IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2019, accepted September 22, 2019, date of publication October 2, 2019,
date of current version October 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945034

VCOS: A Novel Synergistic Oversampling Algorithm in Binary Imbalance Classification

CHUNKAI ZHANG[1], YING ZHOU[2], AND YEPENG DENG[1]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518000, China
[2]Peng Cheng Laboratory, Shenzhen 518000, China

Corresponding author: Chunkai Zhang (ckzhang812@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800900, and in part by the Shenzhen Research Council Grant under Grant GJHZ20180928155209705.

ABSTRACT Learning from class-imbalanced data is a challenging problem as standard classification algorithms are designed to handle balanced class distributions. Scholars solve this problem by modifying classifiers or and generating artificial data by oversampling. The former usually design corresponding classifier to adapt them to the imbalanced data, while the latter exploits the sampling algorithm, which are the data preprocessing steps independent of the classifier. In this paper, we propose a novel synergistic oversampling algorithm to combine the oversampling and classification into one without training the classifier repeatedly, which can generate new pertinent samples according to the classification performance of the classifier without repeat training or deep understanding of the classifier, so the generated samples can guarantee the performance improvement of the classifier. Moreover, The proposed framework enclosures the oversampling method without traditional parameters in oversampling methods. Experimental results on several real-life imbalanced datasets demonstrate the effectiveness and efficiency of the proposed algorithm in binary classification problems.

INDEX TERMS Imbalanced classification, oversampling algorithm, variational auto-encoder, synergistic architecture, expected classifier.

I. INTRODUCTION

Imbalanced data refers to data set that exhibits an imbalanced class distributions, which have appeared in many fields, such as the medical diagnosis [1], [2], facial age estimation [3] and credit card fraud detection [4], image processing [32], [36], anomaly detection [33], where some data that occur with a low frequency but require more attention. Classification is one of the most common tasks of machine learning, and the standard learning algorithms have created much knowledge and discovery in many fields. However, when it comes to imbalanced data, the standard learning algorithms can easily ignore the class with fewer samples, because the standard learning algorithms assume balanced class distributions or equal misclassification costs [5]. The imbalanced classification has drawn much attention because the realistic data is more possible to be imbalanced due to the nature of data or the unequal misclassification costs [34], [35].

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

Imbalanced data usually present with sample rarity [6]. On the one hand, the absolute rarity, when there are few minority samples in the dataset, leads to the inadequate information of the minority class [7]. On the other hand, the relative rarity, when the size of the minority data is enough for training but relatively smaller than that of the majority data, increases classifier bias towards the majority class [8]. Inevitably, in the imbalanced classification, the noise will have a great impact on the standard learning algorithms [9], [9]. Furthermore, boundary sample overlapping [10] and data fragmentation [11] also make the classifier more difficult to represent the distributive characteristics of the data.

Solutions of the imbalanced classification consist of two aspects: the data level and the algorithm level. The algorithm level solutions try to design suitable learning algorithms in the imbalanced problems, which needs a deep understanding of the data and the classifier. The data-level methods balance data distribution, including oversampling, undersampling [12], and hybrid sampling, etc. Since the data-level methods exist as the data preprocessing, it has a wider range of applications. Moreover, oversampling focus more on the

minority by generating artificial data without losing precious information and has more stable performance, thus, it is favored in the imbalanced classification.

A. SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUES

SMOTE (synthetic minority over-sampling technique) [13] generates new samples by random interpolation between the minority samples and its neighbors. However, SMOTE may add some noise into the dataset, the Borderline-SMOTE [14] selects borderline minority data to be sampled. To avoid noise generated in the oversampling process, the CB-SMOTE [15] uses the local distribution of the boundary data to remove the noise points appearing in boundary and determine the oversampling rate for each sample. These algorithms can partly solve absolute rarity, but don't work in the data complexity.

B. DISTRIBUTION-BASED OVERSAMPLING ALGORITHM

Chen et al. [16] proposed a distribution-based oversampling algorithm to model the probability distribution function of minority data, and then use the model for oversampling. There are similar works such as Gauss distribution [16]–[18] and Weibull distribution [19]. However, they always assume a prior distribution, and they are limited. To avoid the limitations of the prior distribution, scholars use Variational auto-encoder [20], [21] and generative adversarial network [22] to generate new samples. As a result, the original minority subset is largely expanded. However, the independent oversampling architecture cannot guarantee to get the best recognition rate of the minority samples.

C. THE CLASSIFIER CHARACTERISTIC BASED SAMPLING

Zhang *et al.* [23] proposed an improved SMOTE in the Hilbert space rather than the Euclidean space when using SVM as the classifier. Zhang *et al.* [24] proposed CGMOS to guarantee the performance improvement of the oversampling algorithm when using the Bayesian classification. These algorithms need deep understanding for the classifier, and cannot generate exact samples to improve the classification performance.

D. THE CLASSIFICATION RESULT BASED SAMPLING

The above algorithms separate the oversampling process from the classification process, as they generate new samples at first, and use the expanded dataset to train the final classifier. However, the synergistic framework leads to a defect that the generated data does not guarantee the performance improvement of the classifier. For this purpose, there are also some genetic oversampling algorithms using classification performance as fitness. In GASMOTE [25], The genetic algorithm is used to optimize the oversampling rate of each minority sample. The new sample set is used to train the classifier, and the test Gmean of the original dataset is designed as the fitness; The MAHAKIL [26] algorithm simulates the reproductive process in the genetics to generate samples around the boundary rather than the whole original dataset; Guo *et al.* [27] proposed a hybrid sampling method based on the co-evolutionary algorithm. The classification result based sampling usually have good classification results, but it takes an expensive computational complexity, as it needs to train new classifier repeatedly with different training sets.

In this paper, we propose a novel framework that combines the oversampling with the classification process beyond the limitation of domain-knowledge, and it consists of three parts, the generation network, the present classifier, and the expected classifier. The Variational Auto-Encoder (VAE) model [28] is used as the generation network. To measure the quality of generated samples accurately, we creatively propose the expected classifier network, a temporal classifier updated with the generated samples upon the present classifier. To generate exact new samples, the proposed framework simultaneously trains the oversampling model and the classifier model, and both models interact with each other according to the loss function. We theoretically proved the rationality of the proposed framework, and the good performances in experimental results illustrate the superiority of the algorithm. The contributions of this work are as follows:

- The proposed framework combines the sampling process and the classifying process into one, and it measures the generated samples by the performance of the expected classifier to produce targeted samples.
- In the proposed framework, the generation network can interact with the classification network to guarantee the improvement of the classification performance.
- The proposed framework enclosures the oversampling method without traditional parameters, but automatically adjust the generation network according to the performance changes of the classifier, so it is more robust than traditional oversampling algorithms.

The rest of this work is organized as follows. Section II reviews the related work. Section III presents proof of the expected classifier and the proposed framework. We show the experimental results in Section IV and Section V concludes the paper as well as the future work.

II. RELATED WORK

In this section, some preliminaries about the variational auto-encoder based oversampling (VAEOS) and Logistic Regression (LR) classifier are offered in the proposed framework.

A. OVERSAMPLING ALGORITHM BASED ON VARIATIONAL AUTO-ENCODER

The oversampling network uses the variational auto-encoder (VAE) as the model of the probability density function of the minority samples in the dataset. The network architecture is as Fig.1, and the loss function is defined as follows.

$$L_g = \frac{1}{2}\|X - f(z)\|_2^2 \tag{1}$$

$$KL = \frac{1}{2}\left(1 + \log\left(\sigma^2\right) - \mu^2 - \sigma^2\right) \tag{2}$$

$$\text{loss} = L_g + \lambda KL \tag{3}$$

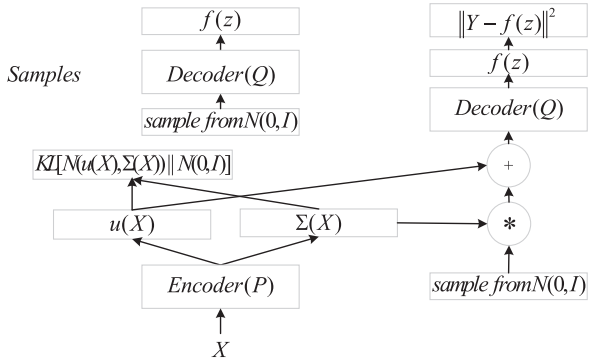


FIGURE 1. VAEOS network architecture.

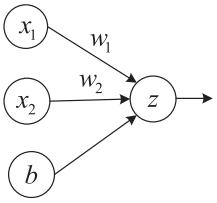


FIGURE 2. Structure diagram of logistic regression.

B. LOGISTIC REGRESSION

The Logistic Regression (LR) is used as the classifier network as shown in Fig.2. Let w be the weight, x be the current input, y be the real class label corresponding to x, the output and loss functions are (4) and (5), respectively, and the activation function of the neural network is shown as (6).

$$\hat{y} = \text{sigmoid}\left(w^T x + b\right) \tag{4}$$

$$\text{loss} = -(y \log \hat{y} + (1-y)\log(1-\hat{y})) \tag{5}$$

$$\text{sigmoid}(x) = s(x) = \frac{1}{\left(e^{-x}+1\right)} \tag{6}$$

III. VCOS ALGORITHM

A. ARCHITECTURE OF VCOS METHOD

The architecture of VCOS method is shown in Fig.3, it consists of four networks: the oversampling network encoder (E) and decoder (D), the classifier (C) and the expected classifier (EC). In this framework, the oversampling network regards the CVAE [30] (E and D) as the probability density function model of the data set. The quality of newly generated samples is measured by the performance of an expected classifier (EC), which is calculated on newly generated samples by random gradient descent based on the classifier trained by the original data. The classifier C collects advanced changes of new samples on EC and is the final classifier. In the following parts, we will give the concept and definition of the expected classifier EC, the gradient of EC loss on the oversampling network parameter E and G, and the overall framework in more detail, respectively.

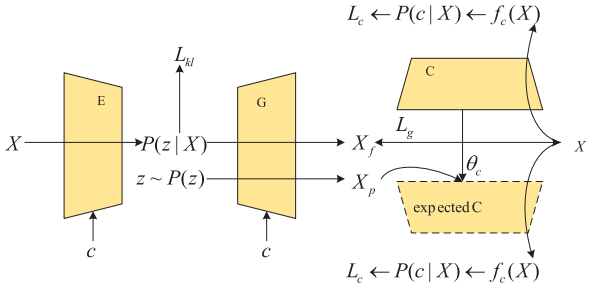


FIGURE 3. Architecture of VCOS.

B. THE EXPECTED CLASSIFIER EC

The impact of newly generated samples on classifier performance is achieved by updating the network parameters (w, b), if w and b calculated by newly generated samples have the same gradient direction as that of based on the original samples, they will reduce the loss value of the new network parameters with the input of original samples. The corresponding updated parameters with the input of newly generated samples are shown in formula (7), (8):

$$w = w - \eta \frac{\partial \text{ loss}}{\partial w} \left(x = x_{new}\right) \tag{7}$$

$$b = b - \eta \frac{\partial \text{ loss}}{\partial b} \left(x = x_{new}\right) \tag{8}$$

Cross-entropy is used as the classification cost function, which is defined as:

$$d\hat{y} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}} \tag{9}$$

$$z = w^T x + b \tag{10}$$

$$\hat{y} = s(z) \tag{11}$$

$$dz = \hat{y}(1-\hat{y}) \times d\hat{y} = \hat{y} - y \tag{12}$$

$$db = dz = \hat{y} - y \tag{13}$$

$$dw = xdz = (\hat{y} - y)x \tag{14}$$

Assuming that x_o is the original data for training and the newly generated data is x_n. The parameters of the expected classifier before updating are (w_o, b_o), and z_o is the output of the activation function with respect to the input x_o. After updating with Stochastic Gradient Descent (SGD), the parameters of the classifier are (w_n, b_n), and then the output of the activation function with the input x_o turn into z_n. Note that generated examples x_n are only used to update parameters, the final results are calculated by the original samples x_o. That is, the performance of the expected classifier with respect to x_o is used to measure the quality of generated samples x_n. According to the definition of SGD:

$$w_n = w - \eta dw \left(x = x_n\right) \tag{15}$$

$$b_n = b - \eta db \left(x = x_n\right) \tag{16}$$

where η indicates learning rate, and we get:

$$dw = \left(s\left(w_o x_n + b_o\right) - y_o\right) x_o \tag{17}$$

$$db = s\left(w_o x_n + b_o\right) - y_o \tag{18}$$

C. LOSS OF EXPECTED CLASSIFIER EC

Suppose the distribution of the new samples is consistent with the original one, and the class label is also the current hypothetical class label, positive. Then there comes the essential assumption of the proposed algorithm that the performance of the expected classifier would be improved if above suppose is given. From a qualitative point of view, this hypothesis is reasonable, as the power of the classifier will be stronger after adding reasonable new information. Based on formulas (17) and (18), in order to evaluate the quality of the generated samples, the classifier network parameters need to be updated with generated samples x_n in time, hence LR is used. To measure the quality of the generated data, the original data cannot be mixed with the new data.

$$w_n = w - \eta x_0 \left[s\left(w_0 x_n + b_0\right) - y_0 \right] \tag{19}$$

$$b_n = b - \eta \left[s\left(w_o x_n + b_o\right) - y_o \right] \tag{20}$$

According to formula (19) and (20), the parameters of the expected classifier are (w_n, b_n) after the SGD based on the new samples. The loss of expected classifier of the known data x_o is shown in formula (23):

$$z_n = w_n x_o + b_n \tag{21}$$

$$\hat{y}_n = s\left(z_n\right) \tag{22}$$

$$\text{loss}_{\text{exp}} = \text{loss}\left(y_0, \hat{y}_n\right) \tag{23}$$

With the chain rule, the gradient of generator parameters by the expected classifier is shown in equation (24).

$$\begin{aligned} \frac{\partial \text{ loss}_{\text{exp}}}{\partial \theta_g} &= \frac{\partial \text{ loss}_{\text{exp}}}{\partial x_n} \frac{\partial x_n}{\partial \theta_g} \\ &= \frac{\partial \text{ loss}_{\text{exp}}}{\partial z_n} \frac{\partial z_n}{\partial x_n} \frac{\partial x_n}{\partial \theta_g} \\ &= \frac{\partial \text{ loss}_{\text{exp}}}{\partial \hat{y}_n} \frac{\partial \hat{y}_n}{\partial z_n} \frac{\partial z_n}{\partial x_n} \frac{\partial x_n}{\partial \theta_g} \end{aligned} \tag{24}$$

where $\frac{\partial \text{ loss}_{\text{exp}}}{\partial \hat{y}_n} \frac{\partial \hat{y}_n}{\partial z_n} = dz_n = \hat{y}_n - y_0$, and $\frac{\partial x_n}{\partial \theta_g}$ are the backpropagation derivation of generator network G.

$$\begin{aligned} \frac{\partial z_n}{\partial x_n} &= \frac{\partial \left(w_n x_o + b_n\right)}{\partial x_n} \\ &= x_o \frac{\partial w_n}{\partial x_n} + \frac{\partial b_n}{\partial x_n} \end{aligned} \tag{25}$$

$$\begin{aligned} \frac{\partial w_n}{\partial x_n} &= \frac{\partial \left(w - \eta x_o \left[s\left(w_o x_n + b_o\right) - y_o \right]\right)}{\partial x_n} \\ &= -\eta x_o w_o s\left(w_o x_n + b_o\right)\left(1 - s\left(w_o x_n + b_o\right)\right) \end{aligned} \tag{26}$$

$$\begin{aligned} \frac{\partial b_n}{\partial x_n} &= \frac{\partial \left(w - \eta \left[s\left(w_o x_n + b_o\right) - y_o \right]\right)}{\partial x_n} \\ &= -\eta w_o s\left(w_o x_n + b_o\right)\left(1 - s\left(w_o x_n + b_o\right)\right) \end{aligned} \tag{27}$$

Formula (25) to (27) is substituted into formula (24) to obtain the gradient of the final expected classifier error to the new samples x_n.

D. DETAILS OF TRAINING VCOS

Corresponding to the above theoretical analysis, we proposed a novel synergistic over-sampling algorithm based on conditional variational auto-encoder (CVAE) and classifier performance, which is name Variational auto-encoder Classifier Oversampling VCOS. Thanks to this architecture, the performance of the classifier can directly feedback to the generator, to generate new samples with pertinence and improve the F1 score of minority data. In order to improve the classification performance focusing on the minority data, the feedback of LR classifier is used to guide the fine-tuning process of the oversampling model to obtain more pertinent generated samples in this architecture. The effect of the generated samples on the classifier is a continuous process, and the classifier accepts the fine-tuning from the generated samples during the process with a certain probability.

The LR classifier and generator are used in the framework to evaluate the performance of the over-sampling algorithm and adjust the model parameters of the generator in time to generate better samples. Firstly, a classifier and a generator with relatively acceptable ability are trained to generated samples. The performance of the classifier in the next iteration is determined by the updating direction of the expected classifier on the new examples, which in turn measures the quality of the newly generated samples. It means that good generated data should improve the classification performance of classifiers on minority data.

In order to guarantee the fundamental performance of the generator and the classifier, pre-trained operations are needed. Step1: The basic VAE model and LR classifier should be trained. VAE model references VAEOS. Step2: A batch of generated samples x_n are sampled from the generator, and then the classifier is updated with the generated samples with data label as the condition c in the generation process. The temporarily updated parameters are regarded as the expected classifier under the generated samples x_n. Step3: Calculate the loss function of the expected classifier based on the original training data set x_o, updating the parameters of the generator to minimize the expected classifier loss function. Step4: The classifier parameters are replaced by the expected classifier parameters with certain probability if the generated samples x_n can reduce the loss of classifier.

In order to generate pertinent samples, in addition to the Kullback-Leibler divergence and the reconstruction loss of examples, the objective function of the generator in VCOS also includes the expected classifier loss $loss_{exp}$.

$$\text{loss}_c = KL + \left\| x - x' \right\|^2 + loss_{exp} \tag{28}$$

During which, the calculation of KL and square error are the same as that of VAEOS. The effect of x_n on the expected classifier is calculated by the formula (24).

In the proposed method, we use the idea of the adversarial network, the parameters of the classifier are modified when the classification performance of the original minority data can be improved with the expected classifier. Otherwise,

Algorithm 1 VCOS Algorithm

Require:

Training set $X = \{x_i, y_i\}, i = 1, 2 \cdots m$, $y_i \in \{0, 1\}$, batch size m, initializes parameters of network θ_e, θ_g, θ_c, learning rate η, testing dataset X_{test}, training epoch $epoch$.

Ensure:

The predicted output of the testing dataset Y_{pred}

Pre-train

the conditional Variational auto encoder θ_e, θ_g

Pre-train

the classifier θ_c

for $i = 0, i < epoch, i++$ **do**

  Sample m samples from the real dataset $\{X_r, c\} \sim P_r$

  $L_c \leftarrow -\log\left(P\left(c|X_r\right)\right)$

  $z \sim E\left(X_r, c\right)$

  Sample according to the standard normal distribution $z_p \sim P_\varepsilon$

  $L_{kl} \leftarrow KL\left(Q\left(z|x_r, c\right) \| P_z\right)$

  $X_f \leftarrow G(z, c)$

  $X_p \leftarrow G\left(z_p, c\right)$

  $L_g = \frac{1}{2}\left\|X_r - X_f\right\|_2^2$

  Calculate θ_{cexp} according to formula (7) and (8)

  $L_{exp} \leftarrow \text{loss}_{exp}\left(c, \theta_{cexp}\right)$

  $\theta_c \leftarrow \theta_c - \eta \frac{\partial L_c}{\partial \theta_c}$

  $\theta_g \leftarrow \theta_g - \eta \frac{\partial \left(L_g + L_{\text{exp}}\right)}{\partial \theta_g}$

  $\theta_e \leftarrow \theta_e - \eta \frac{\partial \left(L_{kl} + L_g\right)}{\partial \theta_e}$

end for

$Y_{pred} = C\left(X_{test}\right)$

the parameters of the generator are modified. Under such a greedy strategy mentioned above, LR classifier is prone to fall into local minimum points, so Simulate Anneal strategy is used in this paper to accept some local minimum solutions with a certain probability.

Detailed algorithm pseudocode is shown in algorithm 1, VCOS uses VAE as the model in oversampling. In the phase of training classifier, the LR classifier, in the form of a single-layer neural network, is used to estimate the modification direction of the classifier on new samples. When the classification performance of the expected classifier is poor, the modification direction of new samples to the classifier is likely to be wrong, so the parameters of the generator are to be modified. When generating new samples, the application of CVAE makes it unnecessary to consider the reasonableness of the label in the phase of training in that the current samples are generated according to the condition label, and there must be some internal correlations. In the step of updating generator parameters, z and c in the VAE still remain unchanged, only the generator model is adjusted. Meanwhile, z is sampled from a uniform Gaussian distribution in Cumulative Distribution Function (CDF) to avoid the deviation of model adjustment based random sampling.

TABLE 1. Description of datasets.

No.	Datasets	#Ex.	#Atts.	IR
1	breastw(malignant)	699	9	1.9
2	heart2	303	13	1.18
3	diabetes0	768	8	1.87
4	yeast1	1484	8	2.46
5	vehicle2	846	18	2.88
6	vehicle3	846	18	2.99
7	vehicle1	846	18	2.99
8	vehicle0	846	18	3.25
9	ecoli1	336	7	3.36
10	segment-challenge(brick face)	1500	19	6.32
11	yeast6	1484	8	8.1
12	yeast3	1484	8	8.1
13	satimage4	6435	36	8.15
14	yeast-0-2-5-7-9_vs_3-6-8	1004	8	9.14
15	yeast-0-2-5-6_vs_3-7-8-9	1004	8	9.14
16	cardiotocography1	2126	23	11.08

TABLE 2. Confusion matrix.

Confusion Matrix		Predicted label	
		0	1
Real label	0	TP(True Positive)	FN(False Negative)
	1	FP(False Positive)	TN(True Negative)

IV. EXPERIMENTAL RESULTS

A. DATASETS

In order to prove the effectiveness of the proposed algorithm, 16 benchmark data sets in UCI and KEEL are used in this paper. The details are shown in Table 1, where the imbalanced rate is the sample size ratio. For the sake of ensuring the fairness of the experimental results, each data set was preprocessed uniformly in advance. What's more, the average value of 10-fold cross-validation is compared.

In this paper, we focus on the binary classification on imbalanced datasets. Choosing one class of the multi-class dataset as the minority class and combines other data into the majority class to obtain the highly imbalanced dataset.

B. PERFORMANCE EVALUATION

In the imbalanced binary classification, the confusion matrix, as shown in Table 2, is used to calculate the classification status of each class of samples. The following three metrics are used for analyzing the performances of the proposed algorithm: F1 score [31] of the minority samples, F1 score of majority samples, and Gmean. Formula (29) to Formula (32) are some commonly used evaluation criteria.

$$recall = \frac{TP}{TP + FN} \tag{29}$$

$$precision = \frac{TP}{TP + TP} \tag{30}$$

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{31}$$

$$Gmean = \sqrt{\frac{TP}{\text{TP} + \text{FN}} \times \frac{TN}{\text{TN} + \text{FP}}} \tag{32}$$

TABLE 3. F1 scores of the minority from LR classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	**82.10**	78.46	79.93	79.81	**82.10**	81.83	79.34	81.88	81.88
2	62.81	67.18	67.53	67.05	63.78	67.48	67.14	68.03	**68.31**
3	95.31	95.98	95.49	95.98	95.31	95.53	94.37	95.11	**96.43**
4	49.60	58.72	58.71	58.50	NA	NA	57.99	58.46	**93.85**
5	91.80	93.70	93.53	93.28	76.37	94.50	87.48	**94.55**	92.51
6	44.80	59.34	67.83	56.12	23.94	59.18	47.18	59.80	**96.17**
7	43.43	59.87	65.08	59.48	42.53	61.00	47.47	62.20	**97.98**
8	91.44	92.78	**93.41**	92.56	90.82	91.91	89.21	91.98	92.81
9	95.39	95.23	91.16	95.38	93.86	NA	95.93	95.21	**96.14**
10	98.69	98.69	98.46	98.69	NA	NA	93.07	98.97	**99.23**
11	71.94	73.99	72.93	73.59	NA	NA	73.74	72.03	**74.34**
12	63.61	76.60	74.06	75.72	67.38	NA	35.71	78.64	**95.90**
13	94.08	93.91	**95.72**	93.83	92.61	93.28	93.24	94.06	94.53
14	78.18	79.99	43.06	79.28	73.00	NA	80.46	77.77	**96.17**
15	38.64	59.29	62.49	59.72	43.41	NA	58.15	49.98	**92.51**
16	81.29	83.31	83.40	84.90	82.81	**85.47**	83.59	83.51	83.57

TABLE 4. F1 scores of the majority from LR classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	**85.51**	80.05	82.49	81.29	**85.51**	84.54	79.63	85.12	85.03
2	83.40	79.58	77.48	79.29	**83.48**	80.63	74.23	81.74	78.02
3	97.48	97.79	97.42	97.79	97.48	97.58	96.73	97.37	**97.99**
4	83.59	79.49	73.27	80.21	NA	NA	78.65	80.29	**99.30**
5	97.23	97.76	97.67	97.59	93.21	97.98	94.80	98.08	**99.35**
6	86.05	84.22	84.72	83.29	85.81	80.23	83.67	85.91	**99.67**
7	86.60	84.81	83.25	84.54	86.91	82.97	83.07	86.05	**99.85**
8	97.37	97.64	**97.79**	97.56	97.12	97.32	96.16	97.51	97.74
9	96.67	96.20	91.07	96.31	94.60	NA	96.82	96.44	**97.13**
10	99.81	99.81	99.77	99.81	NA	NA	98.79	**99.85**	99.23
11	96.89	96.69	95.70	96.65	NA	NA	96.96	95.94	**96.99**
12	96.57	97.29	95.75	97.05	96.61	NA	95.41	97.30	**99.64**
13	99.28	99.23	**99.47**	99.22	99.11	99.13	99.15	99.28	99.34
14	97.80	97.90	88.40	97.73	97.71	NA	97.71	97.73	**99.67**
15	94.71	95.60	93.81	95.52	95.32	NA	96.24	91.92	**99.35**
16	97.11	97.04	96.74	97.28	97.36	97.17	**98.18**	97.25	97.36

C. EXPERIMENTAL RESULTS

LR is used as a classifier to verify the rationality of VCOS. As an aside, the hyperparameters settings such as learning rate are also consistent with the VCOS. To compare the performance with different classifiers, SVM and random forest is also used and the hyperparameters are set as follows.

SVM: the data are normalized to [-1,1], radial basis function (rbf) kernel is used and gamma is set as $1/n_{features}$. LR: the learning rate is 0.001. Random forest 1: the number of trees is 100, and the criterion is Gini. Random forest 2: the number of trees is 250, and the criterion is Gini. All these classifiers are implemented with sklearn [37].

As we can see from Table 3, 4, 5, the borderline-SMOTE performs better in these compared oversampling except the proposed one, because the LR classifier is affected by the distribution of the borderline samples.

The results in Table 6, 7, 8 show that SVM needs more complex generated minority samples compared with LR, because the SVM divides the samples into different classes in Hilbert space while theses oversampling methods determine the nearest neighbors of samples in Euclidean space.

In table 9, 10, 11, the random forest 1 shows its power in classification and the proposed only gets 9 highest Gmean as an ensemble classifier. The random forest is more balanced in the F1 scores of the two classes so that it gets highest Gmean in some datasets. CGMOS performs best in the rest of oversampling methods, as it is designed in Bayes network classifier, which is more suitable for the random forest.

Results of Table 12, 13, 14 also show the superiority of the proposed method, and the CGMOS gains 3 highest F1 scores.

TABLE 5. Gmean of LR classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	**83.32**	79.11	80.93	80.35	**83.32**	82.84	79.57	83.14	82.95
2	70.05	74.44	74.50	74.33	70.86	74.62	73.59	75.02	**75.21**
3	96.29	97.02	97.17	97.02	96.29	96.50	96.68	96.19	**97.72**
4	60.46	70.20	69.88	69.89	NA	NA	70.37	70.04	**99.31**
5	93.94	96.10	96.34	95.94	80.80	97.26	94.51	96.58	**98.59**
6	57.10	72.98	80.83	70.33	37.09	74.38	61.17	72.34	**99.40**
7	55.22	72.44	79.38	72.33	54.15	74.67	61.40	73.70	**99.73**
8	94.43	96.04	**96.93**	95.96	93.90	95.90	95.34	95.11	95.56
9	95.90	95.75	91.34	95.90	94.46	NA	96.57	95.75	**96.62**
10	98.72	98.72	98.90	98.72	NA	NA	97.79	99.20	**99.45**
11	80.60	85.24	**91.10**	84.89	NA	NA	82.43	86.93	82.91
12	84.94	71.02	91.70	84.58	75.34	NA	44.44	85.10	**98.89**
13	96.14	96.95	**97.89**	96.87	94.99	96.98	96.77	96.07	96.20
14	85.06	87.29	71.35	87.78	79.51	NA	90.04	86.36	**99.31**
15	49.54	71.86	86.39	71.12	52.08	NA	69.53	74.22	**98.59**
16	88.18	90.43	91.91	91.84	89.86	**93.07**	92.18	90.37	91.95

TABLE 6. F1 of the minority from SVM classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	79.84	80.78	81.56	81.36	80.42	81.05	79.93	80.66	**81.88**
2	63.27	67.86	66.67	66.58	38.60	67.58	67.74	66.70	**68.31**
3	95.31	95.33	95.07	95.33	80.32	95.11	94.54	95.53	**96.43**
4	40.40	58.43	57.64	58.66	40.73	NA	58.93	59.10	**93.85**
5	91.87	93.04	**93.73**	92.86	88.57	92.06	90.00	90.97	92.51
6	46.79	62.50	63.55	62.29	45.99	62.62	59.98	61.67	**96.17**
7	55.49	64.79	66.51	65.47	56.10	65.10	61.86	63.92	**97.98**
8	92.83	**93.09**	91.94	92.82	90.02	92.65	87.61	91.59	92.81
9	95.44	95.23	93.67	95.63	84.31	NA	95.93	95.10	**96.14**
10	98.97	98.97	97.83	98.97	NA	NA	93.27	98.72	**99.23**
11	70.62	75.26	67.87	**75.51**	NA	NA	68.83	72.36	74.34
12	71.83	76.04	65.92	75.45	NA	NA	71.88	72.85	**95.90**
13	94.01	93.83	82.38	93.64	91.53	93.60	**94.95**	94.12	94.53
14	77.37	76.94	48.69	77.06	76.58	NA	83.50	73.78	**96.17**
15	28.23	53.67	41.76	54.15	29.69	NA	49.61	51.89	**92.51**
16	82.98	84.43	79.71	84.60	79.00	**86.33**	83.50	84.40	83.57

TABLE 7. F1 of the majority from SVM classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	83.63	82.38	84.06	83.08	83.84	83.85	80.38	82.15	**85.03**
2	**83.58**	80.17	78.38	79.31	74.55	80.94	75.32	80.77	78.02
3	97.48	97.47	97.20	97.47	88.69	97.36	96.85	96.87	**97.99**
4	83.96	80.82	73.41	81.05	84.06	NA	80.58	80.01	**99.30**
5	97.23	97.52	97.75	97.43	96.12	97.10	96.02	96.80	**99.35**
6	86.69	85.44	83.10	85.71	86.73	82.79	85.21	82.88	**99.67**
7	87.68	85.04	83.45	85.39	88.49	83.37	83.54	83.39	**99.85**
8	**97.74**	97.71	97.31	97.64	96.89	97.55	95.38	95.86	97.74
9	96.62	96.26	94.80	96.50	84.20	NA	96.76	96.35	**97.13**
10	**99.85**	**99.85**	99.65	**99.85**	NA	NA	98.83	99.81	99.23
11	96.86	96.88	93.82	96.92	NA	NA	96.73	92.92	**96.99**
12	96.89	96.96	93.37	96.88	NA	NA	96.92	96.19	**99.64**
13	99.28	99.22	97.11	99.19	99.00	99.18	**99.39**	98.74	99.34
14	97.79	97.24	87.81	97.43	97.62	NA	98.22	96.65	**99.67**
15	94.89	94.78	86.32	94.74	94.99	NA	96.06	91.67	**99.35**
16	97.30	97.34	96.25	97.29	97.50	97.17	**98.25**	95.89	97.36

TABLE 8. Gmean of SVM classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	81.19	81.43	82.44	82.02	81.65	81.99	80.26	81.26	**82.95**
2	70.36	75.02	73.95	73.97	50.92	74.73	74.35	74.06	**75.21**
3	96.29	96.49	96.77	96.49	85.27	96.17	96.61	95.47	**97.72**
4	51.98	69.64	69.12	69.86	52.25	NA	70.93	70.49	**99.31**
5	94.13	95.71	96.23	95.78	91.87	95.31	95.38	94.24	**98.59**
6	58.70	75.43	77.98	75.02	57.81	77.00	73.05	76.12	**99.40**
7	65.98	77.60	80.04	78.04	65.57	78.65	75.42	77.61	**99.73**
8	95.59	**96.63**	96.07	96.37	93.27	96.47	94.34	94.97	95.56
9	95.98	95.77	94.44	96.15	84.43	NA	96.52	95.74	**96.62**
10	99.20	99.20	99.22	99.20	NA	NA	97.83	98.94	**99.45**
11	78.78	85.76	**91.64**	85.79	NA	NA	76.52	82.72	82.91
12	79.65	86.34	90.95	85.94	NA	NA	79.75	86.89	**98.89**
13	95.81	96.87	96.04	96.78	93.61	**97.15**	96.46	92.02	96.20
14	83.50	87.19	83.55	86.53	83.21	NA	89.80	87.37	**99.31**
15	38.87	66.85	72.40	66.85	40.69	NA	59.72	75.71	**98.59**
16	89.25	91.00	91.73	91.22	86.63	**93.41**	90.80	85.95	91.95

TABLE 9. F1 score of the minority from random forest 1 classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	81.46	81.58	82.00	80.80	81.19	82.49	**89.64**	81.70	81.88
2	61.82	66.88	66.10	66.84	64.16	66.68	**68.53**	63.48	68.31
3	94.84	94.94	94.54	94.96	93.99	94.69	86.44	95.54	**96.43**
4	53.57	58.89	58.22	55.25	NA	NA	66.10	52.23	**93.85**
5	**97.92**	97.50	97.73	97.49	97.70	97.28	73.25	97.50	92.51
6	50.17	57.90	59.37	58.17	49.71	61.90	64.93	50.59	**96.17**
7	53.82	60.36	61.68	59.06	54.63	62.12	63.03	51.58	**97.98**
8	94.00	94.29	93.89	94.68	94.04	93.33	73.28	**95.09**	92.81
9	95.44	94.28	95.48	95.56	94.65	NA	**97.78**	95.42	96.14
10	98.69	98.96	99.22	98.96	NA	NA	72.00	98.96	**99.23**
11	73.95	74.59	**77.34**	73.86	NA	NA	65.59	74.28	74.34
12	74.80	75.42	77.93	76.57	NA	NA	67.46	73.20	**95.90**
13	96.47	**96.70**	96.48	96.63	96.15	96.44	72.03	96.68	94.53
14	74.80	74.48	76.89	79.08	77.55	NA	65.32	78.68	**96.17**
15	51.84	54.26	54.04	56.60	55.58	NA	66.63	49.33	**92.51**
16	82.91	82.00	77.76	81.36	81.19	82.15	68.97	81.06	**83.57**

TABLE 10. F1 score of the majority from random forest 1 classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	85.43	83.50	84.98	83.58	84.44	85.87	**88.43**	85.26	85.03
2	82.02	80.98	80.45	81.32	82.34	81.48	67.09	**82.64**	78.02
3	97.28	97.24	97.02	97.25	96.67	97.15	90.93	97.58	**97.99**
4	84.20	83.35	81.86	83.68	NA	NA	73.50	84.29	**99.30**
5	99.29	99.12	99.20	99.12	99.21	99.04	85.59	99.13	**99.35**
6	87.05	85.92	85.43	86.61	87.03	85.98	77.77	86.69	**99.67**
7	86.43	85.76	85.35	85.49	86.38	85.95	74.44	85.14	**99.85**
8	98.14	98.21	98.05	98.27	98.13	97.80	87.01	**98.43**	97.74
9	96.59	95.32	96.60	96.51	95.57	NA	**98.14**	96.07	97.13
10	99.81	99.85	**99.88**	99.85	NA	NA	93.60	99.85	99.23
11	**97.11**	96.98	96.98	96.95	NA	NA	92.98	97.11	96.99
12	97.14	97.05	97.05	97.25	NA	NA	93.60	97.04	**99.64**
13	99.56	**99.59**	99.57	99.58	99.52	99.55	94.94	**99.59**	99.34
14	97.63	97.34	97.44	97.78	97.68	NA	93.68	97.72	**99.67**
15	95.33	95.30	95.05	95.48	95.71	NA	94.11	95.38	**99.35**
16	97.68	97.53	96.19	97.46	**97.70**	95.60	94.99	97.41	97.36

TABLE 11. Gmean of random forest 1 classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	82.92	82.29	83.00	81.88	82.32	83.76	**89.16**	83.03	82.95
2	69.62	73.99	73.49	74.03	71.42	73.94	71.11	70.74	**75.21**
3	95.96	96.27	96.07	96.29	95.72	95.95	91.33	96.60	**97.72**
4	64.25	69.72	69.51	65.95	NA	NA	76.25	62.80	**99.31**
5	98.37	98.52	**98.60**	98.37	98.15	98.29	86.51	98.23	98.59
6	61.33	70.29	72.28	69.87	61.12	74.49	79.83	62.27	**99.40**
7	65.45	72.45	73.99	71.32	66.14	74.12	77.05	64.23	**99.73**
8	96.17	96.61	96.82	**97.58**	96.33	96.95	87.79	97.36	95.56
9	95.96	94.80	95.99	96.05	95.05	NA	**98.20**	95.80	96.62
10	98.72	98.98	99.23	98.98	NA	NA	93.80	98.98	**99.45**
11	81.60	83.93	88.74	83.15	NA	NA	**93.23**	82.22	82.91
12	82.73	84.66	89.40	84.83	NA	NA	93.80	80.95	**98.89**
13	97.99	98.33	97.61	98.26	97.94	**98.35**	95.07	98.07	96.20
14	82.12	83.05	85.38	85.97	84.03	NA	93.88	86.14	**99.31**
15	63.22	66.29	66.56	69.21	65.52	NA	93.90	61.17	**98.59**
16	89.49	90.53	87.07	89.97	89.11	90.12	**95.12**	88.16	91.95

TABLE 12. F1 score of the minority from random forest 1 classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	79.79	81.16	79.68	82.97	81.08	81.40	**89.68**	81.76	81.88
2	62.99	65.33	67.15	66.07	63.21	67.36	**69.04**	63.90	68.31
3	95.33	95.38	94.72	94.95	93.99	95.34	86.86	95.76	**96.43**
4	54.22	59.37	58.26	55.58	NA	NA	65.86	52.61	**93.85**
5	97.68	**98.16**	97.50	97.73	97.26	97.49	73.38	97.46	92.51
6	49.71	57.03	59.03	59.11	48.90	59.76	64.55	49.61	**96.17**
7	53.59	59.81	61.68	58.76	54.12	63.06	62.71	56.52	**97.98**
8	94.32	93.83	93.62	**94.55**	93.81	93.57	73.77	93.60	92.81
9	95.44	95.84	94.90	95.30	94.28	NA	**97.78**	95.56	96.14
10	98.96	98.96	98.73	99.22	NA	NA	72.24	98.69	**99.23**
11	74.06	76.40	**77.86**	74.07	NA	NA	65.14	74.05	74.34
12	75.20	75.18	77.09	75.63	NA	NA	67.18	75.02	**95.90**
13	96.32	96.55	96.34	96.43	96.22	96.56	71.96	**96.70**	94.53
14	75.55	79.37	77.07	78.63	79.02	NA	65.24	74.68	**96.17**
15	53.88	53.11	54.54	56.86	53.44	NA	67.06	53.20	**92.51**
16	83.44	81.19	81.48	82.94	**84.45**	81.50	68.77	81.66	83.57

TABLE 13. F1 score of the majority from random forest 2 classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	83.86	83.35	83.14	85.30	85.14	84.87	**88.44**	84.45	85.03
2	82.34	80.37	80.73	80.74	82.13	81.77	68.18	**82.83**	78.02
3	97.47	97.45	97.14	97.24	96.67	97.47	91.33	97.69	**97.99**
4	84.12	82.70	81.97	83.94	NA	NA	73.22	83.83	**99.30**
5	99.21	**99.37**	99.12	99.20	99.05	99.12	85.69	99.13	99.35
6	86.83	85.61	84.92	86.94	86.88	85.59	77.29	85.98	**99.67**
7	86.25	85.72	85.11	85.18	86.13	85.70	74.09	85.97	**99.85**
8	98.21	98.05	97.97	**98.29**	98.05	97.88	87.42	97.97	97.74
9	96.59	96.83	95.93	96.18	95.34	NA	**98.14**	96.51	97.13
10	99.85	99.85	99.81	**99.88**	NA	NA	93.70	99.81	99.23
11	97.07	**97.12**	97.01	96.90	NA	NA	92.81	97.07	96.99
12	97.14	97.06	96.90	97.13	NA	NA	93.52	97.14	**99.64**
13	99.55	99.57	99.56	99.55	99.53	99.57	94.92	**99.59**	99.34
14	97.63	97.77	97.49	97.66	97.78	NA	93.69	97.40	**99.67**
15	95.46	95.38	94.88	95.54	95.44	NA	94.17	95.48	**99.35**
16	97.83	97.39	96.85	97.64	**97.96**	95.57	94.95	97.22	97.36

Considering that the Gini of each leaf node is calculated when the random forest splits the leaf nodes, which is related to the Bayesian probability of the data distribution.

D. HYPOTHESIS TESTING

For a clearer analysis of the experimental comparison results, we use the Wilcoxon signed-rank test. The null hypothesis is that the overall performance of the two methods on different data sets is the same. Alternative hypothesis is that the overall performance of the two methods on different data sets is different. Here the significance level α is taken as 0.05.

As we can see from Table 15, the p-value in the table is much smaller than α, we can conclude the VCOS is definitely better than the CGMOS. What's more, the classifier does

TABLE 14. Gmean of random forest 2 classifier.

No.	none	smote [13]	border [14]	ndo [17]	weibu ll [19]	maha kil[26]	cgmos [24]	vaeos [20]	vcos
1	81.36	82.08	81.01	83.68	82.64	82.74	**89.19**	82.63	82.95
2	70.33	72.85	74.37	73.40	70.70	74.42	71.97	71.17	**75.21**
3	96.39	96.69	96.17	96.28	95.72	96.49	91.69	96.82	**97.72**
4	64.83	70.34	69.57	66.19	NA	NA	76.01	63.42	**99.31**
5	98.15	**98.76**	98.52	98.60	97.99	98.37	86.60	98.07	98.59
6	61.34	69.79	72.38	70.74	60.40	72.56	79.42	61.84	**99.40**
7	65.33	71.93	74.16	71.27	65.84	75.23	76.76	68.47	**99.73**
8	96.61	96.45	96.56	96.86	96.25	**97.02**	88.16	96.37	95.56
9	95.96	96.34	95.37	95.74	94.71	NA	**98.20**	96.05	96.62
10	98.98	98.98	98.94	99.23	NA	NA	93.89	98.72	**99.45**
11	82.25	85.76	89.61	83.53	NA	NA	**93.07**	82.18	82.91
12	83.41	84.25	89.51	84.32	NA	NA	93.72	83.29	**98.89**
13	97.90	98.18	97.52	98.23	98.01	**98.37**	95.05	98.26	96.20
14	82.84	86.59	85.65	86.49	85.98	NA	93.89	83.03	**99.31**
15	64.93	64.59	68.96	69.15	64.66	NA	94.34	65.17	**98.59**
16	90.45	89.75	90.32	90.21	90.98	88.75	**95.08**	88.69	91.95

TABLE 15. Result of hypothesis testing in F1 scores for minority samples.

	none	smote [13]	borderline [14]	ndo [17]	cgmos [24]	vaeos [20]
SVM	0.0005	0.0131	0.0008	0.0131	0.0008	0.0008
LR	0.0005	0.0023	0.0061	0.0072	0.0005	0.0031
rf1	0.0200	0.0262	0.0437	0.0262	0.0013	0.0200
rf2	0.0174	0.0340	0.0340	0.0494	0.0013	0.0229

TABLE 16. Result of hypothesis testing in Gmean.

	none	smote [13]	borderline [14]	ndo [17]	cgmos [24]	vaeos [20]
SVM	0.0005	0.0097	0.0061	0.0097	0.0006	0.0004
LR	0.0006	0.0045	0.0174	0.0045	0.0019	0.0041
rf1	0.0032	0.0113	0.0229	0.0151	0.0299	0.0072
rf2	0.0052	0.0200	0.0131	0.0229	0.0340	0.0052

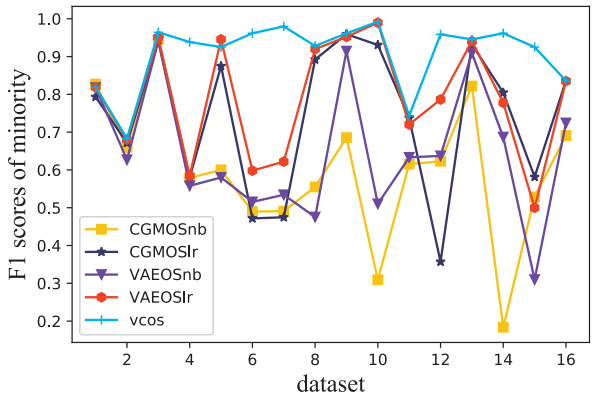


FIGURE 4. Comparison of oversampling frameworks based on different classifiers.

have an influence on the classification result with a same oversampling method, and a stronger classifier can reduce this impact such as the ensemble classifier random forest, and different oversampling may change the classification performance. Results in Table 16 show that the proposed method has superiority in comparison of Gmean.

E. COMPARISON OF CLASSIFIERS

Fig. 4 shows the classification performance comparison of CGMOS algorithm in the case of Naive Bayesian Model and Logical regression model. From the experimental results,

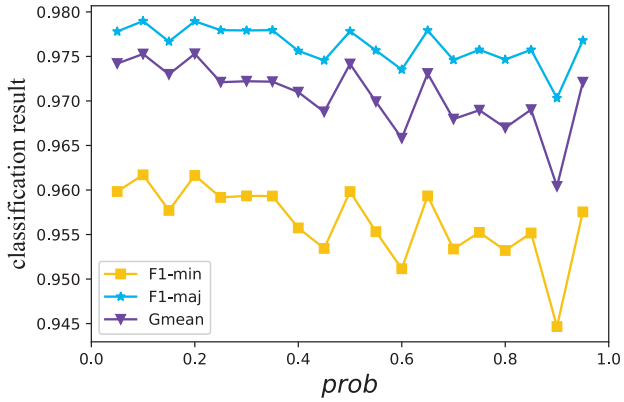


FIGURE 5. Effects of different *prob* on classification performance.

we can find that the results of our algorithm are better than other frameworks. The combination of oversampling and its suitable classifier proves the effectiveness of VCOS and the synergistic oversampling algorithm.

F. PARAMETER ANALYSIS

As we can see from Fig. 5, *prob* is the probability of updating the parameters of C with those of EC, which is constantly updated by new samples. There is a huge difference between the F1 scores of the minority and majority samples, and this is the specific form of imbalanced problems. Different probabilities in updating C result in classifiers with different performance, and a best *prob* is set as 0.5.

V. CONCLUSION

In this paper, a synergistic training framework for generation and classification is proposed creatively to adjust the parameters of the generator pertinently according to the classification performance of the expected classifier, to ensure the promotion of the generated samples to the classifier. The framework integrates the step of oversampling into the process of classifier training. Note that it does not need to set the oversampling rate, to avoid the negative impact caused by the improper oversampling rate. The final experimental result shows that the proposed network can effectively improve the F1 score of the minority samples, the hypothesis testing proves the correctness of the conclusion. The proposed method is designed on the binary imbalanced classification, we are going to improve it in multi-class classification with less limitation of the classifier.

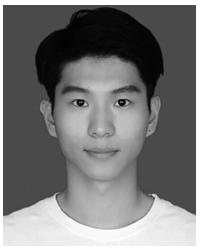
REFERENCES

[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2008.
[2] H.-L. Yin and T.-Y. Leong, "A model driven approach to imbalanced data sampling in medical decision making," *Stud. Health Technol. Inform.*, vol. 160, no. 2, pp. 60–856, 2010.
[3] N. Zarinabad, M. Wilson, S. K. Gill, K. A. Manias, N. P. Davies, and A. C. Peet, "Multiclass imbalance learning: Improving classification of pediatric brain tumors from magnetic resonance spectroscopy," *Magn. Reson. Med.*, vol. 77, no. 6, pp. 2114–2124, 2016.
[4] W.-L. Chao, J.-Z. Liu, and J.-J. Ding, "Facial age estimation based on label-sensitive learning and age-oriented regression," *Pattern Recognit.*, vol. 46, no. 3, pp. 628–641, 2013.

[5] P. K. Chan and S. J. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 1998, pp. 164–168.
[6] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Oct. 2002.
[7] G. M. Weiss and H. Hirsh, "A quantitative study of small disjuncts: Experiments and results," in *Proc. 17th Nat. Conf. Artif. Intell., 11th Conf. Innov. Appl. Artif. Intell.*, 2000, pp. 1–6.
[8] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 7–19, Jun. 2004.
[9] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi, "A resampling ensemble algorithm for classification of imbalance problems," *Neurocomputing*, vol. 143, pp. 57–67, Nov. 2014.
[10] V. García, R. A. Mollineda, and J. S. Sánchez, "On the k-NN performance in a challenging scenario of imbalance and overlapping," *Pattern Anal. Appl.*, vol. 11, nos. 3–4, pp. 269–280, 2008.
[11] F. J. Provost and G. M. Weiss, "Learning when training data are costly: The effect of class distribution on tree induction," 2003, *arXiv:1106.4557*. [Online]. Available: https://arxiv.org/abs/1106.4557
[12] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017.
[13] 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, no. 1, pp. 321–357, 2002.
[14] 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. Adv. Intell. Comput.*, 2005, pp. 878–887.
[15] L. Xiaojun, S. Yuxuan, and L. Haitao, "Clustering boundary over-sampling classification method for imbalanced data sets," *J. Zhejiang Univ. Sci.*, vol. 47, no. 6, pp. 944–950, 2013.
[16] S. Chen, "A generalized Gaussian distribution based uncertainty sampling approach and its application in actual evapotranspiration assimilation," *J. Hydrol.*, vol. 552, pp. 745–764, Sep. 2017.
[17] H. Zhang and Z. Wang, "A normal distribution-based over-sampling approach to imbalanced data classification," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2011, pp. 83–96.
[18] H. R. Sanabila, I. Kusuma, and W. Jatmiko, "Generative oversampling method (GenOMe) for imbalanced data on apnea detection using ECG data," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst.*, Oct. 2017, pp. 572–579.
[19] D.-C. Li, S. Hu, L.-S. Lin, and C.-W. Yeh, "Detecting representative data and generating synthetic samples to improve learning accuracy with imbalanced data sets," *PLoS ONE*, vol. 12, no. 8, 2017, Art. no. e0181853.
[20] C. Zhang, Y. Zhou, Y. Chen, Y. Deng, X. Wang, L. Dong, and H. Wei, "Over-sampling algorithm based on VAE in imbalanced classification," in *Proc. Int. Conf. Cloud Comput.*, 2018, pp. 334–344.
[21] Z. Wan, Y. Zhang, and H. He, "Variational autoencoder based synthetic data generation for imbalanced learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov./Dec. 2017, pp. 1–7.
[22] C. Lu and H. Shen, "Virtual sample generation approach for imbalanced classification," in *Proc. 9th Int. Symp. Parallel Architectures, Algorithms Program. (PAAP)*, Dec. 2019, pp. 177–182.
[23] C. Zhang, J. Guo, and J. Lu, "Research on classification method of high-dimensional class-imbalanced data sets based on SVM," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace*, Jun. 2017, pp. 60–67.
[24] X. Zhang, D. Ma, L. Gan, S. Jiang, and G. Agam, "CGMOS: Certainty guided minority oversampling," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1623–1631.
[25] H. Yudan, G. Qing, C. Zhihua, and Y. Lei, "Classification method for imbalance dataset based on genetic algorithm improved synthetic minority over-sampling technique," *J. Comput. Appl.*, vol. 35, no. 1, pp. 121–124, 2015.
[26] K. E. Bennin, J. Keung, P. Phannachitta, A. Monden, and S. Mensah, "MAHAKIL: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction," *IEEE Trans. Softw. Eng.*, vol. 44, no. 6, pp. 534–550, Jun. 2015.
[27] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern Recognit.*, vol. 46, no. 12, pp. 3460–3471, Dec. 2013.
[28] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114
[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1988.
[30] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
[31] T. K. Ho, "A data complexity analysis of comparative advantages of decision forest constructors," *Pattern Anal. Appl.*, vol. 5, no. 2, pp. 102–112, 2002.
[32] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 41–50, 2014.
[33] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2315–2322, Aug. 2018.
[34] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: Go beyond artificial intelligence," *Mobile Netw. Appl.*, vol. 23, no. 2, pp. 368–375, Apr. 2018.
[35] H. Lu, D. Wang, Y. Li, J. Li, X. Li, H. Kim, S. Serikawa, and I. Humar, "CONet: A cognitive ocean network," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 90–96, Jun. 2019.
[36] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 142–148, May 2018.
[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

CHUNKAI ZHANG received the Ph.D. degree from Shanghai Jiao Tong University, in 2001. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include data mining and cyber security.

YING ZHOU was born in Shaoyang, Hunan, China, in 1994. She received the bachelor's degree in computer science and technology from the Dalian University of Technology, Dalian, China, and the master's degree in computer science and technology from the Harbin Institute of Technology in Shenzhen, in 2019. She has been with the Peng Cheng Laboratory, Shenzhen, China, since 2019.



YEPENG DENG received the bachelor's degree in Internet of Things from the Taiyuan University of Technology, China. He is currently pursuing the master's degree in computer science and technology with the Harbin Institute of Technology, Shenzhen, China. His current research interests include computer vision, data mining, and machine learning.