Computing and Informatics, Vol. 43, 2024, 213–239, doi: 10.31577/cai_2024_1_213

HS-CGK: A HYBRID SAMPLING METHOD FOR IMBALANCE DATA BASED ON CONDITIONAL TABULAR GENERATIVE ADVERSARIAL NETWORK AND K-NEAREST NEIGHBOR ALGORITHM

Xiaoyan Zhao, Shaopeng Guan*, Yuewei Xue, Hao Pan

School of Information and Electronic Engineering
Shandong Technology and Business University
Yantai 264005, China
e-mail: 330020920@qq.com, konexgsp@gmail.com

Abstract. Class imbalance problem in datasets can lead to biased classification decisions in favor of majority class samples. Additionally, class overlap can cause fuzzy classification boundaries, affecting the performance of classification algorithms. To address these issues, we propose a hybrid sampling method based on conditional tabular generative adversarial network (CTGAN) and K-nearest neighbor (KNN) algorithm. Firstly, we introduce an oversampling algorithm, named DB-CTGAN, based on CTGAN. This algorithm filters noisy and boundary samples using the density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm and generates synthetic samples that conform to the real data distribution using CTGAN. Finally, we combine the expanded fraudulent samples generated by DB-CTGAN with the normal samples and use the KNN overlap undersampling algorithm to remove the samples in the overlap region, solving the class overlap problem. Experimental results show that compared with eight sampling methods using four standard classification models (Random Forest, Decision Tree, Support Vector Classification, and XGBoost), the proposed method significantly improves the F1, AUC, and G-mean metrics on five real datasets.

Keywords: Imbalanced data, class overlap, conditional tabular generative adversarial network, K-nearest neighbor algorithm, hybrid sampling

* Corresponding author

1 INTRODUCTION

Credit card fraud not only causes losses to individuals and financial institutions, but also jeopardizes the healthy development of the credit card industry. Accurate detection of fraudulent transactions is important to protect the interests of consumers and financial institutions. Currently, machine learning and data mining [1, 2, 3, 4, 5, 6], among others, are the dominant techniques for credit card fraud transaction detection. These techniques view fraud detection as a binary classification problem. However, the number of normal transactions in credit card transaction data is usually much higher than the number of fraudulent transactions, i.e., there is an imbalance problem in the dataset. The class imbalance problem affects the performance of fraud detection algorithms by biasing the classification results toward the majority class samples at the expense of the classification accuracy of the minority class samples. The harm arising from misidentifying fraudulent transactions is much greater than the harm arising from misidentifying normal transactions. Therefore, the accurate identification of minority class samples is more important in the field of fraud detection.

Several solutions have been proposed to solve the class imbalance problem, mainly divided into data-level methods and algorithm-level methods [7]. The data level focuses on data preprocessing operations to construct balanced datasets mainly through resampling techniques. The algorithm level is used to improve the recognition accuracy of classification algorithms for minority classes of samples by constructing new algorithms or improving existing algorithms. The algorithm-level approach requires in-depth knowledge of classification algorithms and loss functions, which are less applicable [8]. In contrast, data-level methods are independent of classification algorithms, which are more generalizable and easier to implement. Therefore, this paper focuses on data-level methods.

Credit card data is a type of tabular data that contains both data variables and categorical variables. However, traditional data-level methods such as oversampling and undersampling create new problems when resampling credit card data. For instance, the synthetic data generated by oversampling does not reflect the true data distribution well, while undersampling tends to lose some important information. To address these problems, Generative Adversarial Network (GAN) [9] has been proposed as a new perspective for solving class imbalance problems [10]. GAN-based oversampling methods can capture the distribution characteristics of the input data and generate synthetic data that approximates the real data, thus avoiding the problems of insufficient data diversity and overfitting that exist in traditional sampling algorithms. However, GAN is not a good solution to the problems such as class overlap and noise that exist in imbalanced datasets [11].

In credit card transactions, fraudsters constantly vary their fraudulent behavior to make it as identical as possible to the normal transaction behavior, resulting in a blurred boundary between normal and fraudulent transactions and causing class overlap in the dataset. The class overlap problem can lead to fuzzy classification boundaries and deteriorating classifier performance. In addition, there is a noise

problem in the credit card dataset, and noisy data can affect the classification performance of the classifier.

To address the problems of class overlap and noise in imbalanced credit card datasets, we propose a hybrid sampling algorithm based on CTGAN and KNN: HS-CGK. First, we designed a CTGAN-based DB-CTGAN sampling algorithm to construct the balanced dataset, and then used the KNN overlap undersampling algorithm to remove the majority class samples in the overlap region. Our main contributions are summarized as follows:

- We designed an oversampling algorithm DB-CTGAN for table data generation. Firstly, we used the DBSCAN clustering algorithm to filter the noisy samples and boundary samples in the dataset, and then used CTGAN to learn the minority class samples after filtering, and generate synthetic samples conforming to the real data distribution.
- The DB-CTGAN oversampling algorithm solves the imbalance problem in the dataset but it can aggravate the class overlap phenomenon and cause the classification boundary to be blurred. To eliminate the negative effects of class overlap, we proposed to use the KNN overlap undersampling algorithm to remove most of the class samples in the overlap region.
- We evaluated the performance of the proposed HS-CGK method. Experimental results on four commonly used fraud detection models, RF, DT, SVM, and XGBoost, show that the HS-CGK algorithm achieves optimal results on the F1, AUC, and G-mean metrics.

The rest of the paper is organized as follows: Section 2 discusses the existing class imbalance treatment methods. Section 3 introduces the basic principles of CTGAN. Section 4 presents the proposed HS-CGK sampling method. Section 5 describes the experimental methods used to evaluate the performance of the HS-CGK algorithm. Section 6 analyzes and discusses the experimental results. Finally, Section 7 summarizes the work presented in this paper.

2 RELATED WORK

Various methods have been proposed to address the class imbalance problem in machine learning, mainly focusing on data-level processing methods such as undersampling, oversampling, and hybrid sampling [12]. In this section, we review some of the most commonly used methods in each category.

2.1 Undersampling

The undersampling method balances the dataset by removing some of the majority class samples [13]. Random undersampling is the most common type of undersampling algorithm, which randomly selects some majority class samples and combines

them with the original minority class samples to form a balanced dataset [14]. However, this approach results in the loss of useful information in the majority class samples. To address this problem, Vuttipittayamongkol and Elyan [8] proposed a neighborhood-based undersampling framework, which identifies and eliminates negative samples in overlapping regions by introducing four different knn-based methods. Feng et al. [15] designed a new clustering undersampling method by improving the SBC algorithm, which achieves undersampling by selecting a different number of majority class instances from different clusters. Zheng et al. [16] proposed a three-stage undersampling framework that removes noisy and unrepresentative samples from the majority class, resulting in improved classification performance for imbalanced data. However, on datasets with severe imbalance problems, this method may delete too many majority class samples, resulting in a lack of data problem that affects the generalization ability of the classifier.

2.2 Oversampling

The oversampling method balances the dataset by generating a sufficient number of minority class samples [17]. Random oversampling is the most commonly used oversampling method, which randomly selects some minority class samples for replication and then combines them with the original majority class samples to form a balanced dataset. However, this method is prone to overfitting problems [15]. To address this, Chawla et al. [18] proposed the synthetic minority oversampling technique (SMOTE) algorithm, which synthesizes new minority class samples by interpolating adjacent minority class samples. SMOTE solves the overfitting problem of random oversampling, but it may lead to sample overlap and noise. To address these drawbacks, several SMOTE-based variants have emerged. For instance, Han et al. [19] proposed the Borderline-SMOTE algorithm, which synthesizes new samples by following the boundary lines of a few classes of samples. Arafa et al. [20] proposed a Reduced Noise-SMOTE (RN-SMOTE) that introduces noisy oversampled synthetic samples in a minority class using SMOTE, then applies the DBSCAN algorithm for noise detection and removal, and finally uses SMOTE again to balance the dataset. Li et al. [21] proposed the filter-based oversampling algorithm SMOTE-NaN-DE, which generates a minority class of samples using the SMOTE algorithm and then detects boundary and noise samples based on the natural neighbor error detection technique, using a differential evolutionary algorithm to adjust their location. Although the above methods improve the boundary and noise problems of imbalanced datasets, the generated new samples do not always restore the distribution properties of the original samples.

2.3 Hybrid Sampling

The hybrid sampling method combines the advantages of both undersampling and oversampling methods, and the basic idea is to increase the minority class samples while removing the majority class samples to obtain a balanced data set [22]. Some

researchers have proposed hybrid sampling algorithms that can solve the class imbalance and class overlap problems. For instance, Xu et al. [23] proposed a hybrid sampling algorithm RFMSE for solving the imbalanced data classification problem in medical diagnosis. The algorithm uses Misclassification-oriented Synthetic minority over-sampling technique (M-SMOTE) algorithm to add minority class samples and Edited nearest neighbor (ENN) to remove noisy data in majority classes. Additionally, Wang et al. [24] proposed a hybrid sampling algorithm ESMOTE + SSLM, which selects a minority class sample close to the classification boundary for oversampling by ESMOTE, while SSLM is used to remove the majority and minority class instances that exceed the learning boundary. However, these methods are not adapted to the generation of high-dimensional data.

2.4 GAN

Recently, the GAN has shown great potential in generating synthetic samples that match the distribution of real data [25]. The GAN-based oversampling method provides a new way of thinking to solve the class imbalance problem. For example, Mottini et al. [26] proposed a method for generating Passenger Name Records (PNR) that match the distribution of real data using GAN. Similarly, Rath et al. [27] proposed an integrated model based on long short-term memory (LSTM) and GAN to solve the class imbalance problem in disease prediction data, where GAN is used to generate samples that match the distribution of real data and LSTM model is used for disease prediction. Moreover, Engelmann and Lessmann [28] proposed an oversampling method based on conditional Wasserstein GAN to solve the class imbalance problem in credit card scoring. In addition, Lei et al. [29] proposed an imbalanced generative adversarial fusion network (IGAFN) to deal with the class imbalance problem in credit card transaction data. Compared with traditional methods, GAN-based oversampling can better learn the data distribution and generate synthetic samples that match the real data distribution, thus better solving the class imbalance problem. However, this method does not take into account the class overlap problem that exists in imbalanced data sets [11].

To address the class overlap problem in GAN-generated data, Zhu et al. [11] proposed a hybrid sampling algorithm based on GAN, which uses GAN to generate an initial balanced dataset and a novel adaptive neighborhood-based weighted undersampling method to remove the overlapping samples from the generated and original samples. This method can solve the class overlap problem in GAN-generated data. However, it does not consider noisy data in imbalanced datasets and is not applicable to the generation of tabular data.

In this paper, we devised an oversampling algorithm (DB-CTGAN) grounded in CTGAN. This algorithm incorporates the DBSCAN clustering technique to eliminate noisy and boundary samples, and subsequently leverages CTGAN to generate synthetic samples that adhere to the authentic data distribution. Subsequently, we blend the expanded fraudulent samples obtained through DB-CTGAN with normal samples, and employ the KNN overlap undersampling algorithm to eliminate

samples within the overlapping region, effectively resolving the issue of class overlap.

3 CONDITIONAL TABULAR GENERATIVE ADVERSARIAL NETWORK

CTGAN is a GAN-based generative model designed for modeling and sampling distributions of tabular data [30]. Before introducing CTGAN, let us first introduce GAN. GAN is a deep learning model for data synthesis that consists of two networks: generator and discriminator. The classical GAN model structure is shown in Figure 1.

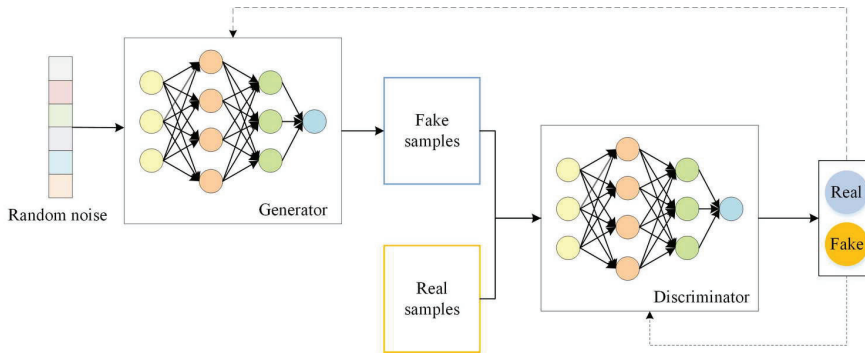


Figure 1. Structure of a typical GAN

Among them, the generator generates synthetic samples that are as close as possible to the real data distribution by learning the distribution pattern of the input samples, and the discriminator distinguishes the generated samples from the original ones. During the training process, the two networks play competitively with each other to finally reach a balance and generate new samples that match the real data distribution [9]. The loss function of the GAN is defined as:

$$\min_G \max_D V(G,D) = E_{x\sim P_r}\{\log[D(x)]\} + E_{z\sim P_z}\{\log[1-D(G(z))]\}, \tag{1}$$

where x represents real sampling, P_r represents the real sampling distribution, z represents random noise, P_z represents random noise distribution, $G(z)$ represents fake sample data generated by generator G, and $D(\cdot)$ represents the output value of the discriminator D.

GAN is mainly used for data generation in the context of unstructured data (e.g., images) and is not applicable to the expansion of tabular data [28]. CTGAN is an improvement on the GAN architecture to solve the problem of synthesizing tabular data. CTGAN uses mode-specific normalization to overcome the non-Gaussian and multimodal distribution of continuous columns. Specifically, CTGAN uses the

variational Gaussian mixture model (VGM) [31] to estimate the number of modes in each continuous column and to fit a Gaussian mixture. For each value in a continuous column, the probability that it comes from each mode is also calculated. Then, a random pattern is selected from the given patterns and normalized using that pattern. After normalization, the importance of the sample in its chosen Gaussian distribution is denoted by α [32], as shown in Equation (2):

$$\alpha_{i,j} = \frac{c_{i,j} - \eta_n}{4\Phi_n}, \tag{2}$$

where $\alpha_{i,j}$ is the normalized value of column i and row j, $c_{i,j}$ is the value of column i and row j, η_n is the n^{th} mode in column i, and Φ_n is the standard deviation of the n^{th} Gaussian distribution in column i.

CTGAN uses a conditional generator and training by sample to deal with the imbalance of classification columns. First, CTGAN encodes each column and categorical variable in the tabular data into conditional vectors. These conditional vectors are sampled according to the log frequency of the categories to ensure that rare categories are sampled uniformly. Then, the conditional vector is used as the input of the generator [33], which enables the generated samples to cover the entire category space, thus better solving the imbalance of classification columns.

In addition, CTGAN incorporates the latest GAN training advances to improve the stability of training and the quality of generated samples. By adopting WGAN-GP [34] as the loss function of GAN and introducing the gradient penalty technique, CTGAN can better avoid the vanishing gradient problem during training and generate more realistic samples. The WGAN-GP loss function is shown in Equation (3):

$$L = E_{G(z)\sim P_g}[D(G(z))] - E_{x\sim P_r}[D(x)] + \lambda E_{y\sim P_y}\left[(||\nabla_y D(y)|| - 1)^2\right], \tag{3}$$

where P_r and P_g represent the distributions of the real and generated data, λ represents the multiplicative gradient coefficient, and y is the sample linearly interpolated to the real data x.

To overcome the mode collapse problem, the critic structure of CTGAN uses the PacGAN [35] framework with the generator network structure $G(z, cond)$ [30] as:

$$\begin{cases} h_0 = z \oplus cond, \\ h_1 = h_0 \oplus \mathrm{Re}LU(BN(FN_{|cond|+|z|\to 256}(h_0))), \\ h_2 = h1 \oplus \mathrm{Re}LU(BN(FN_{|cond|+|z|+256\to 256}(h_1))), \\ \hat{\alpha}_i = \tanh(FC_{|cond|+|z|+512\to 1}(h_2)), \\ \hat{\beta}_i = gumbel_{0.2}(FC_{|cond|+|z|+512\to m_i}(h_2)), \\ \hat{d}_i = gumbel_{0.2}(FC_{|cond|+|z|+512\to |D_i|}(h_2)), \end{cases} \tag{4}$$

where z represents the random noise and *cond* represents the conditional probability. In the generator, CTGAN uses a batch normalization and relu activation function to generate synthetic row representations after two hidden layers using a hybrid

activation function. The scalar value $\hat{\alpha}_i$ is generated by tanh, and the pattern indicator $\hat{\beta}_i$ and the discrete value $\hat{d}_i$ are generated by gumbel softmax.

The critic (pac size of 10) network structure $C(r1, \ldots, r10, cond1, \ldots, cond10)$ [30] is given by:

$$\begin{cases} h_0 = r_1 \oplus \cdots \oplus r_{10} \oplus cond_1 \oplus \cdots \oplus cond_{10}, \\ h_1 = drop(leaky_{0.2}(FC_{10|r|+10|cond|\to 256}(h_0))), \\ h_2 = drop(leaky_{0.2}(FC_{256\to 256}(h_1))), \\ C(.) = FC_{256\to 1}(h_2), \end{cases} \tag{5}$$

here, r_i represents an example. CTGAN uses the leaky ReLU function and dropout on each hidden layer in the discriminator.

Credit card data is a type of high-dimensional tabular data that contains both data and classification information. Based on the above concept, CTGAN can effectively learn the distribution of credit card data and generate synthetic samples that match the real data distribution.

4 METHODS

The detection of credit card fraud transactions requires building a model and assigning a class label to each user based on the attributes of the credit card user. Suppose we have a dataset T with n transactions:

$$T = x_1, x_2, \ldots, x_n, y_i, \tag{6}$$

here, $y_i \in 0, 1$ and x_i represents the feature vector of each user. y_i represents the class label of each user. We determine the category y_i of a user by building a detection model D to learn the attributes x_i of the user. User labels are classified as normal and fraudulent, represented by 0 (normal) and 1 (fraudulent). The class imbalance problem refers to the fact that one class of labels in y_i is much more than another class of labels. The class imbalance problem can affect the performance of the classification algorithm.

CTGAN is a generative adversarial network for synthetic tabular data. It is able to learn the distribution of complex data in a data-driven manner without relying on any a priori assumptions, generating synthetic samples similar to the original data. Based on CTGAN and KNN, we propose a hybrid sampling method for imbalanced datasets, HS-CGK. Its structure is shown in Figure 2.

HS-CGK is a two-stage approach that employs DB-CTGAN, a CTGAN-based oversampling technique, to tackle the challenge of class imbalance, along with KNN overlapping undersampling to mitigate the presence of overlapping samples. Precisely, DB-CTGAN is utilized to learn the distribution of the original samples belonging to the minority class and generate synthetic samples that conform to the real data distribution, thereby augmenting the minority class samples. Subsequently,

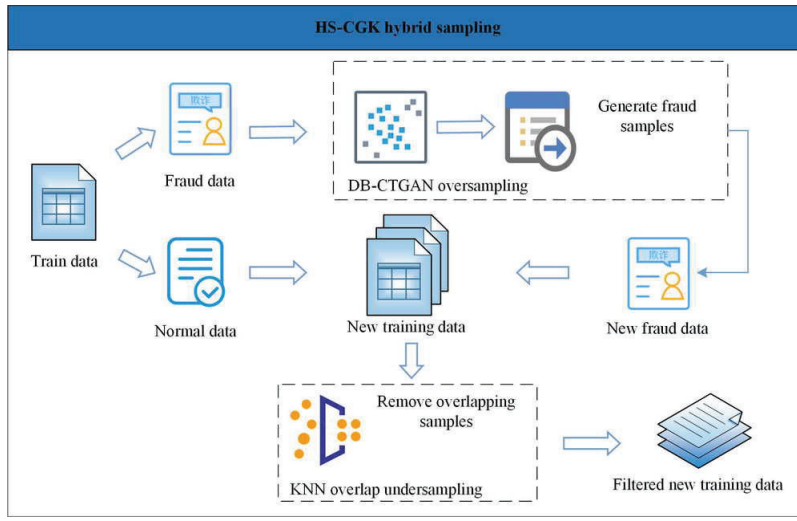


Figure 2. HS-CGK structure diagram

the augmented minority class samples are integrated with the original majority class samples, and the KNN overlap undersampling algorithm is employed to refine the resulting training set by removing majority class samples located in the overlapping region, yielding a training set characterized by distinct classification boundaries. The DB-CTGAN oversampling algorithm executes the following specific steps to address the imbalance in the original dataset: it leverages the DBSCAN clustering algorithm to filter the original data and eliminate noise as well as boundary samples from the minority class. It then employs the CTGAN algorithm to expand the minority class samples, employing a conditional generator that generates new samples adhering to the actual sample distribution. The conditional generator incorporates batch normalization and ReLU activation functions. Its working process is depicted below.

First, the conditional generator learns the conditional distribution of the real data:

$$P_g(row \mid D_{i*} = k^*) = P(D_{i*} = k), \tag{7}$$

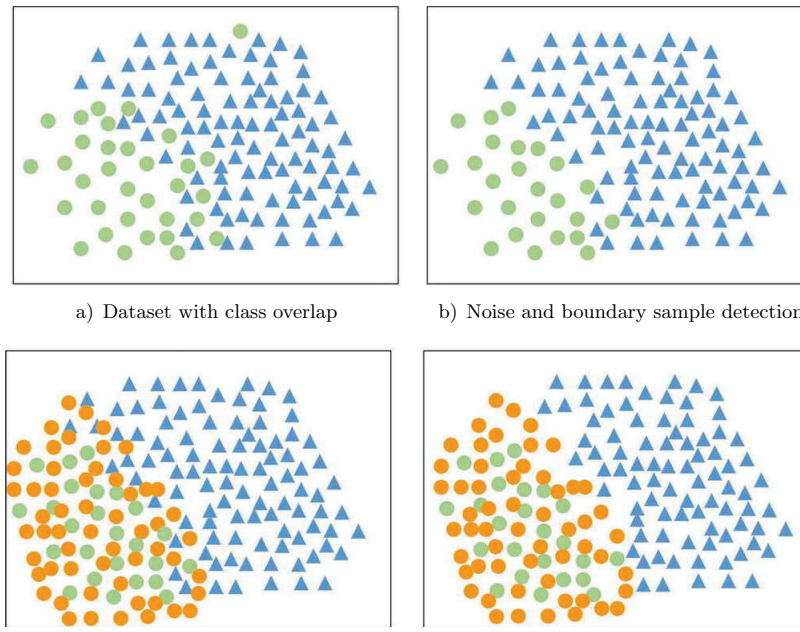
where k^* represents the value from the i^{th} discrete column $D_i{}^*$.

Next, the conditional generator constructs the original data distribution based on the learned real data distribution:

$$P(row) = \sum_{k \in D_{i*}} P_g(row \mid D_{i*} = k^*) P(D_{i*} = k). \tag{8}$$

To improve training stability and the quality of generated data, CTGAN adopts the discriminator structure of PacGAN and the loss function of WGAN-GP to address the issues of mode collapse and gradient vanishing in GANs.

The complete process of data processing by HS-CGK is illustrated in Figure 3.



a) Dataset with class overlap b) Noise and boundary sample detection

c) Minority class sample generation d) Removal of overlapping samples

Figure 3. Sample example

Figure 3 a) displays the unprocessed dataset, revealing the overlap of minority class samples with majority class samples and the presence of noise. Figure 3 b) shows the dataset after removing the noise and some boundary samples. It is evident that filtering the boundary and noise samples in the minority class samples using the DBSCAN clustering algorithm allows for more representative samples to be extracted. Figure 3 c) shows the dataset after CTGAN generates minority class samples. It can be observed that the CTGAN-based oversampling exacerbates the class overlap. Figure 3 d) depicts the dataset after removing the overlapping samples. It is apparent that KNN overlap undersampling can eliminate most of the class samples at the overlap boundary and make the classification boundary clearer.

The structure of the DB-CTGAN model, proposed in the first phase of HS-CGK, is illustrated in Figure 4.

The specific process of generating fraud samples using this model is outlined below:

Step 1: The original dataset is divided into a training set T_{train} and a test set T_{test}. T_{train} is used to train the DB-CTGAN model and the classifier, while T_{test} is kept for evaluating the results.

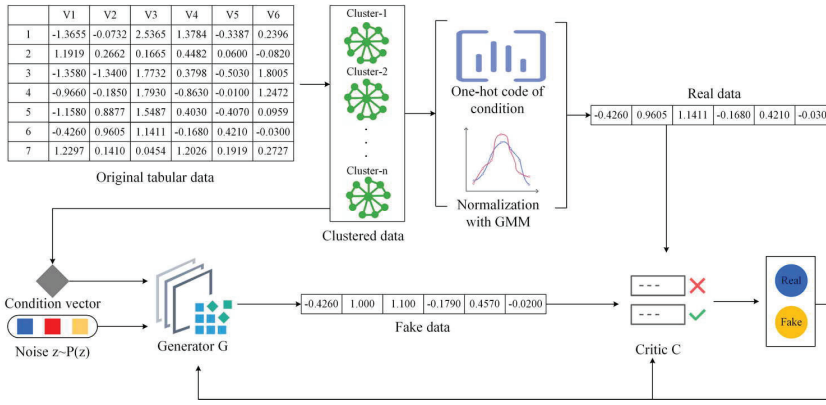


Figure 4. DB-CTGAN structure diagram

Step 2: The data in T_{train} are preprocessed, and the preprocessed data are further divided into normal samples T_{normal} and fraudulent samples T_{fraud}. T_{normal} is used to undersample the majority class samples, and T_{fraud} is used to oversample the minority class samples.

Step 3: The minority class samples are partitioned into core points, boundary points, and noise points using the DBSCAN clustering algorithm. Some boundary and noise samples are filtered.

Step 4: The generator produces a set of fake samples based on the input condition and random noise z. Then, the generated fake samples and the clustered and normalized real samples are input to the critic C simultaneously. The parameters of the critic are updated based on the error between them, to make the critic more accurate for the input data.

Step 5: After training the critic, the parameters of C are fixed. When the generator generates samples again, the samples are input to C, and the error is back-propagated to the generator to update the generator's parameters. This step aims to generate samples closer to the real minority class samples.

Step 6: After n iterations of the DB-CTGAN network, the convergence trend of the generator and critic's loss function determines whether the DB-CTGAN network has finished training.

Step 7: The trained DB-CTGAN model generates fraudulent samples T_{new} with similar distributions to the real samples.

Step 8: Mix the synthesized new sample T_{new} with the original minority class sample T_{fraud} to get the new minority class sample set T'_{fraud}.

Step 9: Mix the extended fraud sample T'_{fraud} with the normal sample T_{normal} to balance the dataset and obtain the new balanced training set T'_{train}.

After oversampling with DB-CTGAN, the issue of class overlap persists. To address this problem, the second stage of HS-CGK employs the KNN overlap undersampling algorithm, which removes most samples from the overlap boundary. KNN is a supervised classification algorithm that determines whether a sample belongs to a class based on whether the majority of the K most similar samples in the feature space also belong to that class [36]. The KNN overlap undersampling algorithm is presented in Algorithm 1.

Algorithm 1 KNN overlap undersampling

Require: The training set after CTGAN processing: T_{train}, number of nearest neighbors: K, minimum count point for a minority class t $(1 \leq t \leq K)$

Ensure: The training set after removing overlaps: T'_{train}

1: Split the training set T_{train} into normal transaction samples T_{normal} and fraudulent transaction samples T_{fraud}
2: **for** each sample $x \in T_{train}$ **do**
3: Define x as the unknown sample and all samples except for x as the known samples y
4: Calculate the distance d between x and all known samples y as $d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
5: Sort the distances between x and all known samples y in increasing order
6: Find the nearest K sample points N_k that are at the distance between x and the unknown sample, $N_k = x_1, x_2, \ldots, x_K$
7: **if** $x_i \in T_{fraud}$ **then**
8: Count the number of minority class samples in N_k as m: $m = m + 1$
9: **end if**
10: **if** $m \geq t$ **then**
11: Remove x from the training set T_{train}
12: **end if**
13: **end for**
14: Output the training set after removing overlaps: T'_{train}

In Algorithm 1, the DB-CTGAN-balanced training set is first divided into normal transaction samples and fraudulent transaction samples (line 1). Next, each sample is considered as an unknown sample and its distance to all known samples is calculated (lines 2–4). Then, the K sample points with the smallest distance from the unknown sample are identified, and the number of minority class samples among the K nearest neighbors is counted (lines 5–9). If this number is greater than or equal to a defined minimum count t belonging to a minority class, the point is removed (lines 10–13). Finally, the algorithm outputs the training set T'_{train} after removing the overlap (line 14).

It is important to note that if all K nearest neighbors of an unknown sample are majority class samples, the point will not be removed. Similarly, if the number of minority class samples among the K nearest neighbors of the point is less than t, the

point will not be removed. Therefore, the KNN overlap undersampling algorithm only operates on the class overlap region.

5 EXPERIMENTAL SETTINGS

We conducted experiments to evaluate the effectiveness of HS-CGK in handling imbalanced data, and compared it with traditional sampling methods and the latest sampling methods in combination with various classification algorithms. We address the following research questions:

RQ1: Is the performance of HS-CGK optimal compared to several classical methods and current popular methods?

RQ2: How robust is HS-CGK for different classification algorithms?

RQ3: Does HS-CGK enhance the performance of different classification algorithms?

RQ4: Which classification algorithm in combination with HS-CGK is most suitable for solving credit card fraud transaction detection?

5.1 Experimental Data and Evaluation Metrics

To evaluate the performance of HS-CGK in processing imbalanced datasets, we selected four credit card imbalance datasets from UCI and a real credit card dataset used in [37]. These datasets have varying numbers of features and imbalance rates, and their detailed descriptions are provided in Table 1.

Dataset	Instances	Features	Source	Minority	Majority	IR
Australian	680	16	UCI	307	383	1.24
German	1 000	20	UCI	300	700	2.33
Taiwan	30 000	25	UCI	6 636	23 364	3.52
Bank	4 521	16	UCI	521	4 000	7.67
Europe	284 807	31	[37]	492	284 315	577.87

Table 1. Datasets Description

Table 1 provides a summary of each dataset, including its abbreviation, total number of samples (instances), number of features, source, minority class, majority class, and imbalance ratio (IR). For example, in the German dataset, there are 1 000 transaction records, each containing 20 features, where 300 and 700 represent the number of fraudulent and normal transactions, respectively. The fraud rate is therefore 30 %, and the IR is 2.33. The Taiwan dataset consists of 30 000 credit card customer records from April 2005 to September 2005. Among these, 6 636 are late credit card payments and 23 364 are normal credit card payments, resulting in a delinquency rate of 22 % and an IR of 3.52. Both real datasets are typical examples of imbalanced datasets.

We used five performance evaluation metrics: accuracy, recall, area under the curve (AUC) based on the receiver operating characteristic (ROC), F1, and G-mean. These metrics are commonly used to evaluate algorithms dealing with imbalanced datasets [11, 38].

5.2 Classifiers and Baseline Approaches

We used four different classifiers: RF, SVC, DT, and XGBoost to evaluate the performance of HS-CGK, classical sampling methods, and the latest sampling methods. All classification algorithms are implemented in the scikit-learn library on Python, and their default parameters remain unchanged.

We compared the performance of original data, HS-CGK, and 8 other sampling algorithms on different classifiers. All sampling algorithms are presented in Table 2.

Sampling Algorithms	Detailed Information
NearMiss (NN)	The positive sample with the smallest average distance from the N nearest negative samples is selected.
SMOTE	By randomly generating new samples on the concatenation between a minority of neighboring classes of samples.
ADASYN	The number of synthetic samples generated is determined automatically based on the distribution of each sample in the sample space.
SMOTETomek (ST)	SMOTETomek is a hybrid method based on the combination of Tomek link and SMOTE.
GAN	The generator generates the data, the discriminator discriminates the generated data from the original data, and the two play against each other to obtain a new sample that matches the real data distribution.
GAN + Tomek (GT)	GAN generates the data and Tomek link removes the overlapping data.
CTGAN	CTGAN is a tabular data generation model, details of which are in Section 3.
CTGAN + Tomek (CTGANT)	CTGAN generates the data and Tomek link removes the overlapping data.
HS-CGK	The method proposed in this paper.

Table 2. Sampling methods in the experiments

6 EXPERIMENTAL RESULTS DISCUSSION

In this section, we present a comparison of the performance of HS-CGK with several classical and latest sampling methods and provide insights on its competitiveness

and robustness across different classification models. This comparison allows us to answer the research questions posed in Section 5.

6.1 Comparison of Classification Performance

The aim of the first module of our experimental study is to evaluate the performance of HS-CGK compared to other methods (RQ1) and to analyze its robustness on different classification models (RQ2). To achieve this, we measured the F1, AUC, and G-mean metrics of the original dataset, the dataset processed by 8 sampling methods, and the dataset sampled by HS-CGK on 4 different classifiers. Tables 3, 4 and 5 present the performance metrics for each dataset and classifier combination.

Table 3 shows that HS-CGK ranked first in 11 out of 20 scenarios analyzed for the F1 metric. Notably, HS-CGK ranked first in F1 for all four classifiers in the German dataset. Although the F1 of HS-CGK on the SVC classifier was slightly worse than the SMOTE method, it outperformed other sampling methods on the RF, DT, and XGBoost classifiers. Meanwhile, Table 4 indicates that HS-CGK achieved the best results in 15 out of 20 scenarios analyzed for the AUC metric, particularly on the German and Bank datasets. Moreover, HS-CGK outperformed other sampling methods on all four classifiers, including RF, DT, SVC, and XGBoost, as shown in the experimental results. Table 5 further reveals that HS-CGK achieved the best results in 15 out of 20 scenarios analyzed for the G-mean metric and showed a more balanced performance across the four classifiers. In summary, HS-CGK demonstrated optimal results in F1, AUC, and G-mean metrics, surpassing both oversampling, undersampling, and hybrid sampling methods.

Furthermore, Tables 3, 4 and 5 reveal that HS-CGK performed better on RF, DT, and XGBoost classifiers than other sampling methods. On the SVC classifier, HS-CGK outperformed other sampling methods and achieved one best F1, three best AUCs, and three best G-means. Notably, in the Europe dataset, the F1 of HS-CGK was 0.7902, which is 6 % better than the untreated dataset, while the F1 of the SMOTE method was only 0.1558, much lower than the untreated dataset. This result indicates that HS-CGK has stronger stability and robustness than other methods. Thus, HS-CGK outperforms both traditional and latest sampling methods on class-imbalanced datasets and shows robustness across classifiers of different nature.

6.2 Comparison of Experimental Results with the Original Data Set

The purpose of the second module in this experimental study is to compare the performance of the HS-CGK processed dataset with the original dataset using different classifiers to evaluate whether it can improve the performance of various classification algorithms (RQ3). The Australian dataset and the Taiwan dataset were selected for the experiments, and the accuracy, recall, F1, and AUC of the original dataset

Classifier	Dataset	NONE	NN	SMOTE	ADASYN	ST	GAN	GT	CTGAN	CTGANT	HS-CGK
RF	Australian	0.8275	0.8314	0.8342	0.8268	0.8361	0.8172	0.8156	**0.8409**	0.8187	**0.8409**
	German	0.4895	0.5785	0.5477	0.5394	0.5660	0.5290	0.4507	0.5901	0.5939	**0.6044**
	Taiwan	0.4426	0.3567	0.4769	0.4709	0.4811	0.4478	0.4033	0.4599	0.4749	**0.6069**
	Bank	0.2777	0.3275	0.4467	0.3939	0.4166	0.2910	0.2346	0.4554	0.4071	**0.5028**
	Europe	0.8212	0.0034	0.8469	0.8253	**0.8471**	0.8120	0.8075	0.8384	0.8413	0.8430
DT	Australian	0.7513	0.7796	**0.8111**	0.7849	0.8087	0.7500	0.7630	0.7634	0.7560	0.8042
	German	0.2459	0.5000	0.4946	0.4736	0.5000	0.5326	0.4808	0.5047	0.4623	**0.5676**
	Taiwan	0.3694	0.3483	0.3949	0.3868	0.3872	0.3821	0.3721	0.3824	0.4029	**0.4183**
	Bank	0.4273	0.2843	0.4352	0.4882	0.4396	0.3859	0.4250	0.4180	0.3595	**0.4975**
	Europe	**0.7769**	0.0038	0.4693	0.4920	0.4749	0.3212	0.4069	0.6149	0.6400	0.6491
SVC	Australian	0.8152	0.8359	0.8216	0.8297	0.8216	0.8465	**0.8527**	0.8191	0.8135	0.8481
	German	0.6027	0.5454	0.5951	0.5919	0.5981	0.5919	0.5849	0.6031	0.6077	**0.6200**
	Taiwan	0.4352	0.3629	**0.5079**	0.4957	0.5044	0.4537	0.4393	0.4636	0.4595	0.5027
	Bank	0.2125	0.2871	0.5144	0.4638	**0.5323**	0.1761	0.1513	0.5250	0.4982	0.5000
	Europe	0.7310	0.0597	0.1558	0.0653	0.1435	0.3326	0.3319	**0.8013**	**0.8013**	0.7902
XGBoost	Australian	0.8314	**0.8351**	0.8066	0.8152	0.8333	0.8156	0.8156	0.8268	0.8268	0.8342
	German	0.5294	0.5462	0.5357	0.5442	0.5465	0.5795	0.5795	0.5393	0.5393	**0.5871**
	Taiwan	0.4454	0.3576	0.4772	0.4731	0.4864	0.4507	0.4507	0.4684	0.4684	**0.5871**
	Bank	0.4285	0.3597	0.4718	0.4966	0.5238	0.4140	0.4140	0.4686	0.4686	**0.5919**
	Europe	**0.8409**	0.0037	0.4761	0.3700	0.4785	0.8102	0.8102	0.8362	0.8373	0.8375

Table 3. F1 of real data sets on different classifiers

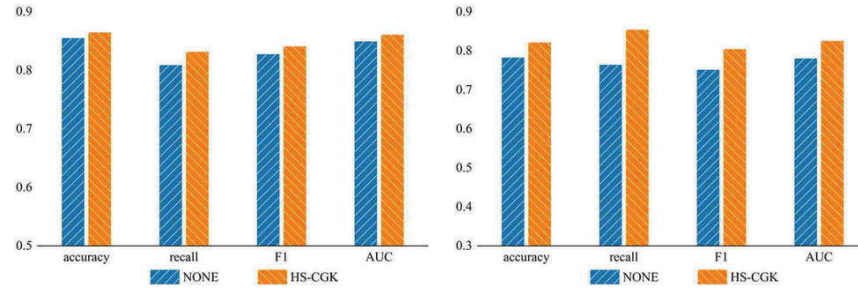
Classifier	Dataset	NONE	NN	SMOTE	ADASYN	ST	GAN	GT	CTGAN	CTGANT	HS-CGK
RF	Australian	0.8494	0.8521	0.8550	0.8479	0.8564	0.8347	0.8358	**0.8606**	0.8424	**0.8606**
	German	0.6515	0.7021	0.6845	0.6799	0.6962	0.6706	0.6301	0.7134	0.7147	**0.7284**
	Taiwan	0.6421	0.5391	0.6657	0.6628	0.6682	0.6447	0.6239	0.6552	0.6676	**0.7284**
	Bank	0.5818	0.6898	0.6698	0.6367	0.6540	0.5885	0.5666	0.6784	0.6471	**0.7218**
	Europe	0.8648	0.5048	0.9046	0.9011	0.9103	0.8648	0.8614	0.9148	0.9148	**0.9227**
DT	Australian	0.7803	0.8071	**0.8338**	0.8084	0.8309	0.7804	0.7934	0.7887	0.7932	0.8252
	German	0.5428	0.6282	0.6412	0.6471	0.6650	0.6277	0.6422	0.6168	0.4623	**0.6956**
	Taiwan	0.5936	0.5283	0.6108	0.6048	0.6059	0.6015	0.6024	0.5998	0.6136	**0.6266**
	Bank	0.6743	0.6257	0.6873	0.7106	0.6934	0.6454	0.6663	0.6740	0.6390	**0.7478**
	Europe	0.8647	0.5547	0.8762	0.8815	0.8937	0.8557	0.8531	0.9040	0.9075	**0.9191**
SVC	Australian	0.8366	0.8548	0.8422	0.8492	0.8422	0.8608	0.8652	0.8393	0.8366	**0.8660**
	German	0.7158	0.6698	0.7191	0.7178	0.7225	0.7071	0.7015	0.7239	0.7263	**0.7294**
	Taiwan	0.6386	0.5582	**0.6898**	0.6870	0.6887	0.6475	0.6405	0.6597	0.6599	0.6828
	Bank	0.5583	0.6399	0.7606	0.7656	0.7739	0.5461	0.5386	0.7225	0.6945	**0.7764**
	Europe	0.7939	0.9086	**0.9273**	0.8669	0.9233	0.7620	0.7620	0.8977	0.8977	0.9010
XGBoost	Australian	0.8521	**0.8549**	0.8296	0.8366	0.8535	0.8358	0.8358	0.8479	0.8479	0.8534
	German	0.6666	0.6740	0.6751	0.6809	0.6821	0.7000	0.7000	0.6762	0.6762	**0.7132**
	Taiwan	0.6434	0.5427	0.6619	0.6593	0.6672	0.6460	0.6460	0.6594	0.6594	**0.7132**
	Bank	0.6530	0.7168	0.6795	0.6993	0.7095	0.6503	0.6503	0.6852	0.6852	**0.8334**
	Europe	0.8749	0.5361	0.9067	**0.9174**	0.9121	0.8749	0.8749	0.9080	0.9114	0.9134

Table 4. AUC of real data sets on different classifiers

Classifier	Dataset	NONE	NN	SMOTE	ADASYN	ST	GAN	GT	CTGAN	CTGANT	HS-CGK
RF	Australian	0.8484	0.8519	0.8543	0.8477	0.8560	0.8347	0.8347	**0.8601**	0.8405	**0.8601**
	German	0.5962	0.7016	0.6592	0.6488	0.6743	0.6352	0.5617	0.7083	0.6999	**0.7206**
	Taiwan	0.5682	0.5010	0.6337	0.6350	0.6369	0.5729	0.5272	0.6212	0.6499	**0.7206**
	Bank	0.4204	0.6722	0.6082	0.5473	0.5838	0.4409	0.3812	0.6244	0.5703	**0.6933**
	Europe	0.8542	0.1829	0.8996	0.8956	0.9059	0.8541	0.8502	0.9108	0.9108	**0.9194**
DT	Australian	0.7801	0.8065	**0.8337**	0.8083	0.8309	0.7778	0.7898	0.7886	0.7873	0.8247
	German	0.3913	0.6270	0.6323	0.6156	0.6351	0.6540	0.6122	0.6417	0.6056	**0.6930**
	Taiwan	0.5543	0.4929	0.5905	0.5839	0.5815	0.5666	0.5352	0.5841	0.6084	**0.6226**
	Bank	0.6289	0.6178	0.6531	0.6778	0.6632	0.5751	0.6068	0.6332	0.5856	**0.7362**
	Europe	0.8541	0.3948	0.8678	0.8738	0.8877	0.8442	0.8409	0.8990	0.9029	**0.9156**
SVC	Australian	0.8365	0.8542	0.8421	0.8487	0.8421	0.8608	0.8638	0.8389	0.8362	**0.8649**
	German	0.6789	0.6650	0.7190	0.7161	0.7222	0.7066	0.7014	0.7213	0.7211	**0.7268**
	Taiwan	0.5508	0.5487	0.6741	0.6811	**0.6753**	0.5725	0.5576	0.6401	0.6542	0.6580
	Bank	0.3601	0.6022	0.7511	0.7648	0.7661	0.3273	0.2976	0.6883	0.6426	**0.7738**
	Europe	0.7666	0.9075	**0.9256**	0.8616	0.9214	0.7249	0.7249	0.8919	0.8919	0.8956
XGBoost	Australian	0.8519	**0.8549**	0.8295	0.8365	0.8534	0.8347	0.8347	0.8477	0.8477	0.8531
	German	0.6454	0.6740	0.6578	0.6651	0.6685	0.6871	0.6871	0.6658	0.6658	**0.7125**
	Taiwan	0.5627	0.5127	0.6077	0.6002	0.6164	0.5670	0.5670	0.6203	0.6203	**0.7125**
	Bank	0.5738	0.7101	0.6199	0.6530	0.6646	0.5713	0.5713	0.6339	0.6339	**0.8323**
	Europe	0.8660	0.3075	0.9022	**0.9141**	0.9082	0.8659	0.8659	0.9034	0.9071	0.9093

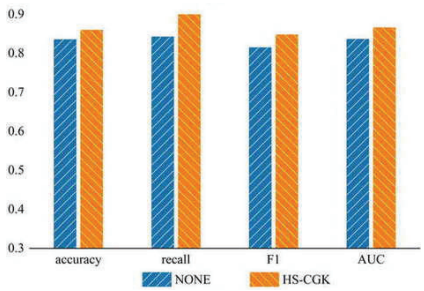
Table 5. G-mean of real data sets on different classifiers

and the dataset processed by HS-CGK on four different classifiers are presented in Figures 5 and 6.

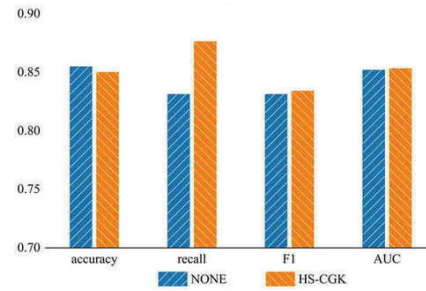


a) Random forests classifier

b) Decision tree classifier



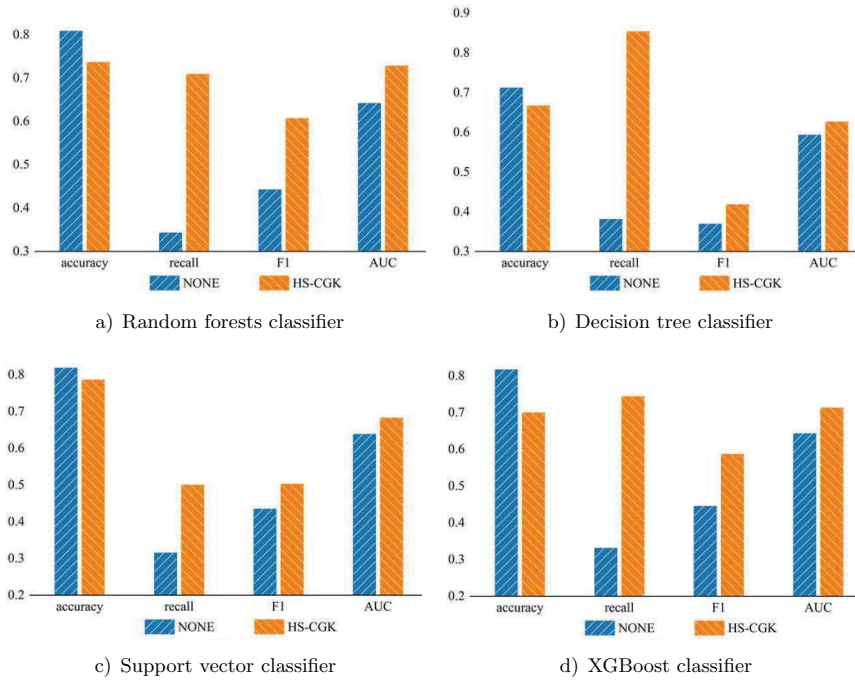
c) Support vector classifier



d) XGBoost classifier

Figure 5. NONE vs. HS-CGK for Australian dataset

As shown in Figures 5 a) and 6 a), the experimental results of the original dataset and the dataset processed by HS-CGK in the RF classifier indicate that the HS-CGK dataset outperforms the original data in terms of accuracy, recall, F1, and AUC for the Australian dataset, and for the Taiwan dataset, HS-CGK had a lower accuracy but significantly improved recall, F1, and AUC. Similar results were observed for DT and SVC classifiers, as shown in Figures 5 b), 6 b), 5 c), and 6 c). Figures 5 d) and 6 d) show the experimental results on XGBoost classifier for the original dataset and the dataset processed by HS-CGK, where HS-CGK outperforms the original data in terms of recall, F1, and AUC metrics, but has lower accuracy. It is important to emphasize that accuracy denotes the probability of correctly classifying a transaction among all transactions. Given that the credit card dataset is characterized by class imbalance, even if all transactions are classified as normal, the model's accuracy would remain high. Evaluation of classification performance in the presence of imbalanced data necessitates the consideration of diverse metrics such as accuracy, recall, F1, and AUC. In both datasets, HS-CGK exhibited superior performance compared to the original data across all four classifiers, as evidenced by higher recall, F1, and AUC values. HS-CGK leverages the DB-CTGAN over-



a) Random forests classifier b) Decision tree classifier

c) Support vector classifier d) XGBoost classifier

Figure 6. NONE vs. HS-CGK for Taiwan dataset

sampling algorithm to generate high-quality minority class samples, while the KNN overlap undersampling algorithm removes majority class samples in the overlapping region, resulting in a balanced dataset. This balanced dataset enables the classifier to treat both classes of samples more equitably, thereby enhancing classifier performance. In summary, HS-CGK effectively enhances the performance of different classifiers.

6.3 Experimental Results of HS-CGK on Different Classifiers

The purpose of the third module in this experimental study is to analyze the optimal fraudulent transaction detection algorithm in combination with HS-CGK (RQ4). The experimental results of accuracy, recall, F1, and AUC for five datasets processed by HS-CGK on four classifiers (RF, DT, SVC, and XGBoost) are shown in Figure 7.

Figure 7 a) shows that datasets 1, 2, and 4 have the highest accuracy on RF, while the first three datasets have the lowest accuracy on DT. Figure 7 b) shows that the recall of XGBoost is optimal since dataset 3 and 5 have the lowest recall on SVC, and datasets 1, 2, and 4 have the lowest recall on RF. Figure 7 c) shows that DT has the worst F1, while the F1 metrics of the remaining three clas-

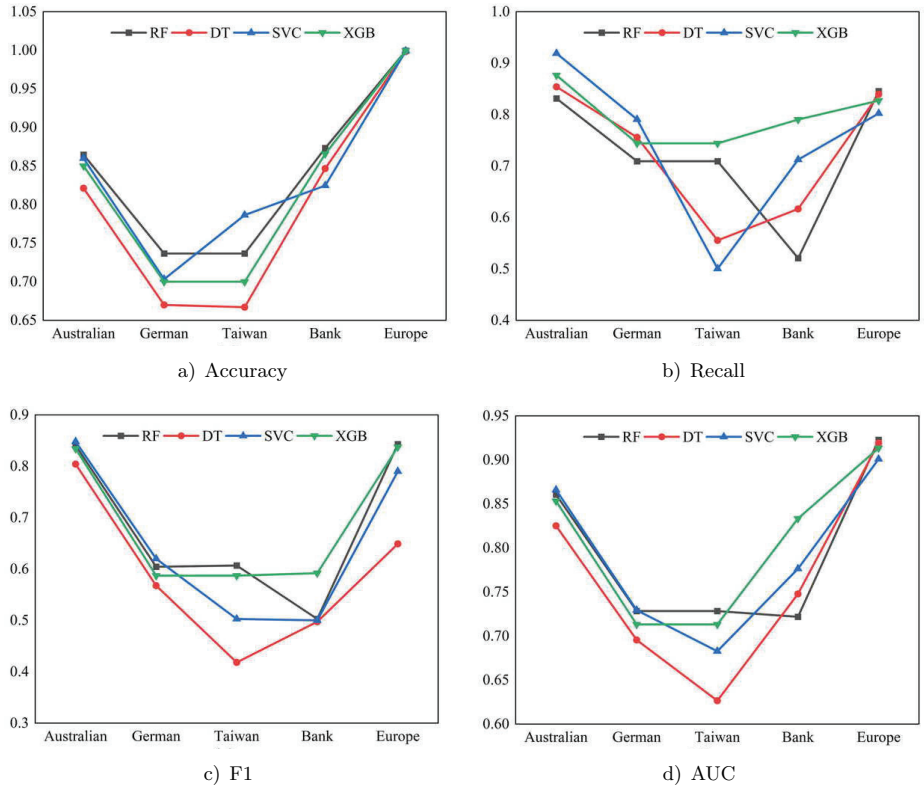


Figure 7. Experimental results of different datasets on RF/DT/SVC and XGBoost classifiers

sifiers are not significantly different. Figure 7 d) shows that XGBoost has the best AUC. The empirical findings from the analysis of five datasets reveal the superiority of XGBoost over the other three classifiers in terms of overall performance. Notably, credit card data is characterized by high dimensionality. XGBoost effectively mitigates overfitting issues associated with high-dimensional data by incorporating L1/L2 regularization penalties. This approach circumvents the problem of data overfitting, thereby enhancing the model's generalization performance. In summary, HS-CGK significantly enhances the classification performance of the four classifiers (RF/DT/SVC/XGBoost) upon achieving dataset balance. Moreover, the fusion of HS-CGK with XGBoost yields superior results compared to the fusion with RF, DT, and SVC classifiers.

6.4 Ablation Study

To conduct a comprehensive analysis of HS-CGK, an ablation experimental study was performed to assess the effectiveness of each module. The experimental data consisted of German dataset, and XGBoost was employed as the classifier. Detailed results of the ablation study are presented in Table 6.

Methods	Model			F1	AUC	G-mean
	KNN	CTGAN	DB			
Original	–	–	–	0.5294	0.6666	0.6454
KNN(O)	✓	–	–	0.5551	0.6827	0.6741
CTGAN(O)	–	✓	–	0.5393	0.6762	0.6658
DB-CTGAN	–	✓	✓	0.5685	0.6970	0.6955
CK	✓	✓	–	0.5806	0.7074	0.7069
HS-CGK	✓	✓	✓	**0.5871**	**0.7132**	**0.7125**

Note: KNN denotes the overlapping data processing module in HS-CGK; CTGAN denotes the data expansion module in HS-CGK; DB denotes the noise filtering module in HS-CGK.

Table 6. Results of the ablation study

In Table 6, the term "Original" denotes the classification results obtained using the original dataset. "KNN(O)" signifies the classification outcomes achieved by solely applying the KNN overlap undersampling module, while "CTGAN(O)" represents the classification results solely employing the CTGAN oversampling module. It is observed that the classification results of "KNN(O)" and "CTGAN(O)" outperform the "Original" results. This improvement can be attributed to the balancing effect induced by "KNN(O)" and "CTGAN(O)," which mitigates the bias arising from imbalanced data samples. Consequently, the classifier can treat both sample types fairly, thereby circumventing decision boundary bias. The entry labeled "DB-CTGAN" indicates the removal of the overlapping data processing module in HS-CGK. Experimental results demonstrate that the exclusion of the KNN module from HS-CGK diminishes its classification performance. This finding underscores the effectiveness of the KNN module, which removes the majority of class samples located in the overlapping region, thereby addressing the exacerbation of overlap phenomena in imbalanced datasets. Similarly, the "CK" entry denotes the elimination of the noise filtering module in HS-CGK, and the results reveal that the absence of DB adversely affects the classification performance of HS-CGK. The DB module filters out noisy samples from a few classes, improving the quality of CTGAN-generated samples and enhancing the classifier's classification performance. The experimental findings from "CK" affirm the effectiveness of the DB module. In conclusion, the modules incorporated in HS-CGK effectively enhance the classifier's classification performance.

7 CONCLUSION

In this paper, we proposed a novel hybrid sampling method, HS-CGK, to address the class imbalance and class overlap problems in credit card transaction datasets. HS-CGK combines the strengths of two existing techniques, CTGAN and KNN, to generate high-quality fraudulent transaction samples and effectively remove normal samples in the overlap region. Specifically, we introduced a DB-CTGAN approach to generate fraudulent transaction samples and used the DBSCAN clustering algorithm to filter noisy and boundary samples. The generated fraud samples were then mixed with the original dataset, and the KNN overlap undersampling algorithm was applied to remove normal samples in the overlap region. We evaluated HS-CGK on five real credit card datasets and compared it with eight other imbalanced data processing methods. The experimental results showed that HS-CGK outperformed all the other methods in terms of F1, AUC, and G-mean. Furthermore, we also tested four classifiers, RF, DT, SVC, and XGBoost on the processed datasets and found that HS-CGK significantly improved the classification performance of all the classifiers.

REFERENCES

[1] Carta, S.—Fenu, G.—Recupero, D. R.—Saia, R.: Fraud Detection for E-Commerce Transactions by Employing a Prudential Multiple Consensus Model. Journal of Information Security and Applications, Vol. 46, 2019, pp. 13–22, doi: 10.1016/j.jisa.2019.02.007.

[2] Arora, S.—Bindra, S.—Singh, S.—Nassa, V. K.: Prediction of Credit Card Defaults Through Data Analysis and Machine Learning Techniques. Materials Today: Proceedings, Vol. 51, 2022, No. 1, pp. 110–117, doi: 10.1016/j.matpr.2021.04.588.

[3] Alfaiz, N. S.—Fati, S. M.: Enhanced Credit Card Fraud Detection Model Using Machine Learning. Electronics, Vol. 11, 2022, No. 4, Art. No. 662, doi: 10.3390/electronics11040662.

[4] Forough, J.—Momtazi, S.: Sequential Credit Card Fraud Detection: A Joint Deep Neural Network and Probabilistic Graphical Model Approach. Expert Systems, Vol. 39, 2022, No. 1, Art. No. e12795, doi: 10.1111/exsy.12795.

[5] Rtayli, N.—Enneya, N.: Selection Features and Support Vector Machine for Credit Card Risk Identification. Procedia Manufacturing, Vol. 46, 2020, pp. 941–948, doi: 10.1016/j.promfg.2020.05.012.

[6] Jurgovsky, J.—Granitzer, M.—Ziegler, K.—Calabretto, S.—Portier, P. E.—He-Guelton, L.—Caelen, O.: Sequence Classification for Credit-Card Fraud Detection. Expert Systems with Applications, Vol. 100, 2018, pp. 234–245, doi: 10.1016/j.eswa.2018.01.037.

[7] Das, S.—Mullick, S. S.—Zelinka, I.: On Supervised Class-Imbalanced Learning: An Updated Perspective and Some Key Challenges. IEEE Transactions on Artificial Intelligence, Vol. 3, 2022, No. 6, pp. 973–993, doi: 10.1109/TAI.2022.3160658.

[8] Vuttipittayamongkol, P.—Elyan, E.: Neighbourhood-Based Undersampling Approach for Handling Imbalanced and Overlapped Data. Information Sciences, Vol. 509, 2020, pp. 47–70, doi: 10.1016/j.ins.2019.08.062.

[9] Goodfellow, I.—Pouget-Abadie, J.—Mirza, M.—Xu, B.—Warde-Farley, D.—Ozair, S.—Courville, A.—Bengio, Y.: Generative Adversarial Nets. In: Ghahramani, Z., Welling, M., C., C., Lawrence, N., Weinberger, K. Q. (Eds.): Advances in Neural Information Processing Systems 27 (NIPS 2014). Curran Associates, Inc., 2014, pp. 2672–2680, doi: 10.48550/arXiv.1406.2661.

[10] Esmaeilpour, M.—Chaalia, N.—Abusitta, A.—Devailly, F. X.—Maazoun, W.—Cardinal, P.: Bi-Discriminator GAN for Tabular Data Synthesis. Pattern Recognition Letters, Vol. 159, 2022, pp. 204–210, doi: 10.1016/j.patrec.2022.05.023.

[11] Zhu, B.—Pan, X.—vanden Broucke, S.—Xiao, J.: A GAN-Based Hybrid Sampling Method for Imbalanced Customer Classification. Information Sciences, Vol. 609, 2022, pp. 1397–1411, doi: 10.1016/j.ins.2022.07.145.

[12] Shi, S.—Li, J.—Zhu, D.—Yang, F.—Xu, Y.: A Hybrid Imbalanced Classification Model Based on Data Density. Information Sciences, Vol. 624, 2023, pp. 50–67, doi: 10.1016/j.ins.2022.12.046.

[13] Li, J.—Wu, Y.—Fong, S.—Tallón-Ballesteros, A. J.—Yang, X. S.—Mohammed, S.—Wu, F.: A Binary PSO-Based Ensemble Under-Sampling Model for Rebalancing Imbalanced Training Data. The Journal of Supercomputing, Vol. 78, 2022, No. 5, pp. 7428–7463, doi: 10.1007/s11227-021-04177-6.

[14] Zhang, Z. L.—Peng, R. R.—Ruan, Y. P.—Wu, J.—Luo, X. G.: ESMOTE: An Overproduce-and-Choose Synthetic Examples Generation Strategy Based on Evolutionary Computation. Neural Computing and Applications, Vol. 35, 2023, No. 9, pp. 6891–6977, doi: 10.1007/s00521-022-08004-8.

[15] Feng, S.—Zhao, C.—Fu, P.: A Cluster-Based Hybrid Sampling Approach for Imbalanced Data Classification. Review of Scientific Instruments, Vol. 91, 2020, No. 5, Art. No. 055101, doi: 10.1063/5.0008935.

[16] Zheng, M.—Li, T.—Zheng, X.—Yu, Q.—Chen, C.—Zhou, D.—Lv, C.—Yang, W.: UFFDFR: Undersampling Framework with Denoising, Fuzzy C-Means Clustering, and Representative Sample Selection for Imbalanced Data Classification. Information Sciences, Vol. 576, 2021, pp. 658–680, doi: 10.1016/j.ins.2021.07.053.

[17] Ding, H.—Chen, L.—Dong, L.—Fu, Z.—Cui, X.: Imbalanced Data Classification: A KNN and Generative Adversarial Networks-Based Hybrid Approach for Intrusion Detection. Future Generation Computer Systems, Vol. 131, 2022, pp. 240–254, doi: 10.1016/j.future.2022.01.026.

[18] Chawla, N. V.—Bowyer, K. W.—Hall, L. O.—Kegelmeyer, W. P.: SMOTE: Synthetic Minority over-Sampling Technique. Journal of Artificial Intelligence Research, Vol. 16, 2002, pp. 321–357, doi: 10.1613/jair.953.

[19] Han, H.—Wang, W. Y.—Mao, B. H.: Borderline-SMOTE: A New over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D. S., Zhang, X. P., Huang, G. B. (Eds.): Advances in Intelligent Computing (ICIC 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3644, 2005, pp. 878–887,

doi: 10.1007/11538059_91.

[20] Arafa, A.—El-Fishawy, N.—Badawy, M.—Radad, M.: RN-SMOTE: Reduced Noise SMOTE Based on DBSCAN for Enhancing Imbalanced Data Classification. Journal of King Saud University – Computer and Information Sciences, Vol. 34, 2022, No. 8, pp. 5059–5074, doi: 10.1016/j.jksuci.2022.06.005.

[21] Li, J.—Zhu, Q.—Wu, Q.—Zhang, Z.—Gong, Y.—He, Z.—Zhu, F.: SMOTE-Nan-DE: Addressing the Noisy and Borderline Examples Problem in Imbalanced Classification by Natural Neighbors and Differential Evolution. Knowledge-Based Systems, Vol. 223, 2021, Art. No. 107056, doi: 10.1016/j.knosys.2021.107056.

[22] Yeh, C. W.—Li, D. C.—Lin, L. S.—Tsai, T. I.: A Learning Approach with Under- and Over-Sampling for Imbalanced Data Sets. 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), IEEE, 2016, pp. 725–729, doi: 10.1109/IIAI-AAI.2016.20.

[23] Xu, Z.—Shen, D.—Nie, T.—Kou, Y.: A Hybrid Sampling Algorithm Combining M-SMOTE and ENN Based on Random Forest for Medical Imbalanced Data. Journal of Biomedical Informatics, Vol. 107, 2020, Art. No. 103465, doi: 10.1016/j.jbi.2020.103465.

[24] Wang, X.—Zhang, R.—Zhang, Z.: A Novel Hybrid Sampling Method ESMOTE + SSLM for Handling the Problem of Class Imbalance with Overlap in Financial Distress Detection. Neural Processing Letters, Vol. 55, 2023, No. 3, pp. 3081–3105, doi: 10.1007/s11063-022-10998-0.

[25] Sauber-Cole, R.—Khoshgoftaar, T. M.: The Use of Generative Adversarial Networks to Alleviate Class Imbalance in Tabular Data: A Survey. Journal of Big Data, Vol. 9, 2022, No. 1, Art. No. 98, doi: 10.1186/s40537-022-00648-6.

[26] Mottini, A.—Lheritier, A.—Acuna-Agost, R.: Airline Passenger Name Record Generation Using Generative Adversarial Networks. CoRR, 2018, doi: 10.48550/arXiv.1807.06657.

[27] Rath, A.—Mishra, D.—Panda, G.—Satapathy, S. C.: Heart Disease Detection Using Deep Learning Methods from Imbalanced ECG Samples. Biomedical Signal Processing and Control, Vol. 68, 2021, Art. No. 102820, doi: 10.1016/j.bspc.2021.102820.

[28] Engelmann, J.—Lessmann, S.: Conditional Wasserstein GAN-Based Oversampling of Tabular Data for Imbalanced Learning. Expert Systems with Applications, Vol. 174, 2021, Art. No. 114582, doi: 10.1016/j.eswa.2021.114582.

[29] Lei, K.—Xie, Y.—Zhong, S.—Dai, J.—Yang, M.—Shen, Y.: Generative Adversarial Fusion Network for Class Imbalance Credit Scoring. Neural Computing and Applications, Vol. 32, 2020, No. 12, pp. 8451–8462, doi: 10.1007/s00521-019-04335-1.

[30] Xu, L.—Skoularidou, M.—Cuesta-Infante, A.—Veeramachaneni, K.: Modeling Tabular Data Using Conditional GAN. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 32 (NeurIPS 2019). Curran Associates, Inc., 2019, pp. 7335–7345, doi: 10.48550/arXiv.1907.00503.

[31] Bishop, C. M.: Pattern Recognition and Machine Learning. Springer New York, NY, 2006.

[32] Habibi, O.—Chemmakha, M.—Lazaar, M.: Imbalanced Tabular Data Modelization Using CTGAN and Machine Learning to Improve IoT Botnet Attacks Detection. Engineering Applications of Artificial Intelligence, Vol. 118, 2023, Art. No. 105669, doi: 10.1016/j.engappai.2022.105669.

[33] Moon, J.—Jung, S.—Park, S.—Hwang, E.: Conditional Tabular GAN-Based Two-Stage Data Generation Scheme for Short-Term Load Forecasting. IEEE Access, Vol. 8, 2020, pp. 205327–205339, doi: 10.1109/ACCESS.2020.3037063.

[34] Gulrajani, I.—Ahmed, F.—Arjovsky, M.—Dumoulin, V.—Courville, A. C.: Improved Training of Wasserstein GANs. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc., 2017, pp. 5767–5777, doi: 10.48550/arXiv.1704.00028.

[35] Lin, Z.—Khetan, A.—Fanti, G.—Oh, S.: PacGAN: The Power of Two Samples in Generative Adversarial Networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 31 (NeurIPS 2018). Curran Associates, Inc., 2018, pp. 1498–1507, doi: 10.48550/arXiv.1712.04086.

[36] Beckmann, M.—Ebecken, N. F. F.—Pires de Lima, B. S. L.: A KNN Undersampling Approach for Data Balancing. Journal of Intelligent Learning Systems and Applications, Vol. 7, 2015, No. 4, pp. 104–116, doi: 10.4236/jilsa.2015.74010.

[37] Dal Pozzolo, A.—Caelen, O.—Johnson, R. A.—Bontempi, G.: Calibrating Probability with Undersampling for Unbalanced Classification. 2015 IEEE Symposium Series on Computational Intelligence, 2015, pp. 159–166, doi: 10.1109/SSCI.2015.33.

[38] Li, Z.—Huang, M.—Liu, G.—Jiang, C.: A Hybrid Method with Dynamic Weighted Entropy for Handling the Problem of Class Imbalance with Overlap in Credit Card Fraud Detection. Expert Systems with Applications, Vol. 175, 2021, Art. No. 114750, doi: 10.1016/j.eswa.2021.114750.

Xiaoyan Zhao is a Master student in the School of Information and Electronic Engineering at the Shandong Technology and Business University, Yantai, China. Her current research interests include big data, deep learning.

Shaopeng Guan is currently Associate Professor with the School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, China. His current research interests include big data, networks and communications, etc.

Yuewei Xue is a Master student in the School of Information and Electronic Engineering at Shandong Technology and Business University, Yantai, China. Her current research interests include big data, deep learning.

Hao Pan is a Master student in the School of Information and Electronic Engineering at Shandong Technology and Business University, Yantai, China. His current research interests include big data, deep learning.