CH-Smote Algorithm: A Novel Approach to Improve Random Forest Classification on Class Imbalanced Datasets

Jiao Wang[1,2], Norhashidah Awang[1*]

[1]School of Mathematical Sciences, UniversitiSains Malaysia, 11800 USM, Penang, Malaysia
[2]School of Mathematics and Statistics, Puer University, 665000, Puer, China
Email: wangjiao@student.usm.my, shidah@usm.my
[*]Corresponding Author

Received: 17.07.2024 Revised: 18.08.2024 Accepted: 20.09.2024

ABSTRACT
The Random Forest algorithm is widely recognized for its high prediction accuracy, robustness to noise, flexibility in parameter tuning, adaptability, and its ability to mitigate over-fitting across various fields. However, its performance degrades significantly when applied imbalanced datasets, often failing to achieve adequate classification accuracy. Although numerous techniques have been proposed in previous research to address this problem, many are computationally complex and tend to introduce additional noise. In contrast, sample generation techniques, are more widely employed than direct modifications to the classification algorithm. Therefore, this study proposes a novel hybrid sampling technique, termed the CH-SMOTE algorithm, which integrates the center of gravity principle with the SMOTE algorithm, and combines both over-sampling and under-sampling methods.. This algorithm is designed to be both computationally straightforward and highly effective. The CH-SMOTE algorithm addresses key limitations of the original SMOTE algorithm, such as blind synthesis and marginalization issues, while simultaneously mitigating over fitting and effectively handling class imbalance. To demonstrate its effectiveness, the CH-SMOTE algorithm was evaluated on seventeen datasets exhibiting varying degrees of class imbalance. The results indicate that the CH-SMOTE algorithm significantly enhances the classification performance of the Random Forest on imbalanced datasets.

Keywords: Random Forest, Imbalanced data, SMOTE algorithm, Classification

1. INTRODUCTION

The Random Forest algorithm, an ensemble learning method developed by Breiman and Cutler [1], utilizes multiple decision trees to train models and make predictions through majority voting. It addresses the limitations of single classifiers by enhancing overall performance. Random Forest exhibits strong performance and is applicable to a wide range of classification and prediction tasks. Studies have demonstrated that Random Forest achieves high predictive accuracy, robustness to outliers and noise, and is resistant to overfitting[1].

With the rapid advancement of technology, the volume of data the volume of data generated across industries is increasing exponentially, and much of this data is characterized by class imbalance. For example, in the telecommunications industry, there is a no table disparity between regular and fraudulent calls. Similarly, in medical diagnosis, the cost of misdiagnosing terminally ill patients as healthy far outweighs the consequences of misdiagnosing asymptomatic patients. In the banking sector, although the number of honest users significantly exceeds that of fraudulent users, the latter tends to caused more severe issues for financial institutions. As a result, in imbalanced datasets, greater emphasis is placed on the detection and classification of the minority class. Traditional classification algorithms are not suitable for handling imbalanced datasets. Consequently, research on addressing class imbalanced in datasets has become a central focus in the field[2].

Imbalanced data refers to data sets in which the number of samples from one class is significantly smaller than that of other classes. The class with the largest number of samples is termed the “majority class,” while the class with the fewer samples is referred to as the “minority class.” In binary classification problems, the minority class is often designated as the “positive class, ”while the majority class is called the “negative class”

Although Random Forest surpasses single classifiers like Decision Trees in terms of accuracy, generalization, and robustness, its performance is adversely affected by imbalanced training data, particularly in classifying minority samples. Traditional classification algorithms tend to produce much

lower prediction accuracy for the minority class compared to the majority class, leading to a significant decline in overall classifier performance. As a result, even with Random Forest's typically high accuracy, substantial misclassification rates often occur. Many studies optimize the performance of the Random Forest by addressing the imbalance of the dataset [4],but challenges such as data data scarcity, outliers, and noise exacerbate this problem[6].

Approaches to addressing the classification of imbalanced datasets are typically categorized into three categories: data-level methods, algorithm-level methods, and hybrid methods [7]. Data-level methods balance the datasets by reducing the number of samples in the majority class or increasing the number of samples in the minority class [9]. Data-level methods mainly use resampling to redistribute the training data of different classes in the data pre-processing stage [9]. Resampling techniques, includes under-sampling, over-sampling, and combined sampling[11].Under-sampling reduces the number of majority class samples, but risks discarding valuable information[12][14],with methods like the Random Under-sampling (RUS) [15] and the Near Miss algorithm (NM) [16]being prominent. Over-sampling, such as the Synthetic Minority Over-sampling Technique (SMOTE), generates new minority class samples but can lead to overfitting [18, 19, 20] and increased computational costs[20].Combined sampling integrates both strategies to balance class distributions.

Algorithm-level methods, on the other hand, involve designing algorithms that are inherently more suitable for imbalanced datasets, such as cost-sensitive learning, feature selection, and single-class learning. Hybrid methods combine the benefits of data-level resampling with algorithmic adjustments, offering a more comprehensive approach[22].

Among data-level methods, oversampling is widely used to mitigate data imbalance by generating new samples [23]. The SMOTE algorithm is onethe most classic over-sampling algorithm [24],though it has several limitations that need to be addressed.

In this study, we propose the CH-SMOTE algorithm, an enhancement of the SMOTE algorithm, to address the class imbalance issue. This algorithm synthesizes a balanced dataset, which is then classified using the Random Forest algorithm. The performance ofthe proposed method was evaluated across 17 imbalanced datasets and compared to other state-of-the-art techniques.

The remainder of this paper is organized as follows: Section 2 reviews related works. Section 3 details the CH-SMOTE algorithm. Section 4 introduces the evaluation metrics for classification algorithm. Section 5 presents the experimental results. The conclusion is drawn in Section 6.

2. Related Works

In random over-sampling, minority class samples are simply copied, resulting in randomness in selecting which samples to replicate. This process also introduces the problem of repeating samples from the original dataset, which does not effectively address the core issue of imbalanced data. To improve upon this, Chawla et al. introduced the Synthetic Minority Over-sampling Technique (SMOTE)[24].The SMOTE algorithm remains one of the most widely adopted data-level methods for addressing class imbalance.

The ccentral premise of the SMOTE algorithm is that neighboring samples around minority class samples are likely also minority class samples. The SMOTE algorithm achieves synthetic sample generation by identifying the K nearest neighbors for each minority class sample and interpolating new synthetic samples along the lines connecting the original sample to its neighbors. The interpolation is carried out using the following formula:

$$X_{new} = X_{origin} + rand(0,1) \times (X_i - X_{origin}) \quad i = 1, 2, \cdots, m \quad (1)$$

where X_{new} represents the synthetic sample, X_{origin} is the minority class sample, *N* is the total number of minority class samples, X_i （$i = 1, 2, \ldots, m$） is the *m* nearest neighbor samples adjacent to X_{origin} and rand (0,1) is a random number between 0 and 1. Fig. 1 displays the principle of the SMOTE algorithm.

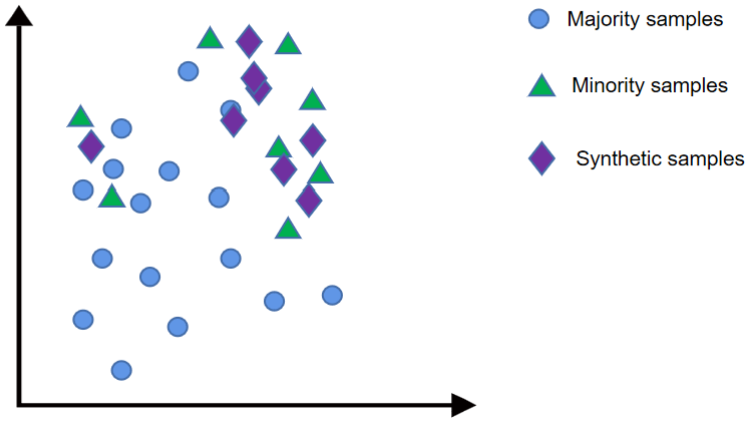

Fig 1: The principle of the SMOTE algorithm

While SMOTE offers improvements over random over-sampling, it still suffers from several limitations. Figure1 shows that the algorithm may generate synthetic minority class samples that include noise, boundary, and overlapping samples.A significant issue with the SMOTE algorithm is its indiscriminate selection of nearest neighbors, as there is no clear guidance on which factors should influence the selection of neighbors, and different choices can significantly affect the results. Additionally, the appropriate K value remains difficult to determine, often leading to poor handling of the underlying data distribution and marginalization of minority samples.

To overcome these limitations, several variant senhancements of the SMOTE algorithm have been proposed. The SMOTE - Nominal Continuous (SMNC) algorithm is a generalized version that accommodates both continuous and nominal data [24]. The Borderline-SMOTE algorithm (BSM) focuses on boundary samples to improve classification accuracy[26]. The Adaptive Synthetic Sampling algorithm (ADA) adjusts the generation of synthetic samples based on majority class densities in the 2 K-nearest neighbors of each sample [27]. The Safe-Level SMOTE algorithm (SLS) guides synthetic sample generation using the "Safe-Level" parameter[28], while Density-based SMOTE (DSM) clusters minority samples using density reachability concepts [29]. The Relocating Safe-level SMOTE algorithm (RSLS) refines synthetic sample placement based on the "Safe-Level" parameter [30].Douzas etal. [23] combined clustering techniques with the SMOTE algorithm to mitigate noise in the synthesize samples. These methods effectively increase the volume of data and balance the datasets. However, many focus onlyon increasing local sample information, often neglecting the global data distribution, which may lead to discrepancies between the synthetic and original datasets.

To address these concerns, Razavi et al.[31] introduced data repair techniques to improve classification performance on imbalanced datasets. Douzas and Bacao[33] applied Conditional Generative Adversarial Nets (CGAN) to generate new samples that better align with the overall distribution of the dataset. Mukherjee and Khushi[33]proposed the SMOTE-ENC algorithm, handles both continuous and nominal features, offering a more comprehensive solution for mixed data types.

The CH-SMOTE Algorithm

To address the limitations of the SMOTE algorithm, this study introduces an improved variant, termed the CH-SMOTE algorithm. Inspired by the K-Mean SMOTE [23], the CH-SMOTE algorithm employs K-Means clustering; however, randomness arises in selecting the value of *k*. Based on the principles of gravitational theory, samples within the same class are expected to share a common center of gravity [34], which serves as the representative point of the class. During sample generation, synthetic samples are positioned near the center of gravity of the minority class, providing directionality and overcoming the random nature of SMOTE's neighbor selection. Furthermore, samples generated excessively far from the center are eliminated, improving the distribution of synthetic samples, particularly near the class boundaries, and addressing the marginalization issue inherent in the SMOTE algorithm. Expanding the interpolation space within reasonablebounds effectively mitigates overfitting. The flow chart of the CH-SMOTE algorithm is displayed in Fig 2.

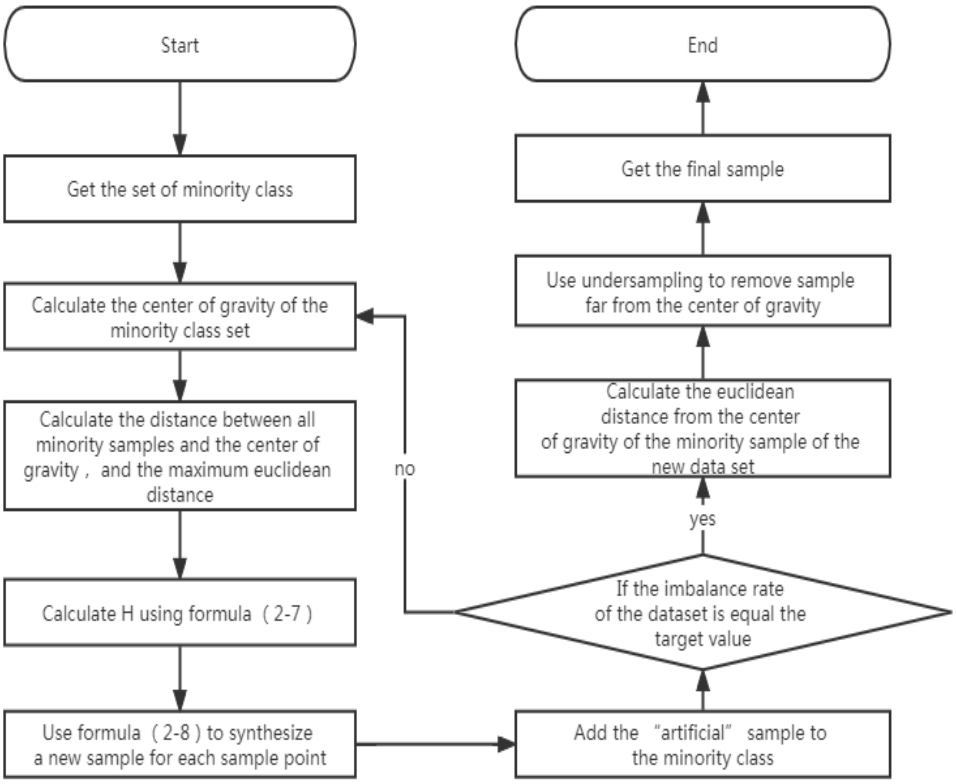

Fig 2: Flow chart of the CH-SMOTE algorithm

The steps of the CH-SMOTE are as follows:
Suppose the training set is$T$, each sample has *r* attributes, and the sample sizes of the minority class and the majority class are n_1 and n_2, respectively. Then the minority class sample set is denoted as **X**: $\mathbf{X}=\{\mathbf{X}_1,\mathbf{X}_2,\cdots,\mathbf{X}_{n_1}\}$, $\mathbf{X}_i=(x_{i1},x_{i2},\cdots,x_{ir}),(i=1,2,\cdots,n_1)$. The majority sample set is denoted as **Y**: $\mathbf{Y}=\{\mathbf{Y}_1,\mathbf{Y}_2,\cdots,\mathbf{Y}_{n_2}\}$, $\mathbf{Y}_j=(y_{j1},y_{j2},\cdots,y_{jr}),(j=1,2,\cdots,n_2)$.

Step 1 involves calculating the center of gravity for the minority class, denoted as X_{center}.

The center of gravity of the minority samples is obtained by Eq. (2). If each minority sample is visualized as a point in space, the center of these points represents minority class's center of gravity. Samples closer to the center exhibit a higher concentration of minority class characteristics. The CH-SMOTE algorithm's The CH-SMOTE algorithm's core principle is to generate synthetic samples around the center of gravity.

$$X_{center}=\left(\frac{1}{n_1}\sum_{i=1}^{n_1}x_{i1},\ \frac{1}{n_1}\sum_{i=1}^{n_1}x_{i2},\ \cdots,\ \frac{1}{n_1}\sum_{i=1}^{n_1}x_{ir}\right) \quad (2)$$

The base point of the over-sampling interpolation formula is modified to be the center of gravity. Now, the interpolation formula is:

$$X_{new}=X_{center}+rand(0,1)\times(X_{center}-X_{origin}) \quad (3)$$

where X_{new} is the new interpolated sample, X_{center} is the center of gravity of the minority sample, X_{origin} is the original minority class samples with X_{center} as the center of gravity , and $rand(0,1)$ is a random number between 0 and 1.

All the new interpolated samples are between the center of gravity and the original samples, as shown in Fig 3.Howeverthis interpolation method confines the interpolation space, potentially leading to overfitting.

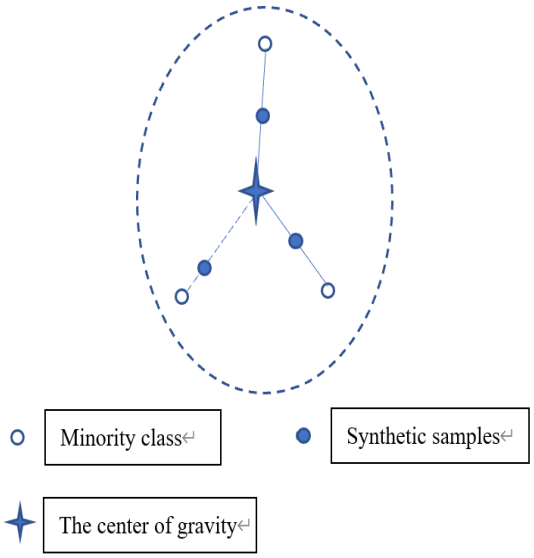

Fig 3: The principle of new interpolated samples

Step 2involves calculating the Euclidean distance between the center of gravity and each minority class sample.

Let Drepresent the Euclidean distance set from the center of gravity to each samples, which can be expressed as:

$$D = \{ d_1, d_2, \ldots, d_{n1} \} \quad (4)$$

where d_i represents the Euclidean distance between the center of gravity and the minority class samples.

Once the Euclidean distances are computed, the maximum distance $D_{\max}$ is determined as follows:

$$D_{\max} = \max \{ d_1, d_2, \ldots, d_{n1} \}. \quad (5)$$

Next, the relationship between the Euclidean distance d_i and the maximum Euclidean distance is evaluated. Eq. (6) is used to calculate the multiples H_i of d_i and $D_{\max}$.

$$H_i = D_{\max} / d_i. \quad (6)$$

Next, an integer for H_i is determined as

$$H = [H_i]. \quad (7)$$

Step 3involvessynthesizing new samples. The SMOTE algorithm interpolation formula is modified is modified as follows to obtain the new interpolation method:

$$X_{new} = X_{\text{center}} + rand(0, H) \times (X_{\text{center}} - X_{origin}) \quad (8)$$

where X_{new} is the new interpolated sample, X_{center} is the center of gravity of the minority samples, X_{origin} is the original minority class samples with X_{center} as the center of gravity, and $rand(0, H)$ represents a random number between 0 and *H*. The interpolation effect is shown in Fig 4. The range of the interpolation of the SMOTE algorithm is extended.The range is on the extension line between the center of gravity and the original minority class samples, but it does not exceed the minority class sample range.

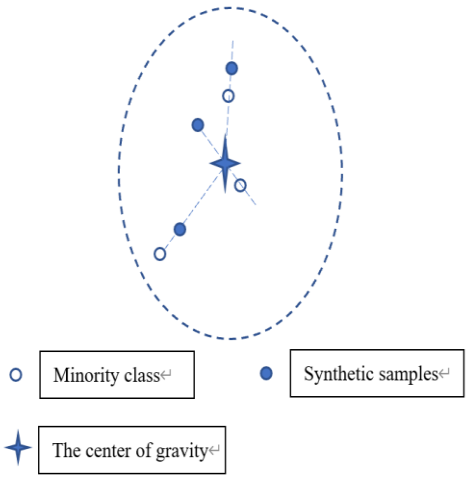

Fig 4: The principle of CH-SMOTE samples

Step 4 applies under-sampling to the synthesized minority class samples. By synthesizing samples according to Eq. (8), the number of generated synthetic samples for the minority class typically exceeds the required sample size.To correct this, under-sampling is applied to remove excess samples. Specifically, samples located farther from the center of gravity are eliminated, prioritizing the removal of samples that are spatially distant from the minority class center.

There are several methods to compute the distance between a sample and the center of gravity, including the Euclidean distance, the Manhattan distance, the Chebyshev distance, the Minkowski distance, the standardized Euclidean distance, the Mahalanobis distance, the cosine of the included angle, the Hamming distance.The appropriate distance metric can be selected based on the specific application.Ultimately, the number of synthetic samples is adjusted to achieve the desired class balance. The pseudocode for theCH-SMOTE algorithm is presented in Algorithm 1.

Algorithm 1: CH-SMOTE algorithm
Input: Imbalanced dataset *T*
Output: New balanced dataset

1. Divide imbalanced dataset *T* into minority class *X* and majority class *Y*; and record thesample sizes as *n1* and *n2*
2. For minority class *X* do
3. Calculate the center of gravity $X_{center} = (\frac{1}{n_1}\sum_{i=1}^{n_1} x_{i1}, \frac{1}{n_1}\sum_{i=1}^{n_1} x_{i2}, \cdots, \frac{1}{n_1}\sum_{i=1}^{n_1} x_{ir})$
4. For *i*=1 to *n1* do
5. Calculate $d_i(X_i, X_{center})$ based on Euclidean distance
6. Take the maximum Euclidean distance $D_{\max}$
7. Calculate the the multiple $H_i = D_{\max} / d_i$
8. Calculate the $H = [H_i]$
9. For new = 1: z(n1-n2) , $z > 1$ do
10. Synthesize new samples $X_{new} = X_{\text{center}} + rand(0, H) \times (X_{\text{center}} - X_{origin})$
11. For each X_{new} do
12. Calculate the Euclidean distance between X_{new} and X_{center}
13. Remove samples far from X_{center}
14. Make n1 + number of X_{new} = n2

15. Add X_{new} to X
16. End for
17. End for
18. End for
19. End for
20. Return the new balanced dataset

The synthetic samples generated by the CH-SMOTE algorithm are closer to the minority class's center of gravity, while fewer synthetic samples are generated in regions farther from the center. This selective sample generation reduces the occurrence of erroneous synthetic samples, particularly in sparse areas. For example, Fig 5shows the distribution of original and synthetic minority class samples for the Haberman dataset from the UCI database, illustrating the effectiveness of the CH-SMOTE algorithm..

Fig 5: An example used to illustrate the CH-SMOTE algorithm

4. Evaluation Indices

4.1 Evaluation Indices of the Imbalanced Dataset

The Imbalance Ratio **(IR)** is a crucial metric in the evaluation of imbalanced datasets. It quantifies the ratio between the negative (majority) class and the positive (minority) class [35]. The equation is defined as:

$$IR = \text{the number of negative} / \text{the number of positive} \quad (9)$$

The value of IR close to 1 indicates that the dataset is balanced. As the *IR* value increases, the disparity between the sizes of negative and positive classes becomes more pronounced.

4.2 Classification Performance Evaluation Indices of Random Forest

The Random Forest algorithm is primarily employed for classification and prediction tasks. Therefore, its classification performance is a key indicator for assessing the algorithm's effectiveness. The classification performance can be evaluated using a confusion matrix, as shown in Table 1[24].

Table 1: Confusion matrix of two-class data

		Classified Class	
		Positive	Negative
Actual Class	Positive	TP (True Positives)	FN (False Negatives)
	Negative	FP (False Positives)	TN (True Negatives)

In line with previous studies [37], several evaluation metrics are utilized, including Classification

Accuracy (Accuracy), Geometric mean (G-mean), *F*-values with β =1, and Out-of-Bag (OOB) error.The formula for OOB error is based on the number of decision trees (*nTree*) in the Random Forest. The indices are defined as:

Accuracy = (TP + TN)/ （TP+TN+FN+FP), (10)

$$G-\text{mean}=\sqrt{\text{Sensitivity}\times\text{Specificity}}, \quad (11)$$

where Sensitivity = TP/(TP + FN), Specificity = TN/(FP + TN)

$$F-\text{value}=\frac{(1+\beta^2)\times\text{Recall}\times\text{Precision}}{\beta^2\times(\text{Recall}+\text{Precision})}\times 100\%, \quad (12)$$

where Recall=TP/(TP+FN), Precision= TP/(TP+FP)

$$OOB\ \text{error}=\frac{\sum_{i}^{nTree} OOB\ \text{error}_i}{nTree}. \quad (13)$$

These evaluation indices provide a comprehensive understanding of the classification performance of the Random Forest algorithm on imbalanced datasets.

5. Experimental Analysis

5.1 Description of Data

To validate the effectiveness of the CH-SMOTE algorithm, 17 real-world datasets were collected from the UCI machine learning repository (http://archive-beta.ics.uci.edu/), as summarized in Table 2. The collection includes 9 binary-class datasets and 8 multi-class datasets. For the multi-class datasets, the one-versus-rest method was applied for binarization[37] or only two classes were selected for analysis [40].

Table 2: Brief description of the datasets

Name	Description
Spambase	A classification question of whether an email is spam or not.
Brest-Cancer	It is original Wisconsin Breast Cancer Database.
Tic-Tac-Toe Endgame	Possible configuration of Tic-Tac-Toe.
Heart Failure Clinical Records	Itis the medical records of 299 patients with heart failure.
Blogger	Contains information from 100 bloggers.
South German Credit	It is about whether actual credit is good or bad.
Car-Evaluation	It is derived from a simple hierarchical decision model.
Haberman	The survival of breast cancer surgery patients conducted.
Blood	It is obtained from the Blood Transfusion Service Center in Hsin-Chu City, Taiwan.
Contraceptive Method Choice	A subset of the 1987 Indonesian National Contraceptive Prevalence Survey.
HCV	It islaboratory value data for blood donors and hepatitis C patients. In this paper, 26 samples containing missing values were deleted.
Page Blocks	A classification of blocks of the document page layout detected
Ecoli	Describe the problem of classifying Ecoli proteins.
Cardiotocography	Contain 2126 samples data of fetal heart rate classification and electrocardiogram characteristics of uterine contractions by obstetricians.
Balance	The result of a simulation psychology experiment.
Poker Hand Hand（1vs3）	It isdata that predicts poker games, selects classes 1 and 3.
Poker Hand Hand（1vs4）	It is data that predicts poker games, selects classes 1 and 4.

Table 3 describes the characteristics of these datasets. It can be observed that these datasets have 3 to 57 features, 100 to 11,112 samples, and imbalance ratios from 1.54 to 113.97. These datasets belong to

different fields, and their features are diverse and wide-ranging, which provide a factual basis for the analysis and generalization of the results of the method in this study.

Table 3: Characteristics of the datasets

Name	Data size	No. of features	No. of positive	No. of negative	Imbalance Ratio
Spambase	4601	57	1813	2788	1.54
Breast-Cancer	683	9	239	444	1.86
Tic-Tac-Toe Endgame	958	9	332	626	1.89
Heart Failure Clinical Records	299	12	96	203	2.11
Blogger	100	5	32	68	2.13
South German Credit	1000	20	300	700	2.33
Car-Evaluation	1728	6	518	1210	2.34
Haberman	306	3	81	225	2.78
Blood	748	4	178	570	3.2
Contraceptive Method Choice	1473	9	333	1140	3.42
HCV	589	12	63	526	8.35
Page Blocks	5473	10	560	4913	8.77
Ecoli	336	7	29	307	10.59
Cardiotocography	2126	21	176	1950	11.08
Balance	625	4	49	576	11.76
Poker Hand Hand（1vs3）	11112	10	513	10599	20.66
Poker Hand Hand（1vs4）	10692	10	93	10599	113.97

5.2 Data Processing and Experimental Environment

The Random Forest algorithm was implemented in the R 4.1.1 using the “Random Forest” package. The Euclidean distance wasemployed to compute the distance between eachsample point and the center of gravity of the minority class.While various distance measures exist, the Euclidean distance was chosen for its widespread usagein related work.

In this experiment, the cross-validation was utilized,dividing each dataset into an training set (80% of the original dataset) and a test set (20% of the original dataset). To evaluate the efficacy of the CH-SMOTE algorithm in improving Random Forest's classification performance, several resampling methods were appliedto the datasets. The preprocessed dataset were subsequently classified using the Random Forest.

The CH-SMOTEalgorithm was compared with three baseline algorithms: ORI (direct classification using original data), under-sampling methods, and over-sampling methods, as summarized in Table 4. Two under-sampling methods, Random Under-Sampling (RUC) and Near Miss (NM), were utilized, alongside seven over-sampling methods: SMOTE (SM), Borderline-SMOTE (BSM), Density-based SMOTE (DSM), Safe-Level SMOTE (SLS), Relocating Safe-Level SMOTE (RSLS), SMOTE-Nominal Continuous (SMNC), and Adaptive Synthetic (ADA). Method 11 is the optimization algorithm proposed in this paper (CH-SMOTE).

Table 4: The description of resampling methods used in the experiment

Model	Method	Abbreviation
1	Original data is directly classified	ORI
2	Random Under-Sampling	RUC
3	Near Miss	NM
4	SMOTE	SM
5	Borderline-SMOTE	BSM
6	Density-based SMOTE	DSM
7	Safe-Level SMOTE	SLS
8	Relocating Safe-level SMOTE	RSLS
9	SMOTE - Nominal Continuous	SMNC
10	Adaptive Synthetic	ADA
11	CH-SMOTE	CH-SMOTE

In the R software environment, the “ROSE”, “them is”, and “smote family” packages were employed to implement the resampling methods used for comparison. The CH-SMOTE algorithm’s R code was designed to balance the above 17 datasets. It's important to note that default parameter settings were employed for all the compared methods.

5.3 Experimental Results

As discussed in the preceding section, data processed by ORI and resampling methods were classified using Random Forest. Different methods varied results across datasets, with the Random Forest classifiier trained on the training set and evaluated on the test set. Evaluation metrics, including Accuracy, G-mean, *F*-value, and OOB error, were used to assess the classifier's performance. Ther esults are presented in Tables 5-8, with the best results highlighted in bold for clarity.

Table 5 demonstrates the significant impact of data imbalance on the classification. For different methods, it is found that there are different results on different datasets. For instance, the SM algorithm performed best on the HCV dataset, while the SMNC algorithm excelled on the Cardiotocography and Poker Hand (1vs4) dataset. Overall, however, the CH-SMOTE algorithm consistently yielded the highest classification accuracy across most datasets.

Table 5: Results of experiment in terms of Accuracy

Data	ORI	RUC	NM	SM	BSM	DSM	SLS	RSLS	SMNC	ADA	CH-SMOTE
Spambase	0.949	0.940	0.953	0.966	0.955	0.948	0.964	0.967	0.964	0.959	0.964
Brest-Cancer	0.942	0.967	0.969	0.968	0.978	0.971	0.978	0.978	0.972	0.978	0.983
Tic-Tac-Toe Endgame	0.922	0.901	0.940	0.973	0.938	0.958	0.950	0.946	0.964	0.959	0.984
Heart Failure Clinical Records	0.817	0.821	0.872	0.911	0.890	0.883	0.870	0.831	0.902	0.854	0.915
Blogger	0.900	0.692	0.923	0.926	0.846	0.885	0.769	0.923	0.786	0.857	0.964
South German Credit	0.765	0.678	0.733	0.808	0.810	0.825	0.801	0.801	0.836	0.833	0.843
Car-Evaluation	0.977	0.971	0.986	0.989	0.992	0.984	0.984	0.989	0.988	0.980	0.994
Haberman	0.710	0.563	0.606	0.692	0.759	0.724	0.760	0.693	0.800	0.761	0.889
Blood	0.727	0.507	0.653	0.715	0.786	0.786	0.806	0.809	0.768	0.735	0.820
Contraceptive Method Choice	0.773	0.710	0.642	0.836	0.842	0.831	0.866	0.858	0.851	0.853	0.868
HCV	0.966	0.800	0.923	1.000	0.995	0.986	0.980	0.990	0.986	0.991	0.972
Page Blocks	0.975	0.959	0.964	0.997	0.985	0.988	0.992	0.983	0.986	0.984	0.987
Ecoli	0.941	0.715	0.750	0.975	0.992	0.983	0.991	0.991	0.967	0.967	0.984
Cardiotocography	0.988	0.897	0.972	0.994	0.982	0.993	0.989	0.992	0.995	0.990	0.986
Balance	0.936	0.632	0.700	0.915	0.939	0.946	0.890	0.898	0.935	0.914	0.974
Poker Hand（1vs3）	0.961	0.653	0.636	0.976	0.978	0.973	0.981	0.982	0.977	0.975	0.973
Poker Hand（1vs4）	0.992	0.658	0.842	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.995
Average	0.897	0.768	0.827	0.920	0.921	0.921	0.916	0.919	0.922	0.917	0.947

Fig 6illustrates the increase in Accuracyvalues achieved by each method relative to the ORI. The results indicate that various methods exert differing impacts on Random Forest classification accuracy across distinct datasets. While under-sampling techniques reduced classification accuracy for specific datasets, over-sampling approaches, particularly the CH-SMOTE algorithm, generally enhanced Random Forest classification accuracy in the majority of cases.

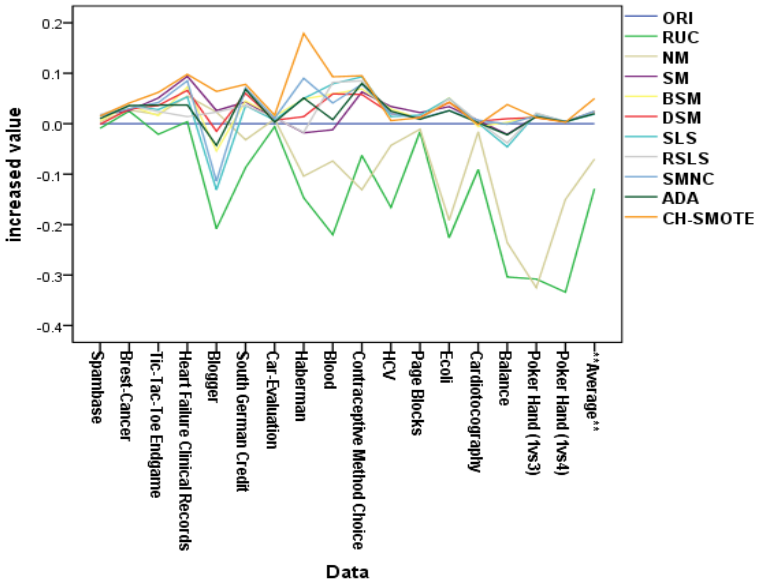

Fig 6: The increase in the value of Accuracy compared to ORI

In the context of imbalanced datasets, misclassification of the minority class as the majority class may still result in seemingly high overall classification accuracy[38]. For example, in a dataset with 100 samples, 10 representing the minority class and 90 the majority class, classification accuracy may still be misleading. Even if all minority class samples are misclassified while the majority class samples are correctly identified, the overall classification accuracy still remain at 90%. However, the Random Forest algorithm does not inherently account for imbalances in data distribution. Although it may achieve high overall classification accuracy, Random Forest tends to favor the majority class, leading to lower recognition rates for minority class samples in imbalanced datasets. This inherent bias diminishes the effectiveness of minority class recognition. To assess the impact of the imbalance ratio on classification performance, changes in the Sensitivity index were examined as the imbalance ratio increases, as shown in Fig 7.

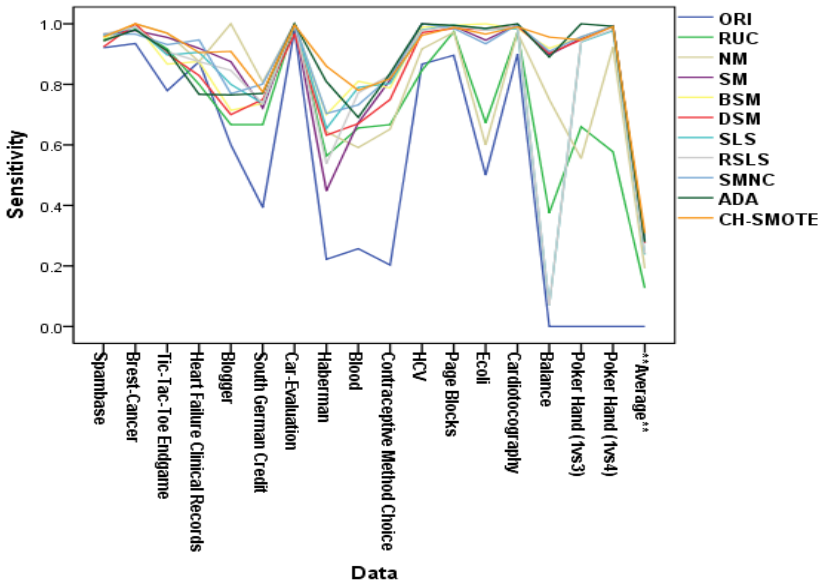

Fig 7: The Sensitivity of Random Forest with the increase of the imbalance ratio

Fig 7demonstrates that as the imbalance ratio increases, the sensitivity index on the original dataset declines consistently. Thus, larger imbalance ratios are associated with lower recognition rates of minority class samples by the Random Forest classifier. In the Balance, Poker Hand (1vs3), and Poker Hand (1vs4), which exhibit the highest imbalance ratios (*IR*),Random Forest misclassifies all minority class samples in both the training and the test sets as the majority class samples, while majority class samples are correctly classified. Despite the high overall accuracy, driven by the low representation of minority class samples, Sensitivity and Precision values are 0, while and Specificity reaches 1 in these cases. The CH-SMOTE algorithm, however, consistently maintains higher sensitivity index values compared to other resampling methods, achieving the highest average value overall.

However, it is essential to recognize that Accuracy alone may not sufficiently capture the classification challenges posed by imbalanced datasets. Therefore, this study also considers G-mean and *F*-value as complementary metrics. G-mean is a robust performance metric for imbalanced datasets, calculated as the geometric mean of Specificity and Sensitivity. A high G-mean value indicates strong classifier performance, as it reflects high Specificity and Sensitivity. The G-mean values obtained in this study are presented in

Table **6**. The G-mean values for the Balance, Poker Hand (1 vs. 3) and Poker Hand (1 vs. 4) datasets exhibit extreme cases. Fig 8illustrates the increase in G-mean values following resampling, compared to the ORI. As shown in Figure 8, the RUC method, unlike the other 10 methods, fails to improve G-mean for certain datasets. Nevertheless, in most cases, G-mean values improved after over-sampling. Overall, all resampling methods contribute to enhancing G-mean, with the CH-SMOTE algorithm yielding the most significant improvement in Random Forest classification performance.

Table 6: Results of experiment in terms of G-mean

Data	ORI	RUC	NM	SM	BSM	DSM	SLS	RSLS	SMNC	ADA	CH-SMOTE
Spambase	0.944	0.940	0.953	0.967	0.955	0.944	0.965	0.967	0.964	0.958	0.964
Brest-Cancer	0.94	0.963	0.971	0.968	0.977	0.976	0.978	0.978	0.972	0.978	0.983
Tic-Tac-Toe Endgame	0.883	0.899	0.940	0.973	0.931	0.945	0.947	0.945	0.961	0.954	0.984
Heart Failure Clinical Records	0.834	0.821	0.872	0.912	0.889	0.877	0.875	0.837	0.905	0.853	0.915
Blogger	0.775	0.690	0.935	0.935	0.845	0.837	0.763	0.920	0.784	0.874	0.953
South German Credit	0.604	0.679	0.733	0.801	0.805	0.818	0.795	0.795	0.835	0.830	0.841
Car-Evaluation	0.978	0.968	0.986	0.990	0.992	0.983	0.986	0.990	0.988	0.981	0.994
Haberman	0.449	0.563	0.610	0.612	0.752	0.690	0.731	0.646	0.815	0.758	0.892
Blood	0.473	0.498	0.666	0.712	0.784	0.765	0.803	0.804	0.770	0.732	0.818
Contraceptive Method Choice	0.438	0.713	0.641	0.835	0.835	0.818	0.858	0.855	0.852	0.852	0.872
HCV	0.922	0.797	0.923	1.000	0.995	0.985	0.980	0.989	0.987	0.991	0.972
Page Blocks	0.939	0.958	0.964	0.982	0.985	0.988	0.993	0.984	0.986	0.983	0.987
Ecoli	0.707	0.712	0.717	0.973	0.992	0.983	0.990	0.989	0.967	0.966	0.983
Cardiotocography	0.946	0.894	0.972	0.994	0.982	0.993	0.989	0.992	0.995	0.990	0.986
Balance	0.000	0.554	0.707	0.914	0.938	0.946	0.266	0.267	0.936	0.914	0.974
Poker Hand（1vs3）	0.000	0.653	0.636	0.975	0.978	0.973	0.969	0.970	0.977	0.976	0.973
Poker Hand（1vs4）	0.00	0.69	0.85	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.995

	0	3	9	6	6	6	9	1	6	6	
Average	0.637	0.764	0.829	0.914	0.919	0.913	0.875	0.878	0.923	0.917	0.946

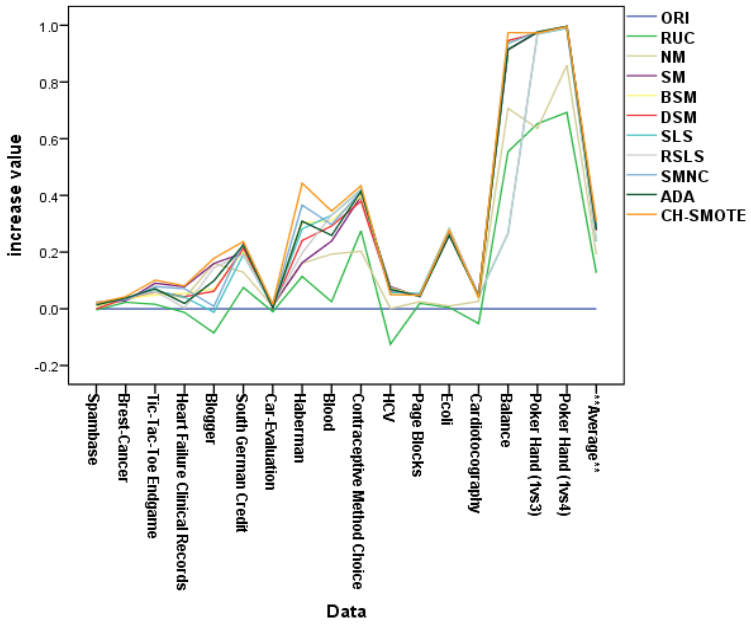

Fig 8: The increase in the value of G-mean compared to ORI

The *F*-value serves as a robust and comprehensive metric for evaluating classification performance in imbalanced datasets. It synthesizes the performance of classifiers by combining Recall and Precision. A higher the *F*-value indicates superior classifier.

Table 7 shows the *F*-values resulting from classifying different resampling methods using Random Forest, whereas Figure 9 depicts the relative increase in *F*-value compared to the ORI. According to

Table **7** and Fig 9, the *F*-value of ORIfor the Balance, Poker Hand (1vs3) and Poker Hand (1vs4) datasets is 0. Nevertheless, all resampling methods improve the *F*-value for these three datasets. Among the 11 methods, for the Ecoli dataset, the *F*-value of the NM and the ORI methods are identical. Moreover, the RUC method fails to improve the *F*-value for the Blogger, Cardiotocography, and HCV datasets, and the DSM method exhibits no improvement for the Spambase dataset. In other instances, resampling improved the *F*-value across datasets, albeit to varying extents depending on the method employed. Overall, the CH-SMOTE algorithm exhibits the most substantial improvement in *F*-value, with an increase of 23.8%.

Table 7: Results of experiment in terms of *F*-value

Data	ORI	RUC	NM	SM	BSM	DSM	SLS	RSLS	SMNC	ADA	CH-SMOTE
Spambase	0.935	0.937	0.953	0.970	0.953	0.934	0.968	0.970	0.963	0.956	0.963
Brest-Cancer	0.915	0.973	0.966	0.967	0.978	0.963	0.978	0.978	0.971	0.980	0.983
Tic-Tac-Toe Endgame	0.876	0.910	0.941	0.973	0.929	0.941	0.945	0.944	0.960	0.953	0.984
Heart Failure Clinical Records	0.718	0.821	0.848	0.907	0.875	0.866	0.853	0.812	0.900	0.846	0.916
Blogger	0.750	0.667	0.909	0.933	0.833	0.824	0.8	0.917	0.769	0.867	0.952
South German Credit	0.505	0.689	0.742	0.784	0.790	0.804	0.777	0.777	0.831	0.82	0.833
Car-Evaluation	0.961	0.973	0.985	0.988	0.992	0.983	0.983	0.988	0.988	0.979	0.994
Haberman	0.308	0.563	0.58	0.520	0.727	0.533	0.654	0.549	0.809	0.776	0.896

Blood	0.305	0.553	0.675	0.690	0.797	0.728	0.762	0.78	0.773	0.717	0.806
Contraceptive Method Choice	0.295	0.716	0.652	0.822	0.814	0.794	0.841	0.838	0.852	0.844	0.871
HCV	0.867	0.815	0.917	1.000	0.995	0.985	0.980	0.989	0.985	0.989	0.971
Page Blocks	0.876	0.962	0.964	0.98	0.985	0.987	0.992	0.981	0.986	0.984	0.987
Ecoli	0.667	0.690	0.667	0.972	0.992	0.981	0.990	0.989	0.966	0.970	0.983
Cardiotocography	0.915	0.904	0.971	0.993	0.982	0.993	0.989	0.992	0.995	0.990	0.986
Balance	0.000	0.462	0.667	0.911	0.936	0.944	0.126	0.133	0.936	0.913	0.973
Poker Hand(1vs3)	0.000	0.660	0.619	0.975	0.977	0.973	0.968	0.969	0.977	0.974	0.973
Poker Hand(1vs4)	0.000	0.698	0.800	0.996	0.996	0.996	0.989	0.990	0.996	0.996	0.995
Average	0.707	0.764	0.815	0.905	0.915	0.896	0.859	0.859	0.921	0.915	0.945

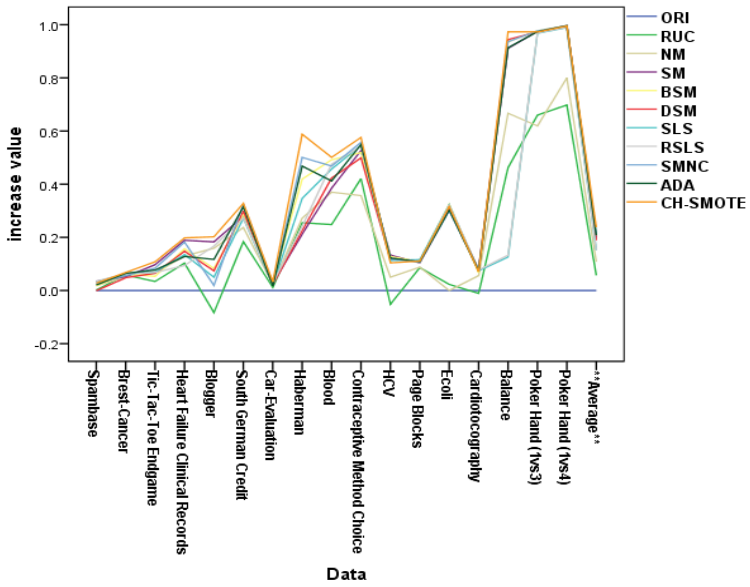

Fig 9: The increase in the value of *F*-value compared to ORI

In Random Forest algorithms, the OOB error is a critical metric for assessing overall classification performance, as it reflects the algorithm's generalization capability. A lower OOB error indicates superior generalization performance. Table 8 presents the OOB error values for Random Forest classification results, while Fig 10 illustrates the relative increase in OOB error across methods compared to ORI. For example, in the Haberman dataset, the classification model yielded an OOB error of 29.10% without resampling. However, applying the ROC method to the Balance dataset caused the OOB error to rise from 8.20% to 55.26%, substantially degrading model performance. Notably, the SLS, SMNC, and CH-SMOTE methods consistently reduced OOB error across all datasets. In general, all over-sampling techniques contributed to a reduction in OOB error, with CH-SMOTE demonstrating the most effective performance. This algorithm improved dataset balance, thereby enhancing the generalization ability of the Random Forest classifier.

Table 8: Results of experiment in terms of OOB error (%)

Data	ORI	RUC	NM	SM	BSM	DSM	SLS	RSLS	SMNC	ADA	CH-SMOTE
Spambase	4.70	5.04	5.55	3.66	3.96	4.89	3.30	3.38	4.57	4.15	3.83

Brest-Cancer	2.20	3.27	2.09	1.36	2.40	2.75	1.90	2.31	1.83	2.21	1.97
Tic-Tac-Toe Endgame	7.31	6.72	5.65	3.29	3.55	6.14	3.50	5.19	4.40	4.08	3.40
Heart Failure Clinical Records	15.06	14.84	17.65	12.66	14.53	12.34	9.21	10.53	12.65	9.48	12.04
Blogger	15.00	20.00	26.61	17.14	18.63	14.00	14.42	13.86	14.81	18.35	13.89
South German Credit	24.38	27.35	29.58	19.04	17.24	19.44	17.47	17.47	14.82	17.21	17.32
Car-Evaluation	1.16	2.83	1.57	1.00	1.04	1.00	1.06	0.95	0.98	1.42	0.88
Haberman	29.10	38.28	31.78	24.27	19.54	23.43	26.26	23.91	23.06	20.92	25.00
Blood	22.07	34.43	45.42	26.61	23.32	20.49	19.66	20.41	21.16	26.44	18.20
Contraceptive Method Choice	21.73	35.18	35.71	16.19	16.03	16.69	16.77	16.80	16.06	15.37	15.19
HCV	2.34	2.06	1.00	0.73	0.83	0.96	0.64	1.02	0.59	0.59	0.71
Page Blocks	2.38	3.64	3.68	1.53	1.52	1.41	1.38	1.17	1.78	1.63	1.37
Ecoli	3.37	10.87	6.52	2.31	1.44	0.64	1.36	1.13	1.43	2.85	1.22
Cardiotocography	1.71	5.19	6.41	0.58	0.61	0.82	0.63	0.77	0.58	0.61	0.45
Balance	8.20	55.26	29.49	8.52	7.21	5.43	7.09	6.89	7.27	7.86	5.32
Poker Hand(1vs3)	4.79	36.85	38.78	2.37	2.44	2.46	2.17	2.22	1.54	2.32	2.36
Poker Hand(1vs4)	0.89	24.32	28.38	0.44	0.42	0.48	0.54	0.55	0.34	0.44	0.43
Average	9.79	19.18	18.58	8.34	7.92	7.85	7.49	7.56	7.52	8.00	7.27

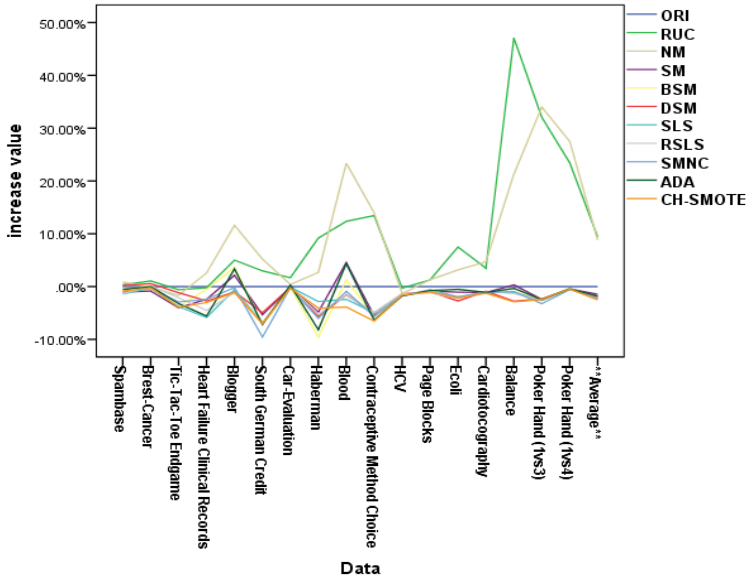

Fig 10: The increase in the value of OOB error compared to ORI

Fig 11presents the boxplots for Classification Accuracy, G-mean, *F*-value, and OOB error across all datasets. The degree of dispersion (i.e., the size of the box showing 25%-75% percentiles) for the CH-SMOTE algorithm was comparatively smaller. This indicates that the CH-SMOTE algorithm demonstrates relatively robust performance.

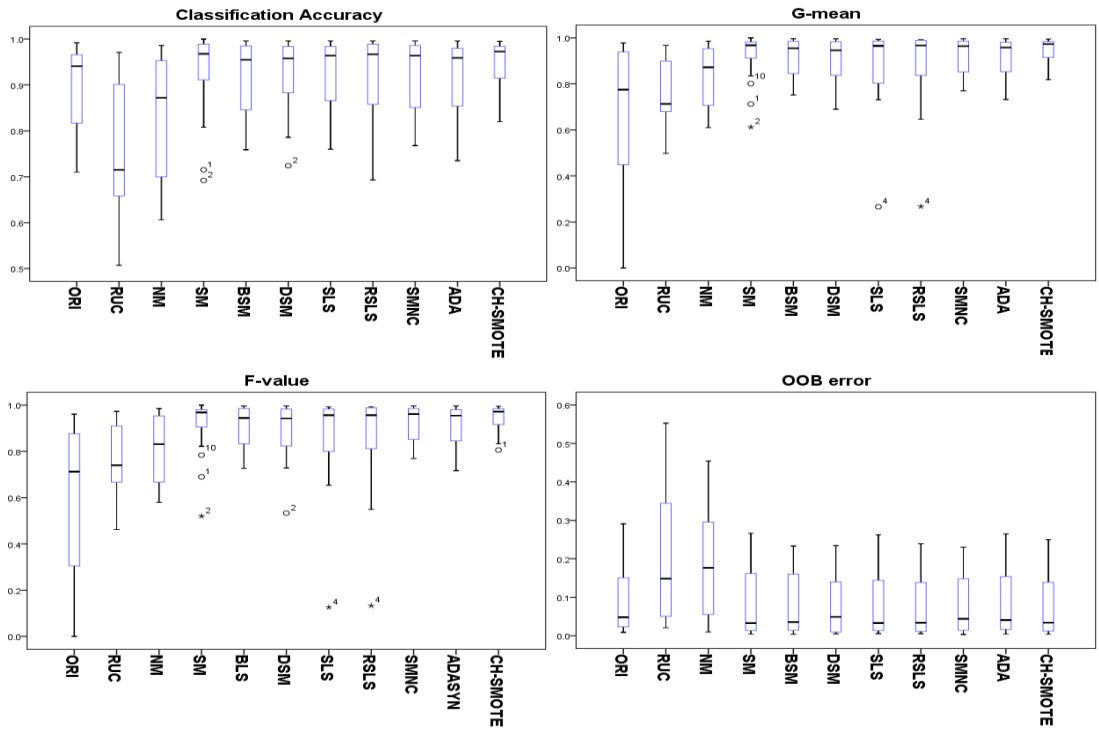

Fig 11: Boxplots of the indices

Based on theafore mentioned experiments, Random Forest classification performance on imbalanced datasets remains suboptimal. This study assesses classification performance using metrics such as Classification Accuracy, G-mean, *F*-value, and OOB error. The results suggest that over-sampling methods are more effective than under-sampling methods in enhancing Random Forest classification performance. Notably, the CH-SMOTE algorithm demonstrates superior capability in handling imbalanced data compared to other resampling methods. This advantage arises from CH-SMOTE's ability to generate synthetic samples that overcome the randomness inherent in the SMOTE algorithm, minimize marginalization issues, and mitigate overfitting. In summary, the CH-SMOTE algorithm outperformed other methods in terms of Classification Accuracy, G-mean, *F*-value, and OOB error. Additionally, the algorithm demonstrates robust performance across classifiers and datasets. Therefore, this study confirms that the CH-SMOTE algorithm is more effective in handling imbalanced datasets compared to other methods.

CONCLUSION

To enhance the performance of Random Forest in processing imbalanced datasets, a novel resampling method, the CH-SMOTE algorithm is proposed. Its efficacy is evaluated using 17 imbalanced datasets from the UCI repository. The results demonstrate that the proposed method is more effective in data preprocessing, significantly enhancing the efficiency and classification performance of the Random Forest algorithm. The CH-SMOTE algorithm presents several advantages over existing approaches. First, the sample synthesis process introduces directionality, positioning newly generated samples closer to the center of gravity of the minority class, thereby overcoming the marginalization issue present in the original SMOTE algorithm. Second, by leveraging the distance between samples and the center of gravity, the algorithm reasonably extends the interpolation space, thereby mitigating overfitting. Third, it incorporates under-sampling principles by removing synthesized samples distant from the center of gravity to minimize structural changes and prevent the generation of noisy values. Finally, the absence of K-nearest neighbor selection in the synthesis process eliminates the randomness associated with sample selection.

In the subsequent phases of this study, the CH-SMOTE algorithm will be applied to various scenarios, with an emphasis on its adaptation from binary-class to multi-class imbalance datasets. Additionally, methods to further enhance the efficiency of the CH-SMOTE algorithm will be investigated.

Statements and Declarations

Credit authorship contribution statement

Jiao Wang: Conceptualization, Methodology, Software, Formal analysis, Writing-original draft. NorhashidahAwang: Supervision, Validation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Submission declaration and verification

Both authors declare that the work described has not been published previously, that it is not under consideration for publication elsewhere, that its publication is approved and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright holder.

Acknowledgements

The first author would like to acknowledge the Education Reform Project of Puer University for supporting this research (Grant No. 2022JGYB008).

Data availability statements

The dataset supporting this study is publicly available in the UCI database: http://archive.ics.uci.edu/datasets**.**

REFERENCES

[1] Breiman L (2001) Random Forests. Machine learning45(1):5-32. https://doi.org/10.1023/A:1010933404324

[2] Thabtah F, Hammoud S, Kamalov F, Gonsalves A (2020) Data imbalance in classification: Experimental evaluation. Information Sciences 513:429-441. https://doi.org/10.1016/j.ins.2019.11.004

[3] Li Q, Mao Y (2014) A review of boosting methods for imbalanced data classification. Pattern Analysis and Applications 17(4):679-693. https://doi.org/10.1007/s10044-014-0392-8

[4] Xu Z, Shen D, Nie T, Kou Y (2020) A hybrid sampling algorithm combining M-SMOTE and ENN based on Random Forest for medical imbalanced data. Journal of Biomedical Informatics 107:103465. https://doi.org/10.1016/j.jbi.2020.103465

[5] Liang D, Yi B, Cao W, Zheng Q (2022) Exploring ensemble oversampling method for imbalanced keyword extraction learning in policy text based on three-way decisions and SMOTE. Expert Systems with Applications 188:116051. https://doi.org/10.1016/j.eswa.2021.116051

[6] Weiss GM (2004) Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6(1):7-19.https://doi.org/10.1145/1007730.1007734

[7] Johnson JM, Khoshgoftaar TM (2019) Survey on deep learning with class imbalance. Journal of Big Data 6(1):1-54. https://doi.org/10.1186/s40537-019-0192-5

[8] Khushi M, Shaukat K, Alam TM, Hameed IA, Uddin S, Luo S, Yang XY, Reyes MC (2021) A comparative performance analysis of data resampling methods on imbalance medical data. IEEE Access 9:109960-109975. https://doi.org/10.1109/ACCESS.2021.3102399

[9] García V, Sánchez JS, Mollineda RA (2010) Exploring the performance of resampling strategies for the class imbalance problem. In Trends in Applied Intelligent Systems: 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part I 23 (pp. 541-549). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13022-9_54

[10] Liu J, Hu Q, Yu D (2008) A comparative study on rough set based class imbalance learning. Knowledge-Based Systems 21(8):753-763. https://doi.org/10.1016/j.knosys.2008.03.031

[11] Burnaev E, Erofeev P, Papanov A (2015) Influence of resampling on accuracy of imbalanced classification. In Eighth international conference on machine vision (ICMV 2015) 9875:423-427. SPIE.https://doi.org/10.1117/12.2228523

[12] Liu AYC (2004) The effect of oversampling and undersampling on classifying imbalanced text datasets. Dissertation, University of Texas at Austin.http://dx.doi.org/10.26153/tsw/12300

[13] Yen SJ, Lee YS (2006) Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In Intelligent Control and Automation: International Conference on

Intelligent Computing, ICIC 2006 Kunming, China, August 16–19, 2006 (pp. 731-740). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-37256-1_89

[14] Jindaluang W, Chouvatut V, Kantabutra S (2014) Under-sampling by algorithm with performance guaranteed for class-imbalance problem. In 2014 International Computer Science and Engineering Conference (ICSEC) pp:215-221. https://doi.org/10.1109/ICSEC.2014.6978197 Alam TM, Shaukat K, Hameed IA, Khan WA, Sarwar MU, Iqbal F, Luo S (2021) A novel framework for prognostic factors identification of malignant mesothelioma through association rule mining. Biomedical Signal Processing and Control 68:102726. https://doi.org/10.1016/j.bspc.2021.102726

[15] Prusa J, Khoshgoftaar TM, Dittman DJ, Napolitano A (2015) Using random undersampling to alleviate class imbalance on tweet sentiment data. In 2015 IEEE international conference on information reuse and integration (pp. 197-202). IEEE. https://doi.org/10.1109/IRI.2015.3

[16] Mani I, Zhang I (2003) kNN approach to unbalanced data distributions: a case study involving information extraction. In Proceedings of workshop on learning from imbalanced datasets 126(1):1-7. ICML. https://www.site.uottawa.ca/~nat/Workshop2003/jzhang.pdf

[17] Marqués AI, García V, Sánchez JS (2013) On the suitability of resampling techniques for the class imbalance problem in credit scoring. Journal of the Operational Research Society 64(7):1060-1070. https://doi.org/10.1057/jors.2012.120

[18] Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, Hawalah A, Hussain A (2016) Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. IEEE Access 4:7940-7957. https://doi.org/10.1109/ACCESS.2016.2619719

[19] Zheng Z, Cai Y, Li Y (2015) Oversampling method for imbalanced classification. Computing and Informatics 34(5):1017-1037. http://dml.mathdoc.fr/item/cai1277

[20] Ganganwar V (2012) An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering 2(4):42-47. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3344&rep=rep1&type=pdf

[21] Leevy JL, Khoshgoftaar TM, Bauder RA, Seliya N (2018) A survey on addressing high-class imbalance in big data. Journal of Big Data 5(1):1-30. https://doi.org/10.1186/s40537-018-0151-6

[22] AbdElrahman SM, Abraham A (2015) Class imbalance problem using a hybrid ensemble approach. International Journal of Hybrid Intelligent Systems 12(4):219-227. https://doi.org/10.3233/HIS-160217

[23] Douzas G, Bacao F, Last F (2018) Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. Information Sciences 465:1-20. https://doi.org/10.1016/j.ins.2018.06.056

[24] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16: 321-357. https://doi.org/10.1613/jair.953

[25] Xu Z, Shen D, Nie T, Kou Y, Yin N, Han X (2021) A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data. Information Sciences 572:574-589. https://doi.org/10.1016/j.ins.2021.02.056

[26] Han H, Wang WY, Mao BH (2005, August) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878-887). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11538059_91

[27] He H, Bai Y, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE. https://doi.org/10.1109/IJCNN.2008.4633969

[28] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Pacific-Asia conference on knowledge discovery and data mining (pp. 475-482). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01307-2_43

[29] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2012) DBSMOTE: density-based synthetic minority over-sampling technique. Applied Intelligence 36(3):664-684. https://doi.org/10.1007/s10489-011-0287-y

[30] Siriseriwan W, Sinapiromsaran K (2016) The effective redistribution for imbalance dataset: relocating safe-level SMOTE with minority outcast handling. Chiang Mai J. Sci 43(1):234-246. http://cmuir.cmu.ac.th/jspui/handle/6653943832/66081

[31] Razavi-Far R, Farajzadeh-Zanjani M, Saif M (2017) An integrated class-imbalanced learning scheme for diagnosing bearing defects in induction motors. IEEE Transactions on Industrial Informatics 13(6):2758-2769. https://doi.org/10.1109/TII.2017.2755064

[32] Douzas G, Bacao F (2018) Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications 91:464-

471.https://doi.org/10.1016/j.eswa.2017.09.030
[33] Mukherjee M, Khushi M (2021) SMOTE-ENC: A novel SMOTE-based method to generate synthetic data for nominal and continuous features. Applied System Innovation 4(1):18. https://doi.org/10.3390/asi4010018
[34] Cao ZF (2014) Study on Optimization of Forest Algorithm. Dissertation, Capital University of Economics and Business. https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CDFD1214&filename=1014220587.nh
[35] Noorhalim N, Ali A, Shamsuddin SM (2019) Handling imbalanced ratio for class imbalance problem using SMOTE. In Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017) (pp. 19-30). Springer, Singapore.
[36] https://doi.org/10.1007/978-981-13-7279-7_3
[37] Li J, Zhu Q, Wu Q, Fan Z (2021) A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. Information Sciences 565:438-455. https://doi.org/10.1016/j.ins.2021.03.041
[38] Sun Y, Kamel M S, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern recognition 40(12):3358-3378. https://doi.org/10.1016/j.patcog.2007.04.009
[39] Rajanala S, Bates S, Hastie T, Tibshirani R (2022) Confidence Intervals for the Generalisation Error of Random Forests. arXiv preprint arXiv:2201.11210. https://doi.org/10.48550/arXiv.2201.11210
[40] Zhang A, Yu H, Huan Z, Yang X, Zheng S, Gao S (2022) SMOTE-RkNN: A hybrid re-sampling method based on SMOTE and reverse k-nearest neighbors. Information Sciences 595:70-88. https://doi.org/10.1016/j.ins.2022.02.038
[41] Hopkins M, Reeber E, Forman G, Suermondt J (1999) Spambase. UCI Machine Learning Repository. https://doi.org/10.24432/C53G6X
[42] Wolberg W (1992) Breast Cancer Wisconsin (Original). UCI Machine Learning Repository. https://doi.org/10.24432/C5HP4Z
[43] Aha D (1991) Tic-Tac-Toe Endgame. UCI Machine Learning Repository. https://doi.org/10.24432/C5688J
[44] Ahmad T, Munir A, Bhatti SH, Aftab M, Ali RM (2020) Heart failure clinical records. UCI Machine Learning Repository. https://archive-beta.ics.uci.edu/dataset/519/heart+failure+clinical+records
[45] Gharehchopogh FS, Khaze SR (2013) BLOGGER. UCI Machine Learning Repository. https://doi.org/10.24432/C5HK6P.
[46] Ulrike G (2019) South German Credit. UCI Machine Learning Repository. https://archive-beta.ics.uci.edu/dataset/522/south+german+credit
[47] Bohanec M (1997) Car Evaluation. UCI Machine Learning Repository. https://doi.org/10.24432/C5JP48
[48] Haberman S (1999) Haberman's Survival. UCI Machine Learning Repository. https://doi.org/10.24432/C5XK51
[49] Yeh IC (2008) Blood Transfusion Service Center. UCI Machine Learning Repository. https://doi.org/10.24432/C5GS39
[50] Lim T (1997) Contraceptive Method Choice. UCI Machine Learning Repository. https://doi.org/10.24432/C59W2D
[51] Dua D, Graff C (2020) HCV data. UCI Machine Learning Repository. https://archive-beta.ics.uci.edu/dataset/571/hcv+data
[52] Malerba D (1995) Page Blocks Classification. UCI Machine Learning Repository. https://doi.org/10.24432/C5J590
[53] Nakai K (1996) Ecoli. UCI Machine Learning Repository. https://doi.org/10.24432/C5388M
[54] Campos D, Bernardes J (2010) Cardiotocography. UCI Machine Learning Repository. https://doi.org/10.24432/C51S4N
[55] Siegler R (1994) Balance Scale. UCI Machine Learning Repository. https://doi.org/10.24432/C5488X
[56] Cattral R, Oppacher F (2007) Poker Hand. UCI Machine Learning Repository. https://doi.org/10.24432/C5KW38