
2	1.19185711131486	...	0.0147241691924927	0	2.69	0
3	-1.35835406159823	...	-0.0597518405929204	1	378.66	0
4	-0.966271711572087	...	0.0614576285006353	1	123.5	0
5	-1.15823309349523	...	0.215153147499206	2	69.99	0
6	-0.425965884412454	...	0.0810802569229443	2	3.67	0
7	1.22965763450793	...	0.00516776890624916	4	4.99	0
8	-0.644269442348146	...	-1.08533918832377	7	40.8	0
9	-0.89428608220282	...	0.14240432992147	7	93.2	0
...	...	...	...	...	...	...

TABLE 6. Precision and F-measure as the number N_g of generated examples is varied.

N_g	Precision			F-measure		
	SMOTE	GAN	VAE	SMOTE	GAN	VAE
0	0.90537	0.90537	0.90537	0.86559	0.86559	0.86559
87	0.90269	0.90340	0.92704	0.86997	0.87070	0.87943
175	0.89208	0.89389	0.92750	0.86558	0.86157	0.88001
349	0.88179	0.89460	0.92444	0.86070	0.87061	0.87713
698	0.83933	0.88943	0.91761	0.84344	0.86626	0.87490
1047	0.81640	0.87697	0.91858	0.83245	0.86336	0.87697
1396	0.79861	0.87275	0.91709	0.82222	0.85677	0.87500
2792	0.77754	0.85473	0.91464	0.80980	0.85530	0.87278
3490	0.77548	0.85524	0.90866	0.80684	0.85440	0.87165
6980	0.76162	0.83863	0.90897	0.79480	0.85029	0.87050
34900	0.75517	0.83206	0.90644	0.79613	0.84029	0.87004

results of the model fell to the lowest point, the value was still higher than the benchmark model. Coincidentally, when the number of virtual positive samples generated by VAE was about 175, the performance of the classifier reached the optimal values for four of the indicators.

E. RESULTS COMPARISON AND ANALYSIS

Due to space limitations, the baseline model, SMOTE model, GAN model and VAE model box line map of different positive samples of each index is omitted here. The average values of the specific results are shown in Table 6 and Table 7.

First, from Table 6 and Table 7, it can be seen that the use of oversampling methods to differentially increase the number of positive cases does have different degrees of impact on the classifier. The recall rate of the SMOTE and GAN methods has increased by 0.02 and 0.03 respectively, reaching 0.85 and 0.86, which is a significant improvement compared to the baseline of 0.83. However, the improvement of these two methods on the other four indicators is not ideal, and there is basically no improvement compared with the baseline. For example, the optimal performance of the GAN model in the F measure is 0.87070, which is only 0.005 higher than the baseline, while the optimal performance of the SMOTE model in the F measure is 0.86997, which is only 0.004 higher than the baseline.

Second, the performance of the VAE model in the five indicators is better than that of the benchmark model, and its performance for precision, F-measure, specificity and accuracy is the best among the four models. The performance of precision and F-measure are especially significant. The optimal values of these two aspects are not only larger than the baseline, but also improved by about 3% than the optimal

TABLE 7. Sensitivity, Specificity and Accuracy as the number N_g of generated examples is varied.

N_g	Sensitivity			Specificity			Accuracy		
	SMOTE	GAN	VAE	SMOTE	GAN	VAE	SMOTE	GAN	VAE
0	0.82943	0.82943	0.82943	0.99987	0.99987	0.99987	0.99961	0.99961	0.99961
87	0.83993	0.84054	0.83676	0.99986	0.99986	0.99990	0.99962	0.99962	0.99965
175	0.84145	0.83177	0.83739	0.99984	0.99985	0.99990	0.99960	0.99959	0.99965
349	0.84093	0.84811	0.83466	0.99983	0.99985	0.99990	0.99959	0.99962	0.99964
698	0.84790	0.84460	0.83626	0.99975	0.99984	0.99989	0.99952	0.99960	0.99964
1047	0.84948	0.85052	0.83922	0.99971	0.99982	0.99989	0.99948	0.99959	0.99964
1396	0.84769	0.84171	0.83689	0.99967	0.99981	0.99988	0.99944	0.99957	0.99964
2792	0.84548	0.85623	0.83485	0.99963	0.99978	0.99988	0.99940	0.99956	0.99963
3490	0.84119	0.85394	0.83777	0.99963	0.99978	0.99987	0.99939	0.99956	0.99962
6980	0.83175	0.86266	0.83540	0.99960	0.99975	0.99987	0.99935	0.99954	0.99962
34900	0.84268	0.84923	0.83669	0.99958	0.99974	0.99987	0.99934	0.99951	0.99962

values of the other two methods. In the experiment, when the number of positive cases injected into the baseline by the VAE model reached 175, the performance of the classifier in all five indicators was significantly improved, and the best results were achieved at the same time. For example, the accuracy of the algorithm at this time reached 0.93, which is higher than the baseline value of 0.02, and also higher than the optimal values of SMOTE and GAN of 0.9.

Finally, when the number of injected positive cases exceeds the optimal number, the overall performance of the three models on five indicators worsens. As shown in Table 6 and 7, when the number of injected positive cases reached 100 times the number of positive cases in the original training set, none of the three models performed as well as the optimal values in the experiment. At this time, in addition to the recall rate indicators, the performance of the SMOTE and GAN methods is worse than the baseline method. One possible reason may be that the diversity of the positive cases is different. After all, among the three models, with the increase of the number of generated cases, the VAE has the best diversity of generated cases, followed by the GAN and SMOTE methods.

VI. CONCLUSION

This paper proposed a new method for detecting credit card fraud, namely an oversampling method based on the VAE. From the training set, cases of minority groups were extracted for training, and finally a large number of cases of minority groups are generated. From the experimental results, we can see that, as expected, the comprehensive samples generated by the model are not all positive data, so the injection of comprehensive samples in the training set will causes an increase in false negatives. Obviously, in an environment with relatively high false negative costs, this may be a limiting factor.

The contribution of this paper is the documentation and testing of a new supervised oversampling method, which is desirable when the application data is characterized by a significant imbalance in class sizes.

Although our method has achieved encouraging results, there are limitations. First, this method cannot be applied to the unsupervised environment. Second, although our model is not inferior to the baseline model in terms of recall index performance, it compares less well with the excellent recall performance of the SMOTE and GAN methods. Finally, because the proposed method belongs to the category of supervised learning methods, the model could perform poorly when it deals with completely novel fraud data.

In the open dataset, we tested the method proposed in this paper, and found that the baseline model had the best performance in the five indicators and the best threshold point of the model performance when the positive sample data generated by VAE reaches 0.5 times the number of positive cases in the original training set. At the same time, compared with the other two models, the recall index value showed the smallest difference. In the future, the focus of our research will include improving the recall rate of the model while increasing the precision and F-measure, to achieve a recall performance comparable with that of the SMOTE and GAN methods.

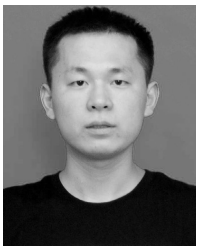
REFERENCES

[1] D. Robertson. The Nilson Report. Kirchhain, Germany. Accessed: Nov. 18, 2019. [Online]. Available: https://nilsonreport.com/mention/407/1link/

[2] A. Salazar, G. Safont, and L. Vergara, "Semi-supervised learning for imbalanced classification of credit card transaction," in *Proc. IJCNN*, Rio de Janeiro, Brazil, 2018, pp. 1–7.

[3] J. Gao, Z. Zhou, J. Ai, B. Xia, and S. Coggeshall, "Predicting credit card transaction fraud using machine learning algorithms," *J. Intell. Learn. Syst. Appl.*, vol. 11, no. 3, pp. 33–63, 2019, doi: 10.4236/jilsa.2019.113003.

[4] O. S. Yee, S. Sagadevan, and N. H. A. H. Malim, "Credit card fraud detection using machine learning as data mining technique," *J. Telecommun., Electron. Comput. Eng.*, vol. 10, nos. 1–4, pp. 23–27, 2018.
[5] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Charlottesville, VA, USA, Apr. 2018, pp. 129–134.
[6] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Zhuhai, China, Mar. 2018, pp. 1–6.
[7] F. Carcillo, Y.-A. Le Borgne, O. Caelen, and G. Bontempi, "Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization," *Int. J. Data Sci. Anal.*, vol. 5, no. 4, pp. 285–300, Jun. 2018, doi: 10.1007/s41060-018-0116-z.
[8] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Inf. Sci.*, to be published, doi: 10.1016/j.ins.2019.05.042.
[9] A. Eshghi and M. Kargari, "Introducing a method for combining supervised and semi-supervised methods in fraud detection," in *Proc. IIIEC*, Jan. 2019, pp. 23–30.
[10] Z. K. Zandian and M. R. Keyvanpour, "MEFUASN: A helpful method to extract features using analyzing social network for fraud detection," *J. AI Data Mining*, vol. 7, no. 2, pp. 213–224, 2019.
[11] L. Tran, N. Roy, and L. Tran, "Solve fraud detection problem by using graph based learning methods," Tech. Rep., 2019.
[12] J. P. Morgan. *Payments Fraud and Control Survey*. Kirchhain, Germany. Accessed: 2016. [Online]. Available: https://www.afponline.org/publications-data-tools/reports/survey-research-economic-data/Index/
[13] F. Provost and T. Fawcett, "Robust classification for imprecise environments," *Mach. Learn.*, vol. 42, no. 3, pp. 203–231, 2001.
[14] T. Hasanin, T. M. Khoshgoftaar, J. L. Leevy, and N. Seliya, "Examining characteristics of predictive models with imbalanced big data," *J. Big Data*, vol. 6, no. 1, p. 69, Dec. 2019.
[15] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, "Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 447–456.
[16] S. Tyagi and S. Mittal, "Sampling approaches for imbalanced data classification problem in machine learning," in *Proc. ICRIC*, Cham, Switzerland, 2020, pp. 209–221.
[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," *Inf. Sci.*, vol. 479, pp. 448–455, Apr. 2019.
[18] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Stat. Sci.*, vol. 17, no. 3, pp. 448–455, 2002.
[19] D. J. Weston, D. J. Hand, N. M. Adams, C. Whitrow, and P. Juszczak, "Plastic card fraud detection using peer group analysis," *Adv. Data Anal. Classification*, vol. 2, no. 1, pp. 45–62, Apr. 2008.
[20] E. Duman and M. H. Ozcelik, "Detecting credit card fraud by genetic algorithm and scatter search," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13057–13063, Sep. 2011.
[21] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, "Association rules applied to credit card fraud detection," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3630–3640, Mar. 2009.
[22] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk," in *Proc. 12th Int. Conf. Mach. Learn. Appl.*, vol. 1, Dec. 2013, pp. 333–338.
[23] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, Feb. 2011.
[24] L. Delamaire, H. Abdou, and J. Pointon, "Credit card fraud and detection techniques: A review," *Banks Bank Syst.*, vol. 4, no. 2, pp. 57–68, 2009.
[25] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013.
[26] Ghosh and Reilly, "Credit card fraud detection with a neural-network," in *Proc. 27th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 1994, pp. 621–630.
[27] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using Bayesian and neural networks," in *Proc. 1st Int. Naiso Congr. Neuro fuzzy Technol.*, 2002, pp. 261–270.
[28] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany, 2004, pp. 39–50.
[29] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 159–166.
[30] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[32] H. Al Majzoub, I. Elgedawy, O. Akaydin, and M. K. Ulukök, "HCAB-SMOTE: A hybrid clustered affinitive borderline SMOTE approach for imbalanced data binary classification," *Arabian J. Sci. Eng.*, vol. 45, pp. 3205–3222, Jan. 2020.
[33] A. S. G. Sardana, "Farthest SMOTE: A modified SMOTE approach," in *Computational Intelligence in Data Mining*. 2019, pp. 309–320.
[34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[36] D. P Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[37] A. Ng, "Sparse autoencoder," *CS294A Lecture Notes*, vol. 72, no. 2011, pp. 1–19, 2011.
[38] V. Van Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "Generative adversarial netsAfraid: Fraud detection via active inference in time-evolving social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2015, pp. 659–666.



HUANG TINGFEI was born in Guangwu, Jieshou, Anhui, China, in 1992. He received the B.Eng. degree in electronic packaging technology from Xidian University, in 2018. He is currently pursuing the master's degree with the College of Systems Engineering, National University of Defense Technology. His research interests include operations research and planning and reinforcement learning.

CHENG GUANGQUAN received the master's and Ph.D. degrees in management science and engineering from the National University of Defense Technology, in 2005 and 2010, respectively. He is currently an Associate Professor with the College of Systems Engineering, National University of Defense Technology. His current research interests include complex network analysis and decision-making support technology.

HUANG KUIHUA received the bachelor's, master's, and Ph.D. degrees from the National University of Defense Technology, in 1999, 2002, and 2011, respectively. He is currently an Associate Researcher with the College of Systems Engineering, National University of Defense Technology. His research interests include intelligent situational awareness and intelligent task planning.