IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 20, 2020, accepted April 24, 2020, date of publication May 14, 2020, date of current version May 28, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2994327*

# Dual Autoencoders Generative Adversarial Network for Imbalanced Classification Problem

**ENSEN WU[1,2], (Member, IEEE), HONGYAN CUI[1,2], (Senior Member, IEEE), AND ROY E. WELSCH[3]**

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]Sloan School of Business, Massachusetts Institute of Technology, Cambridge, MA 02142, USA

Corresponding author: Hongyan Cui (cuihy@bupt.edu.cn)

This work was supported by the China Ministry of Education - CMCC Research Fund under Grant MCM20170306.

**ABSTRACT** The imbalanced classification problem has become greatest issue in many fields, especially in fraud detection. In fraud detection, the transaction datasets available for training are extremely imbalanced, with fraudulent transaction logs considerably less represented. Meanwhile, the feature information of the fraud samples with better classification capabilities cannot be mined directly by feature learning methods due to too few fraud samples. These significantly reduce the effectiveness of fraud detection models. In this paper, we proposed a Dual Autoencoders Generative Adversarial Network, which can balance the majority and minority classes and learn feature representations of normal and fraudulent transactions to improve the accuracy of the fraud detection. The new model firstly trains a Generative Adversarial Networks to output sufficient mimicked fraudulent transactions for autoencoder training. Then, two autoencoders are trained on the normal and fraud dataset, respectively. The samples are encoded by two autoencoders to obtain two sets of features, which are combined to form the dual autoencoding features. Finally, the model detects fraudulent transactions by a Neural Network trained on the augmented training set. Experimental results show that the model outperforms a set of well-known classification methods in experiments, especially the sensitivity and precision, which are effectively improved.

**INDEX TERMS** Fraud detection, imbalanced classification, generative adversarial networks, autoencoders.

## I. INTRODUCTION

With the continuous increase of online transactions via credit cards, more and more fraudulent transactions are increasingly produced, bringing great losses to banks, merchants, and cardholders. To reduce losses without affecting the user's trading experience, the organizations need to develop a useful fraud detection model that can detect fraudulent transactions as much as possible and avoid the misjudgments of normal transactions.

Fraud detection is usually seen as a binary classification problem of identifying suspicious usage patterns from transaction logs by using data mining and machine learning methods [1]–[6]. In particular, due to the excellent modeling capabilities of artificial neural networks (ANN), ANN methods have strong performances in various financial fraud detection problems, including credit card fraud [7]–[9], telecom fraud [10], insurance fraud [11], etc.

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

However, in the actual fraud detection dataset, the positive and negative samples are very imbalanced, and the extremely small number of fraudulent transaction records are available. This extremely imbalanced data may cause the classifier to produce biased results, because classifier may sacrifice the accuracy of the minority samples and treat them as noise [12]. However, these minority samples are what we are interested in and want to classify correctly.

In recent fraud detection researches, Generative Adversarial Networks (GANs) and autoencoders are widely used and have achieved considerable success. GANs can alleviate the imbalanced-class problem because of their abilities to approximate the actual data distribution and generate convincing data for the minority class. Ugo Fiore trained a GAN to output mimicked minority class examples to improve the classifier's effectiveness [13]. Autoencoders can improve detection accuracy and avoid manual feature reconstruction by projecting samples on the input space onto a feature space with a better representation [14]. Panpan Zheng developed one-class adversarial nets (OCAN) for fraud detection, which

adopts the autoencoder to learn the normal user representations and trains generative adversarial nets for distinguishing normal users and malicious users [15]. However, most methods in the fraud detection, like OCAN, focus their efforts on mining the feature information of normal samples and discard fraud samples, because the fraud samples are too few to be modeled by deep learning methods. There are few methods to mine the feature information of fraud samples. However, a small number of fraud samples also contain some important feature information. If the feature information of fraud samples can be used efficiently, normal and fraud samples may be better classified.

In order to make full use of the information of the samples in the dataset and alleviate the imbalanced-class problem, we proposed Dual Autoencoders Generative Adversarial Network (DAEGAN). DAEGAN firstly trains a GAN to output sufficient mimicked fraudulent transactions for autoencoder training. Then, it trains two autoencoders to learn feature representations on the normal and fraud dataset, respectively. The samples are encoded by two autoencoders to obtain two sets of features that are combined to form the dual autoencoding features. Finally, the model detects fraudulent transactions by a Neural Network trained on the augmented training set, which is generated by another GAN.

The advantages of DAEGAN for fraud detection are as follows. Firstly, DAEGAN can generate a collection of credible fraud samples, which can balance the majority and minority classes in the dataset. Secondly, DAEGAN mines the feature information of fraud samples based on the augmented fraud dataset. This improves the effectiveness of classifications. Thirdly, because the category information of a sample is unknown before classification, it cannot determine which encoder should encode a sample. DAEGAN innovatively uses dual feature features to combine the information learned by two autoencoders. So that more information can be used for classification. A careful experimental evaluation showed that the DAEGAN outperforms a set of well-known classification methods, especially the sensitivity and precision, which are effectively improved. While our framework is presented here in the context of credit card fraud detection, it should remark that it is quite general, and it can readily be extended to other application domains of the imbalanced dataset.

The rest of the paper is organized in the following sections. Related studies are described in Section II. A brief summary of GANs and autoencoder is provided in Section III. Section IV presents the proposed DAEGAN method for improving the accuracy of fraud detection. Section V presents the computational experiments, and Section VI concludes with a discussion.

## II. RELATED WORK

With more and more transactions data warehouses are available, fraud detection techniques have been developed in recent years. The main problem of credit card fraud detection is the imbalanced classification problem in the dataset. He and Garcia [12] mentioned that the extremely imbalanced data might cause the classifier to produce biased results and reduce the effectiveness of binary classifiers. Undersampling and oversampling are two significant methods of adjusting the imbalance in datasets [16]. Notably, the Synthetic Minority Oversampling Technique (SMOTE) [17], an oversampling technique, has received much attention because it can generate synthetic examples by interpolating between samples of the same class. SMOTE has given rise to several variants, such as Border SMOTE [18], DBSMOTE [19]. At present, with the great achievements of GANs in image generation [20]–[22] and image classification [23], [24], many studies have begun to use GANs in fraud detection to alleviate the imbalanced classification problem. GANs have shown satisfactory performance in generating credible samples [9], [13], [15].

In addition to balancing the majority and the minority classes by resampling, some researches proposed some semi-supervised methods based on autoencoders in anomaly detections [25]–[27]. They trained deep autoencoders on the data samples with no anomalies and detected anomalous events according to the reconstruction errors of anomalous and normal samples. The autoencoders can learn better representations of samples to improve classification effectiveness [14]. Ng *et al.* [28] proposed the Dual Autoencoding Features (DAF) to relieve the imbalance issue in a pattern classification problem. The OCAN also adopted the autoencoder to learn the normal user representations for better classification [15].

However, due to the small amount of fraud data, there are few methods to use feature learning models to mine the feature information of fraud samples. In this paper, we proposed a framework that combines GAN and autoencoders for improving the effectiveness of fraud detection classification. It can mine the feature information of fraud samples and alleviate imbalanced pattern classification problems by generating sufficient credible fraud samples. Meanwhile, our model uses the dual autoencoding features learned from two autoencoders to train a classifier for better performance.

## III. PRELIMINARY

### A. GENERATIVE ADVERSARIAL NETWORKS

GAN is a framework for the estimation of generative models through an adversarial process, which is firstly proposed by Goodfellow *et al.* [29]. GAN is based on the idea of the game theory, in which a generator $G$ and a discriminator $D$ are trained simultaneously and trying to outsmart each other. The generator $G$ continuously generates fake samples similar to the original data, and the discriminator $D$ estimates the probability that the samples come from the training data rather than $G$.

The generator builds a mapping from a prior noise distribution $p_z$ on a noise variable $z$ to a data space $G(z)$. $G$ generates fake samples and learns a generative distribution $p_G$ over the data $X$ to match the real data distribution $P_{data}$. The generator $G$ struggles to cheat the discriminator $D$ by

generating synthesized instances that appear to be as realistic as possible, in order to increase the error rate of its adversary. Thus, the objective function of $G$ is defined as:

$$\min_{\mathrm{G}} \mathbb{E}_{\mathrm{Z}\sim \mathrm{P_z}}[\log(1-\mathrm{D}\,(\mathrm{G}(\mathrm{z}))\,)] \tag{1}$$

where $\mathrm{D}(\cdot)$ outputs the probability that $\cdot$ is from the real data rather than the generated fake data. Minimizing Equation (1) is achieved if the discriminator is fooled by generated data $\mathrm{G}\,(z)$ and predicts a high probability that $\mathrm{G}\,(z)$ is real data.

The goal of the discriminator $D$ is to continuously improve classification effectiveness for distinguishing whether an input is a real data $x$ or fake data generated by a continuously improving generator. Hence, the objective function of $D$ is defined as:

$$\max_{D} \mathbb{E}_{X\sim P_{data}}\left[\log \mathrm{D}\,(x)\right]+\mathbb{E}_{Z\sim P_z}\left[\log\left(1-\mathrm{D}\,(\mathrm{G}(\mathrm{z}))\right)\right] \tag{2}$$

Overall, this cat-and-mouse game where both competitors improve their ability until an equilibrium is reached can be formalized as a minimax game:

$$\min_{C}\max_{D} \mathbb{E}_{X\sim P_{data}}\left[\log \mathrm{D}\,(x)\right]+\mathbb{E}_{Z\sim P_z}\left[\log\left(1-\mathrm{D}\,(\mathrm{G}(\mathrm{z}))\right)\right] \tag{3}$$

Training GAN is known not to be an easy task [30]. If the discriminator turns out to be significantly more effective than its generative counterpart, the entire GAN would not be correctly trained. If the discriminator turns out to be too weak, the generated fake data can fool the discriminator easily, and the generator will not be improved in the next round of training. Both components compete to prevail on the other one so that they strongly depend on each other for effective training. In the presence of a severe unbalancing where a component fails against the other, the whole GAN fails.

Arjovsky *et al.* [31] made the theoretical steps towards fully understanding why the generator faces the problem of gradient disappearance, and the collapse mode phenomenon exists in the original GAN proposed by Ian Goodfellow. He introduced the Wasserstein distance into the GAN. Because it has superior smoothing characteristics compared with KL divergence and JS divergence, it can theoretically solve the problem of gradients vanishing. WGAN not only solves the problem of unstable training, but also provides a reliable indicator of the training process, and this indicator is indeed highly related to the quality of the generated samples. The indicator is the loss function on training WGAN:

$$L = \mathbb{E}_{X\sim P_{data}}[f_w(x)] - \mathbb{E}_{Z\sim P_z}[f_w(G(z))] \tag{4}$$

where $f_w(\cdot)$ is a 1-Lipschitz continuous function, parameterized by $w$, that the discriminator model needs to learn. Accordingly, the loss functions of the generator and discriminator are (5) and (6).

$$L_G = -\mathbb{E}_{Z\sim P_z}[f_w(G(z))] \tag{5}$$

$$L_D = \mathbb{E}_{Z\sim P_z}[f_w(G(z))] - \mathbb{E}_{X\sim P_{data}}[f_w(x)] \tag{6}$$

Equation (4) is the inverse of Equation (6) and can indicate the training process. The smaller the value, the smaller the Wasserstein distance between the real and generated distributions, and the better the GAN is trained.

### B. AUTOENCODER

Autoencoder [32] is an unsupervised learning process that aims to transform inputs into outputs with the least possible amount of distortion. It plays a vital role in deep architectures for transfer learning and semi-supervised anomaly detections. Autoencoder contains an encoder and decoder.

$$y = f_\theta(x) \tag{7}$$

$$x' = g_{\theta'}(y) \tag{8}$$

Equation (7) and (8) are the calculation formulas for the encoder and decoder, respectively. Where $f$ and $g$ are affine mappings, and $\theta$ and $\theta'$ are vectors of weight and bias parameters of the encoder and the decoder, respectively. The goal of training autoencoder is to minimize the reconstruction error:

$$arg\min_{\theta\theta'} \mathbb{E}_{x\sim X}[\mathrm{L}(x, g_{\theta'}(f_\theta(x)))] \tag{9}$$

Typical choices for $\mathrm{L}(x, x')$ include the squared error $\|x-x'\|^2$ for real-valued vectors and the negative log-likelihood $\sum_{i=1}^{|x|}(x_i log x'_i+(1-x_i)\log(1-x'_i))$ for vectors of bits or bit probabilities (Bernoullis).

## IV. DUAL AUTOENCODERS GAN

DAEGAN contains three phases during training. As shown in the above side of Figure 1, the first phase is to train a WGAN to generate fraud data, which are then merged with training data into an augmented training set. Because WGAN has better training stability and provides a reliable indicator of the training process, we adopt WGAN as an important part of DAEGAN. The first WGAN contains the generator $G_I$ and the discriminator $D_I$. The $G_I$ and $D_I$ are optimized with loss functions Equation (5) and (6), respectively. The trained $G_I$ is fed with random noise $z$ and generates certain fake fraud samples $g\_f$ with the same dimensions as the real fraud samples:

$$g\_f = G_I(z) \tag{10}$$

The generated fraud samples $g\_f$ will be merged with real fraud samples $r\_f$ into an augmented fraud training set $\mathrm{x}\_f$:

$$x\_f = g\_f \cup r\_f \tag{11}$$

The fraud training set $x\_f$ will be used to train the autoencoder of the fraud data $\mathrm{AE}\_f$. It will help the autoencoder to learn good representations of the fraud data and avoid underfitting caused by too few fraud samples in the data set.

The second phase is to train two autoencoders to learn the representations of the normal and fraud samples, as shown in the upper right part of Figure 1. Because the number of fraud samples data is too small, many methods use one-class classification, which firstly uses only one autoencoder to learn the representations of the normal samples, and then identifies abnormal samples by the size of the reconstructed error. Although such methods are a very effective strategy in

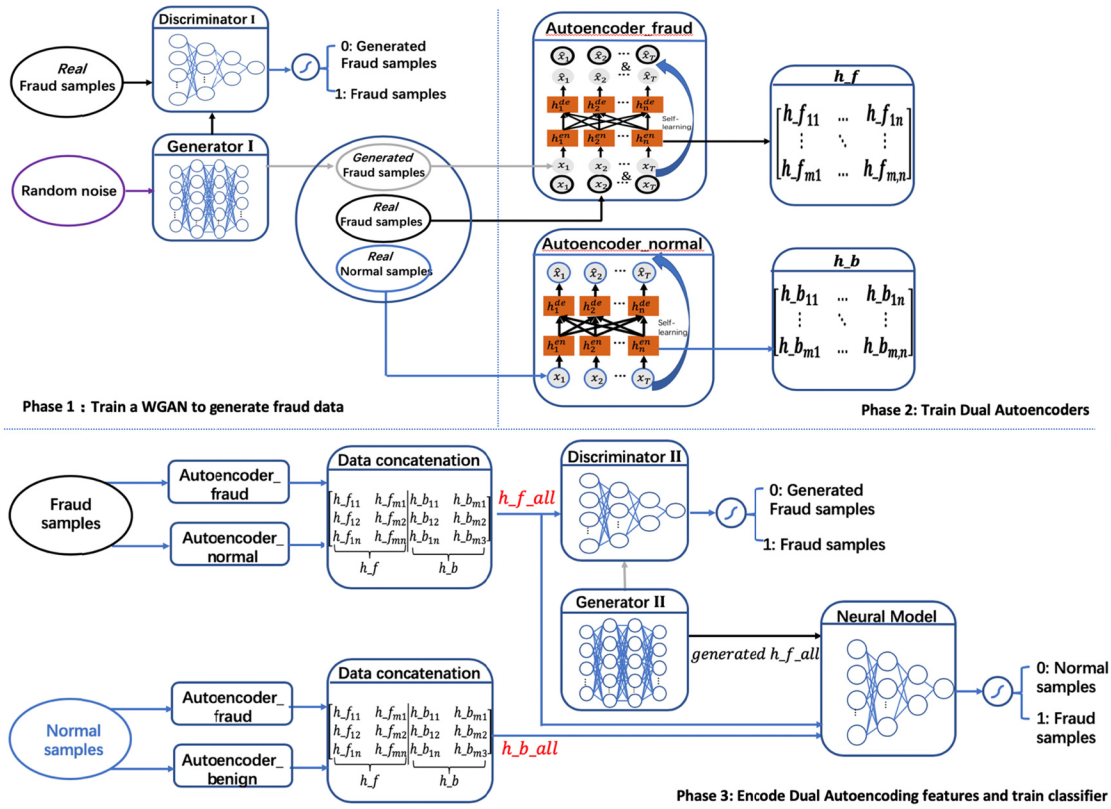



**FIGURE 1.** The architecture of the DAEGAN.

the absence of minority class, when there are still some fraud samples, it is helpful to distinguish normal and fraud samples by mining the feature information of fraud samples as much as possible. Therefore, in order to solve the problem that the autoencoder cannot completely fit the fraud samples data, we propose to train the autoencoder $AE\_f$ on the augmented fraud training set $x\_f$, which contains the real fraud samples and fake fraud samples generated by the first WGAN:

$$AE\_f = arg \min_{\theta\theta'} \mathbb{E}_{x\_f \sim X\_f}[\left\| x\_f, g_{AE\_f}(f_{AE\_f}(x\_f)) \right\|^2] \tag{12}$$

Also, we train the autoencoder $AE\_b$ to learn the representations of the normal samples on the real normal training set $x\_b$:

$$AE\_b = arg \min_{\theta\theta'} \mathbb{E}_{x\_b \sim X\_b}[\left\| x\_b, g_{AE\_b}(f_{AE\_b}(x\_b)) \right\|^2] \tag{13}$$

where $f_{AE\_b}$ and $g_{AE\_b}$ are the encoder and decoder of $AE\_b$, $f_{AE\_f}$ and $g_{AE\_f}$ are the encoder and decoder of $AE\_f$. Autoencoder is an unsupervised learning algorithm, so the inputs and outputs of $AE\_f$ and $AE\_b$ are the raw features of users in their respective categories.

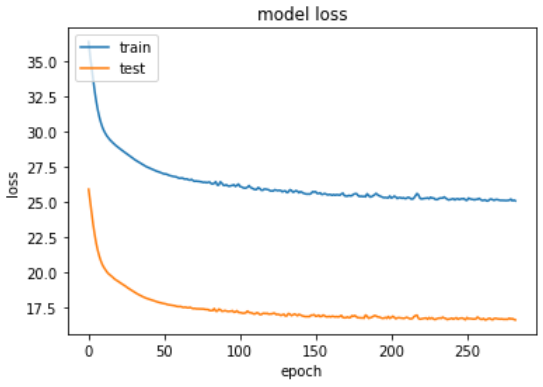
Figure 2 illustrates that the augmented data set can help the autoencoder $AE\_f$ improve the ability to fit data. Since there are too few samples in the actual fraud data set, the neural networks like autoencoder will underfit. As Figure 2 (a) shows, the training and testing errors are large, and such an autoencoder cannot be used to learn representations of fraud samples. The generator $G_I$ of the first WGAN can generate the fake fraud samples that have a similar distribution of real fraud samples in feature space. A sufficient amount of fake fraud samples can complement the fraud data set so that the autoencoder can fit the dataset well and learn the representations of fraud samples. As Figure 2 (b) shows, the errors of both the training set and the test set are reduced after training the $AE\_f$ on the augmented training set.

When $AE\_b$ and $AE\_f$ have been obtained, the representations of the normal samples and fraud samples will be computed by the encoders of the $AE\_b$ and $AE\_f$, respectively:

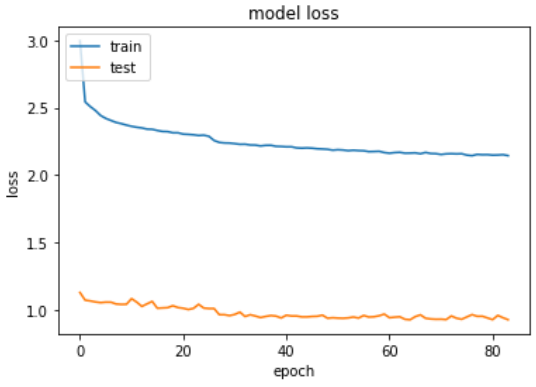
$$h\_b = f_{AE\_b}(x\_b) \tag{14}$$

$$h\_f = f_{AE\_f}(x\_f) \tag{15}$$

where $h\_b$ is the representation of the data encoded by the $AE\_b$, and $h\_f$ is the representation of the data encoded by the $AE\_f$. The representations of the normal samples and fraud samples are more separable in the continuous feature space because the autoencoders project samples on the input space onto a feature space with a better representation.




(a) The errors of $AE\_f$ trained on actual fraud data set




(b) The errors of $AE\_f$ trained on augmented fraud data set

**FIGURE 2.** The training and testing error of $AE\_f$ trained on actual fraud data set and augmented fraud data set.

However, during the training process, we cannot bring the category information into the training data, which will cause data leakage. So, we cannot determine whether $AE\_b$ or $AE\_f$ should encode a sample. To obtain as much data information as possible, we encode the sample by $AE\_b$ and $AE\_f$ to achieve the $h\_b$ and $h\_f$. Then, we concatenate them together as the final representation $h\_all$ of this sample:

$$h\_all = \left[\begin{array}{ccc|ccc} h\_f_{11} & \cdots & h\_f_{1n} & h\_b_{11} & \cdots & h\_b_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h\_f_{m1} & \cdots & h\_f_{m,n} & h\_b_{m1} & \cdots & h\_b_{m,n} \end{array}\right] \tag{16}$$

As shown in the below side of Figure 1, given the representations $h\_f\_all$ and $h\_b\_all$ of the fraud and normal samples, the third phase of DAEGAN is to train another WGAN and a neural network model that can distinguish the normal and fraud samples. The generator of the second WGAN aims to generate complementary representations of fraud samples that are in the dimension of the representation $h\_all$:

$$g\_h\_f = G_{II}(z) \tag{17}$$

where $G_{II}$ is the generator of the second WGAN. The discriminator of the second WGAN aims to separate the real and complementary fraud representations of samples. Finally, $g\_h\_f$, $h\_f\_all$, and $h\_b\_all$ will form an augmented dataset for training a neural network. The neural network can detect fraud samples which locate in separate regions from normal samples. The neural network model uses a SoftMax classifier, and the loss function of it is cross-entropy:

$$L\left(y, \hat{y}\right) = -\frac{1}{N}\sum\nolimits_{n\in N} y_n log\hat{y}_n \tag{18}$$

where $y$ is the true labels, and $\hat{y}$ is the predicted labels.

The pseudo-code of training DAEGAN is shown in Algorithm 1. Given a training set $M_{benign}$ and $M_{fraud}$, that contain feature vectors of normal and fraud samples, we first train a WGAN (Lines 2-6) to generate the complementary fraud samples $g\_f$. Then, we merge $g\_f$ with real fraud samples into an augmented training set $M'_{fraud}$ to train the autoencoder $AE\_f$ (Lines 10-14). We use the same method to train another autoencoder $AE\_b$ on training set $M_{benign}$ (Lines 15-19). After training the two autoencoders, we encode each sample in the $M_{benign}$ and $M_{fraud}$ to obtain $h\_b$ and $h\_f$. Then we concatenate them together as the representation of the sample (Line 20-27). Finally, we train another WGAN and neural network model at the same time (Line 28-39). The WGAN aims to generate useful fraud samples representations to improve the classification of the NN model, and the NN model aims to improve the ability to detect fraud samples from normal samples. For simplicity, we write the algorithm with a minibatch size of 1, i.e., iterating each sample in the training set to train all the models. In practice, we sample $n$ samples in a minibatch, such as 128.

Although DAEGAN's training process is a bit complicated, both the WGANs and autoencoders are trained for a neural network model with excellent classification. After training, the actual fraud detection is straightforward. As Figure 3 shows, a new transaction can be directly determined whether to be fraud or normal by the NN model based on its representation encoded by the dual autoencoders.

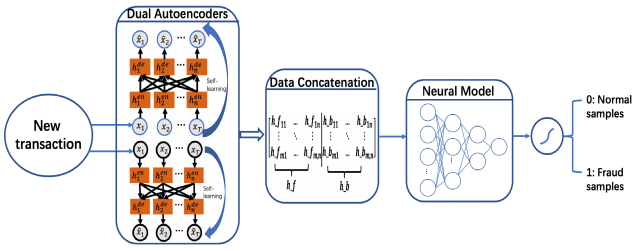



**FIGURE 3.** The detection of a new transaction.

## V. EXPERIMENTS

To assess the effectiveness of the proposed DAEGAN, we conducted our validation on a credit card transaction dataset [33], which contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced; the positive class (frauds) account for

**Algorithm 1:** Training DAEGAN

**Inputs:** Training dataset $M_{benign}$, $M_{fraud}$

initialize parameters in WGANs, autoencoders, and Neural Network model;
**for** iteration $= 1$ **to** $Epoch_{WGAN}$ **do**
  **for each** $m_f$ in $M_{fraud}$ **do**
    optimize the discriminator $D_I$ and generator $G_I$ with loss functions (5) and (6), respectively.
  **end for**
**end for**
$G_I$ generates complementary fraud samples $g\_f$
//merge $g\_f$ with $r\_f$ into a training set
$M'_{fraud} = M_{fraud} \cup g\_f$
**for** iteration $= 1$ **to** $Epoch_{AE\_f}$ **do**
  **foreach** $m_{f'}$ in $M'_{fraud}$ **do**
    optimize the parameters in $AE\_f$ with loss function (12)
  **end for**
**end for**
**for** iteration $= 1$ **to** $Epoch_{AE\_b}$ **do**
  **foreach** $m_b$ in $M_{benign}$ **do**
    optimize the parameters in $AE\_b$ with loss function (13)
  **end for**
**end for**
$\mathcal{V}_{benign} = \emptyset$, $\mathcal{V}_{fraud} = \emptyset$;
**foreach** $m$ in $M_{fraud}$ and $M_{benign}$ **do**
  //compute the representation encoded by $AE\_b$ and $AE\_f$
  $h\_b = f_{AE\_b}(m)$; $h\_f = f_{AE\_f}(m)$
  concatenate $h\_b$ and $h\_f$ with function (16)
  $\mathcal{V}_{benign} += h\_b\_all$, $\mathcal{V}_{fraud} += h\_f\_all$
**end for**
**for** iteration $= 1$ **to** $Epoch_{WGAN}$ **do**
  **foreach** $v_{fraud}$ in $\mathcal{V}_{fraud}$ **do**
    optimize the discriminator $D_{II}$ and generator $G_{II}$ with loss functions (5) and (6), respectively.
  **end for**
  generate $g_{h_f}$ with function (17)
  $\mathcal{V} = g_{h_f} \cup \mathcal{V}_{fraud} \cup \mathcal{V}_{benign}$
  **for** iteration $= 1$ **to** $Epoch_{NN}$ **do**
    **foreach** $v$ in $\mathcal{V}$ **do**
      optimize the parameters in neural network model with loss function (18).
    **end for**
  **end for**
**end for**
**return** Well-trained WGANs, autoencoders, and a Neural Network model

0.172% of all transactions. Due to a confidentiality request by the institution releasing the data, the Credit-card data contains numerical features, labeled $V1$ to $V28$, which are the principal components resulting from Principal Components Analysis (PCA) applied to the original features. The only features which have not been transformed with PCA are 'Time', 'Amount', and 'Class'. Before modeling the data, we preprocessed the data and removed some abnormal points. Then, we divided 80% of the majority and minority class data into the training set and the remaining 20% of the data into the validation set, as Figure 4 shows. In the experiments, because the size of the fraud data set is too small, we used WGAN to generate fake fraud data to supplement the fraud training set. In order to avoid data leakage, we only use the fraud training set, not all fraud data set to train the WGANs, and the validation set is only used for testing the fraud detection ability of the neural network model.

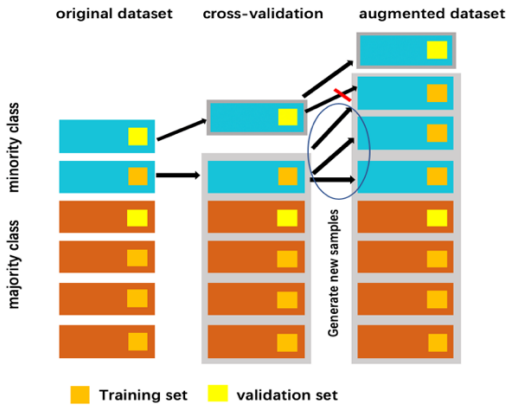



**FIGURE 4.** The data sets of the experiments.

Metrics used in the experiments include the recall, the precision, the F1, the Area Under the Curve (AUC), the Area Under Precision-Recall Curve (AUPRC). The AUC is defined as the area under the ROC curve. The AUPRC is defined as the area under the Precision-Recall Curve. Given the class imbalance ratio, confusion matrix accuracy is not meaning for unbalanced classification, and the AUPRC is recommended to measure the accuracy of the model. It is worth to remind that in applications such as credit card fraud detection, the cost of a false positive and a false negative are not equal. An ideal fraud detection system should identify precisely the fraudulent transactions and reduce the number of false-positive that require control by human investigators. Therefore, an excellent model should have good performance in each of the above indicators to prove that the model has a good accuracy of identifying fraud samples while avoiding the misjudgments of normal samples.

### A. DATA PREPROCESSING

Before modeling data, we preprocessed the data to make it more suitable for modeling. We will only explain some unique steps rather than some routine steps on preprocessing.

Firstly, we used the xgboost to calculate the feature importance of the dataset. As Figure 5 shows, $V14$, $V4$, and $V10$ are the three most important features, and we removed some outliers which are not in the 20%-80% of them. The feature 'Time' is seconds from the first transaction in the dataset, and we converted it to time of day in hours, as Figure 6 shows. We further preprocessed the Credit-card dataset to rescale the features in the interval [0,1]. The resulting dataset contained 457 fraudulent transactions out of 283009 transactions.

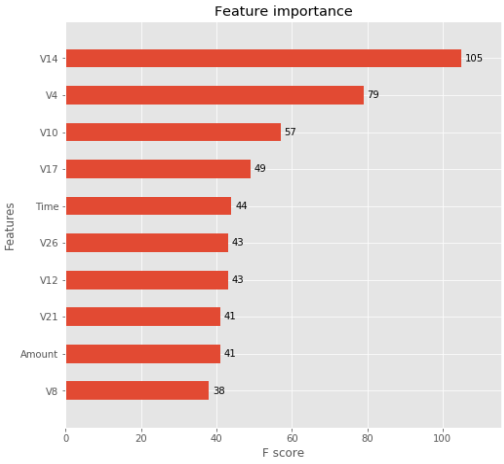



**FIGURE 5.** The feature importance of the data.

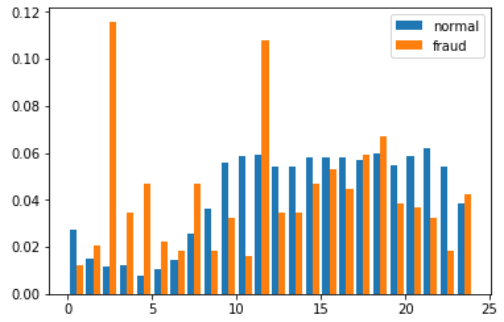



**FIGURE 6.** The distribution of the feature 'Time'.

### B. COMPARISON OF GAN AND WGAN

We compared WGAN with GAN to find which one is suitable for credit card fraud detection. In our experiments, the WGAN and GAN have the same network structure. The discriminators of GAN and WGAN all have three hidden layers, which are 800, 400, 200 dimensions. The output layer of the GAN's discriminator is a sigmoid function which outputs the probability that the sample is from the real data rather than the generated fake data. The output layer of the WGAN's discriminator removes the sigmoid, and its output is an approximate fitted Wasserstein distance. The generators of GAN and WGAN all take the 50 dimensions of noise as input, and also have three hidden layers, which are 200, 400, 800 dimensions. The output layers of the generators have the same dimension as the input layers, which are 30 in our experiments. The loss functions of GAN's discriminator and generator are Equation (1) and (2). The loss functions of WGAN's discriminator and generator are Equation (5) and (6). We trained the GAN and WGAN on the same training set with 5000 training epochs. After each iteration, the WGAN discriminator parameters were updated, and then their absolute values were truncated to no more than a fixed constant. The optimization algorithm of WGAN does not use the optimization algorithms based on momentum, such as momentum and Adam, but uses RMSProp.

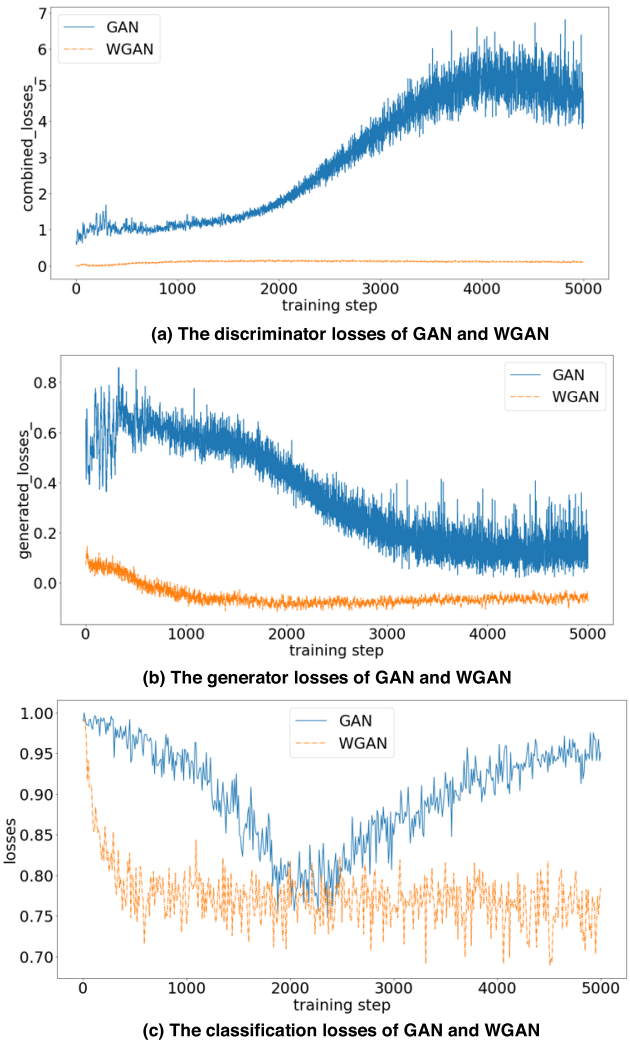



**FIGURE 7.** The losses of GAN and WGAN in training.

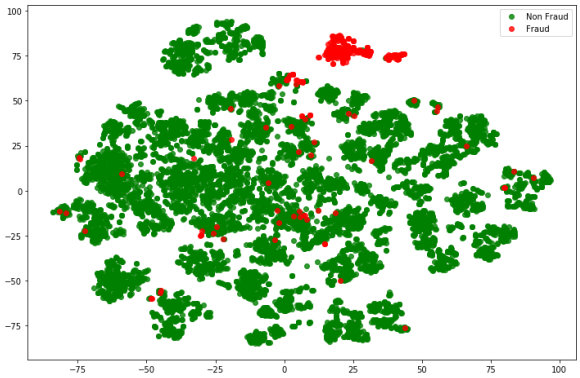
The losses of the discriminators and generators are shown in Figure 7. During the training process, the losses of WGAN are smaller than those of GAN. WGAN's training process is more stable than GAN's, and its convergence speed is faster than GAN. Also, we further checked for the quality of the generated data using a neural network with three hidden layers, which are also 800, 400, 200. The losses of the NN model are shown in Figure 7(c). In order to further

compare the qualities of the data generated by WGAN with GAN, we compared the performance of the DAEGAN using WGAN with the DAEGAN using GAN in fraud detection. The comparison results are shown in Table 1. DAEGAN using WGAN is superior to DAEGAN using GAN in all indicators. Therefore, the quality of the data generated by WGAN is better, which can improve the performance of the NN model classifier.

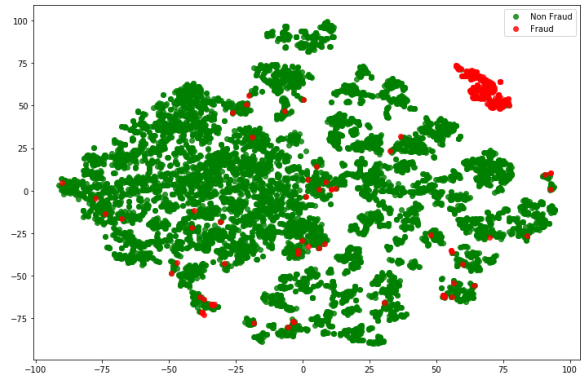
### C. SUPERIORITY OF DUAL AUTOENCODERS

We compared dual autoencoders with only one autoencoder to verify the advantages of introducing one more autoencoder. In our experiments, the autoencoders $AE\_b$ and $AE\_f$ have different network structures. The autoencoder $AE\_b$ has three hidden layers, which are 100, 50,100 dimensions, and the autoencoder $AE\_f$ also has three hidden layers, which are 70, 50, 70 dimensions. The encoders of $AE\_b$ and $AE\_f$ both have the output layer of 50 dimensions, which maps input dimensions to high dimensions to improve data separability. The autoencoder $AE\_b$ was trained on real normal samples, and $AE\_f$ was trained on an augmented data set which merged the real fraud samples with the fake fraud samples generated by the WGAN. The fake fraud samples generated by the WGAN can complement the training set of $AE\_f$ to avoid underfitting. As Figure 2 shows, the fake fraud samples generated by the WGAN play an important role in reducing the losses of $AE\_f$ during training and preventing $AE\_f$ from underfitting.

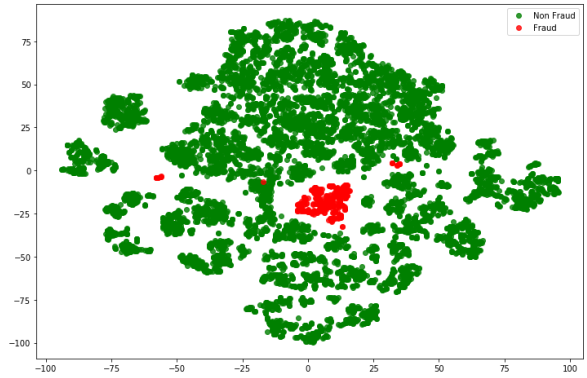
For a more intuitive comparison, we use the t-SNE tool [34] (Maatten and Hinton 2008) to visualize the distributions of the original data, the representations of data computed by only $AE\_b$, and the dual autoencoding features, which combine the representations of data computed by $AE\_b$ and $AE\_f$. As shown in Figure 8, the dual autoencoding features are more separable than the original data and the representations of data computed by only $AE\_b$. It is easier to distinguish the fraud samples from normal samples based on the features computed by dual autoencoders. We also compared the experimental results of DAEGAN with the AEGAN, which has the same network structure with DAEGAN but is missing $AE\_f$. As Table 1, Figure 9 and 10 show, the DAEGAN outperforms the AEGAN in all indicators. It can be seen that by introducing $AE\_f$, DAEGAN can make full use of data to extract more information and improve the classification.




(a) The distribution of the original data.




(b) The distribution of the representations of data (only $AE\_b$)




(c) The distribution of the dual autoencoding features of data ($AE\_b$ and $AE\_f$)

**FIGURE 8.** 2D visualization of three types of data: the original data, the representations of data computed by $AE\_b$, and the representations of data computed by $AE\_b$ and $AE\_f$.

### D. COMPARISON WITH OTHER SEVERAL METHODS

**Baselines**: We compared DAEGAN with the following widely used methods [35] for alleviating the imbalanced-class problem:

- Undersampling reduces the disparity between classes by undersampling the majority class in a training set [17], randomly removing some instances of the majority class.
- Oversampling is to oversample the minority class, which often uses the Synthetic Minority Oversampling Technique (SMOTE) [18] to generate synthetic examples by interpolating between examples of the same class.

In the undersampling method, we randomly selected samples of the majority class to make the ratio of positive and negative samples be 1:1. Finally, we trained a neural network model that has the same network structure with the NN model in DAEGAN. As Table 1 shows, although the undersampling method has a very high recall, the precision is very low,

**TABLE 1.** Fraud detection results on recall, precision, F1, AUC, AUPRC.

| Algorithm | recall | precision | F1 | AUC | AUPRC |
|---|---|---|---|---|---|
| **DAEGAN** | **0.815** | **0.903** | **0.857** | **0.958** | **0.805** |
| AEGAN | 0.772 | 0.910 | 0.835 | 0.910 | 0.683 |
| Undersample_NN | 0.859 | 0.025 | 0.048 | 0.956 | 0.163 |
| SMOTE_NN | 0.75 | 0.495 | 0.596 | 0.956 | 0.700 |
| WGAN | 0.782 | 0.837 | 0.808 | 0.945 | 0.675 |
| OCAN | 0.808 | 0.258 | 0.390 | 0.901 | 0.405 |

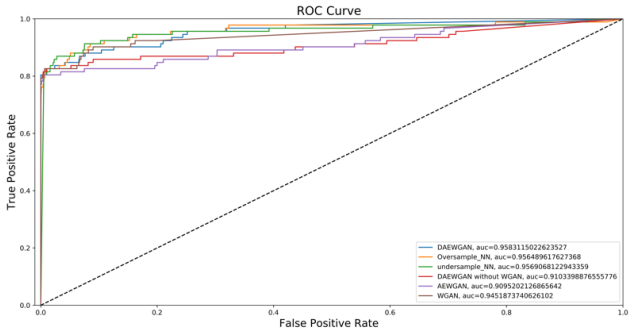



**FIGURE 9.** ROC curves of DAEGAN and several other models.

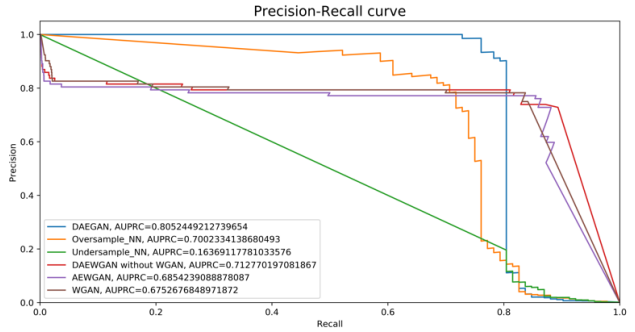



**FIGURE 10.** PRC curves of DAEGAN and several other models.

which means that a lot of normal samples are misidentified as fraud samples. As Figure 10 shows, the AUPRC of the undersampling method is also very low. In the oversampling method, we used the SMOTE method to generate synthetic samples which were merged with real samples into an augmented dataset. Then, we also trained a neural network model on the new training dataset. As Table 1 shows, the results of SMOTE method are all worse than DAEGAN's. In summary, the method DAEGAN is superior to the undersampling and oversampling in various indicators. DAEGAN performs better than the commonly used methods for alleviating the imbalanced-class problem.

We also compared the DAEGAN with other methods of credit card fraud detection using GANs. The results of the GAN trained by Ugo Fiore are shown in Table 2. After comparison, we found that our proposed DAEGAN method

**TABLE 2.** Fraud detection results of the GAN trained by Ugo Fiore.

| $N_g$ | Recall | Precision | F-measure |
|---|---|---|---|
| 0 | 0.70229 | **0.97872** | 0.81778 |
| 79 | 0.70229 | 0.96842 | 0.81416 |
| 158 | 0.71756 | 0.93069 | 0.81034 |
| 315 | 0.72519 | 0.91346 | 0.80851 |
| 630 | **0.73282** | 0.93204 | **0.82051** |
| 945 | 0.72519 | 0.93137 | 0.81545 |
| 1260 | 0.72519 | 0.93137 | 0.81545 |
| 2520 | 0.72519 | 0.93137 | 0.81545 |
| 3150 | 0.73208 | 0.93182 | 0.81883 |
| 6300 | 0.72519 | 0.94059 | 0.81897 |

is better on various indicators. The recall of DAEGAN is 0.815, and it is better than the recall of Ugo Fiore's method, which is 0.73282. It means that the DEGAN can detect more fraud samples than Ugo Fiore's method. In terms of precision, Ugo Fiore's method is higher. However, due to the small number of fraud samples, this degree of gap cannot explain that compared with Ugo Fiore's method, DAEGAN misjudges excessive normal samples as fraud samples. In F-measure, DAEGAN is 0.857, and the best score of Ugo Fiore's method is 0.82051. It shows that DAEGAN is a better method than Ugo Fiore's in the overall performance of the model.

We also compared the DAEGAN with the WGAN. We firstly trained the WGAN to generate fake fraud samples to supplement the fraud data set and then used the NN model trained on the augmented data set for fraud detection. The results of the method only using WGAN show in Table 1. The Figure 9 and 10 also show the AUC and PRC of the method. We can observe that the results of the DAEGAN are also better than WGAN. It means the DAEGAN is better than the methods which only use GANs to generate fraud samples to improve the classification. The autoencoders play a crucial role in improving the effect of classification based on GAN.

We compared the OCAN proposed by Panpan Zheng with our method DAEGAN. The OCAN is a one-class classification which has an autoencoder and GAN. We adopted 700 genuine transactions as a training dataset and 120 fraud and 71082 genuine transactions as a testing dataset, which has the same distribution as the real data. As Table 1 shows, the DAEGAN outperforms the OCAN.

Above all, DAEGAN has achieved the best performance for fraud detection. In particular, the AUC of DAEGAN is 0.958, and the AUPRC of DAEGAN is 0.805. DAEGAN also has higher recall and precision at the same time, which are 0.815 and 0.903, respectively. It means that DAEGAN improves the accuracy of fraud samples detection while avoiding the increasing misjudgments of normal samples.

## VI. CONCLUSION

In this work, we proposed a new neural network model DAEGAN to cope with imbalanced classification problem in credit card fraud detection. DAEGAN adopts WGAN to generate sufficient and credible fraudulent transactions for balancing the minority and majority classes in the dataset. Thanks to the complementary fraud transactions generated by WGAN, the autoencoder of fraud transactions can mine the feature information without underfitting. The dual autoencoders in the DAEGAN can learn the representations of the normal and fraud samples. The dual autoencoding features combining learned feature sets yield better feature representations of samples so that the classifier can achieve better performance. After training, the neural network classification can detect fraudulent transactions in high sensitivity and precision. We have conducted the theoretical and empirical analysis to demonstrate that WGAN can generate more credible fake fraud samples than GAN, and dual autoencoders can yield better feature representations of samples than only one autoencoder. We conducted experiments over real credit card dataset and showed the DAEGAN outperforms representative resampling-based methods and the state-of-the-art fraud detection classification models. The proposed method can be further extended to alleviate the imbalanced-class problems. In our feature work, we plan to optimize the solution to make the whole process lighter and more effective.

## REFERENCES

[1] M. Behdad, L. Barone, M. Bennamoun, and T. French, "Nature-inspired techniques in the context of fraud detection," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1273–1290, Nov. 2012.

[2] M. T. El-Melegy, "Model-wise and point-wise random sample consensus for robust regression and outlier detection," *Neural Netw.*, vol. 59, pp. 23–35, Nov. 2014.

[3] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, Feb. 2011.

[4] M. R. G. Raman, N. Somu, K. Kirthivasan, and V. S. S. Sriram, "A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems," *Neural Netw.*, vol. 92, pp. 89–97, Aug. 2017.

[5] Q. Song, Y.-J. Zheng, Y. Xue, W.-G. Sheng, and M.-R. Zhao, "An evolutionary deep neural network for predicting morbidity of gastrointestinal infections by food contamination," *Neurocomputing*, vol. 226, pp. 16–22, Feb. 2017.

[6] Y.-J. Zheng, H.-F. Ling, J.-Y. Xue, and S.-Y. Chen, "Population classification in fire evacuation: A multiobjective particle swarm optimization approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 70–81, Feb. 2014.

[7] V. V. Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, and M. Snoeck, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decis. Support Syst.*, vol. 75, pp. 38–48, Jul. 2015.

[8] A. Zakaryazad and E. Duman, "A profit-driven artificial neural network (ANN) with applications to fraud detection and direct marketing," *Neurocomputing*, vol. 175, pp. 121–131, Jan. 2016.

[9] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, and S.-Y. Chen, "Generative adversarial network based telecom fraud detection at the receiving bank," *Neural Netw.*, vol. 102, pp. 78–86, Jun. 2018.

[10] A. Mohamed, A. F. M. Bandi, A. R. Tamrin, M. D. Jaafar, S. Hasan, and F. Jusof, "Telecommunication fraud prediction using backpropagation neural network," in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, Malacca, Malaysia, Dec. 2009, pp. 259–265.

[11] W. Xu, S. Wang, D. Zhang, and B. Yang, "Random rough subspace based neural network ensemble for insurance fraud detection," in *Proc. 4th Int. Joint Conf. Comput. Sci. Optim.*, Apr. 2011, pp. 1276–1280.

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

[13] 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.

[14] M. Wasikowski and X.-W. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1388–1400, Oct. 2010.

[15] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, "One-class adversarial nets for fraud detection," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 1286–1293.

[16] 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.

[17] 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.

[18] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, vol. 3644, Oct. 2005, pp. 878–887.

[19] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "DBSMOTE: Density-based synthetic minority over-sampling technique," *Int. J. Speech Technol.*, vol. 36, no. 3, pp. 664–684, Apr. 2012.

[20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[22] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," 2015, *arXiv:1511.06390*. [Online]. Available: http://arxiv.org/abs/1511.06390

[23] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," 2016, *arXiv:1606.03657*. [Online]. Available: http://arxiv.org/abs/1606.03657

[24] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," 2016, *arXiv:1610.09585*. [Online]. Available: http://arxiv.org/abs/1610.09585

[25] D. Wulsin, J. Blanco, R. Mani, and B. Litt, "Semi-supervised anomaly detection for EEG waveforms using deep belief nets," in *Proc. 9th Int. Conf. Mach. Learn. Appl.*, Dec. 2010, pp. 436–441.

[26] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 665–674.

[27] R. Chalapathy, A. Krishna Menon, and S. Chawla, "Anomaly detection using one-class neural networks," 2018, *arXiv:1802.06360*. [Online]. Available: http://arxiv.org/abs/1802.06360

[28] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, "Dual autoencoders features for imbalance classification problem," *Pattern Recognit.*, vol. 60, pp. 875–889, Dec. 2016.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Systems.*, Dec. 2014, pp. 2672–2690.

[30] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Jan. 2017, pp. 1–17.

[31] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[32] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, Dec. 2007, pp. 153–160.

[33] Kaggle. (2018). *Credit Card Fraud Detection Dataset*. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud.

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[35] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Structural, Syntactic, and Statistical Pattern Recognition* (Lecture Notes in Computer Science), vol. 3138. Berlin, Germany: Springer-Verlag, Nov. 2004, pp. 806–814.

**ENSEN WU** (Member, IEEE) is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. His current research interests include financial data analysis and data mining.

**HONGYAN CUI** (Senior Member, IEEE) received the Ph.D. degree in circuit and systems from the Beijing University of Posts and Telecommunications, in 2006. She is currently a Ph.D. Supervisor. She is also a Visiting Scholar with MIT. Her research areas include big data analysis and visualization, intelligent resource management in future networks, cloud technology, and block chain.

**ROY E. WELSCH** received the Ph.D. degree from Stanford University, in 1969. He is currently the Eastman Kodak Leaders for Global Operations Professor of Management and a Professor of Statistics and Data Science at the MIT Sloan School of Management and the MIT Center for Statistics and Data Science. He is also the Director of the MIT Center for Computational Research in Economics and Management Science. He is widely recognized for his book, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, (with Edwin Kuh and David Belsley) on regression diagnostics and for his work on robust estimation, multiple comparison procedures, nonlinear modeling, and statistical computing. He is currently involved in the research on robust process control and experimental design, credit-scoring models and risk assessment, diagnostics for checking model and design assumptions, volatility modeling in financial markets, uncertainty quantification in plots and images, and repurposing existing drugs or combinations of drugs to prevent or delay Alzheimer's disease by creating synthetic clinical trials based on electronic health records.