IRS-BAG-Integrated Radius-SMOTE Algorithm with Bagging Ensemble Learning Model for Imbalanced Data Set Classification

By Angga Pradipta

WORD COUNT	10649	TIME SUBMITTED	24-OCT-2023 05:15PM
		PAPER ID	103693183

Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 7, No. 5, October, 2023

IRS-BAG-Integrated Radius-SMOTE Algorithm with Bagging Ensemble Learning Model for Imbalanced Data Set Classification

Lilis Yuningsih [1], Gede Angga Pradipta [2*], Dadang Hermawan [3], Putu Desiana Wulaning Ayu [2], Dandy Pramana Hostiadi [2], Roy Rudolf Huizen [2]

[1] *Department of Information System, Faculty Computer and Informatics, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar 80234, Indonesia.*

[2] *Post Graduate Department of Information System, Faculty Computer and Informatics, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar 80234, Indonesia.*

[3] *Department of Digital Bussines, Faculty Bussines and Vocation, Institut Teknologi dan Bisnis STIKOM Bali Denpasar 80234, Indonesia.*

Abstract

Imbalanced learning problems are a challenge faced by classifiers when data samples have an unbalanced distribution among classes. The Synthetic Minority Over-Sampling Technique (SMOTE) is one of the most well-known data pre-processing methods. Problems that arise when oversampling with SMOTE are the phenomenon of noise, small disjunct samples, and overfitting due to a high imbalance ratio in a dataset. A high level of imbalance ratio and low variance conditions cause the results of synthetic data generation to be collected in narrow areas and conflicting regions among classes and make them susceptible to overfitting during the learning process by machine learning methods. Therefore, this research proposes a combination between Radius-SMOTE and Bagging Algorithm called the IRS-BAG Model. For each sub-sample generated by bootstrapping, oversampling was done using Radius SMOTE. Oversampling on the sub-sample was likely to overcome overfitting problems that might occur. Experiments were carried out by comparing the performance of the IRS-BAG model with various previous oversampling methods using the imbalanced public dataset. The experiment results using three different classifiers proved that all classifiers had gained a notable improvement when combined with the proposed IRS-BAG model compared with the previous state-of-the-art oversampling methods.

Keywords:
Imbalanced Data;
Oversampling;
SMOTE; Bagging;
Classification;
Machine Learning.

Article History:

Received:	02	April	2023
Revised:	08	August	2023
Accepted:	17	August	2023
Published:	01	October	2023

1- Introduction

Unequal class distributions in any dataset are technically called imbalances. However, a dataset is considered imbalanced when there is a significant difference in the disproportion between the numbers of examples in each class. In other words, in a class imbalance problem, one or more classes (i.e., the minority class) have very few cases, while another class (i.e., the majority class) has many cases. Hence, one or more classes may be underrepresented in the dataset. In machine learning models, this imbalanced data set condition causes the classifier to recognize data more easily in the majority class, and it is difficult to recognize the minority class because the amount of training data is small [1]. Therefore, additional steps are needed to overcome this condition at the data level, algorithmic level, and ensemble learning. Algorithm-level (internal) approaches create or modify existing algorithms to take into account the significance of positive examples [2-4]. Ensemble learning combines several classifiers into one and improves accuracy compared to using a single classifier. However, ensemble learning techniques alone cannot solve the class imbalance problem. Thus, to deal with the problem in question, ensemble learning algorithms need to be specifically adapted. It is usually done by

* **CONTACT**: angga_pradipta@stikom-bali.ac.id

DOI: http://dx.doi.org/10.28991/ESJ-2023-07-05-04

combining an ensemble learning strategy with any methods presented in the previous chapters to deal with the class imbalance, such as data-level preprocessing methods or cost-sensitive learning.

Several studies have applied ensemble learning methods to improve performance in terms of accuracy. Petinrin et al. [5] performed bioactive molecule prediction using the Majority Voting method. The research conducted several experiments involving combinations of single classifiers to determine the highest accuracy, such as SVM, Naïve Bayes, Decision Tree, KNN, and Random Forest. The voting method used was Majority Voting with equal weight among all classifiers. The results showed a combination of SVM, DT, KNN, and RF with an accuracy of 97.1%. Similar to the research by Smith and Martinez [6], where each classifier had the same weight in calculating Majority Voting results. This research compared the results between filtering methods (biased filter and adaptive filter) + single classifier and Majority Voting for a dataset from the UCI machine learning repository with imbalanced dataset characteristics. In the Majority Voting model, the dataset was divided into smaller sub-data, and each was assigned a classifier to predict the class label of the data. The classification methods used were Multilayer Perceptron dengan Back Propagation (MLP), Decision Tree (C4.5), Locally Weighted Learning (LWL), 5-Nearest Neighbors (5-NN), Nearest Neighbor with Generalization (NNge), Naïve Bayes, Ripple Down Rule Learner (RIDOR), Random Forest (RandForest), and RIPPER. The testing results showed that the Majority Voting method improved accuracy more than the data filtering method. In addition, the Majority Voting method was more effective in computing training data than data filtering.

Onan et al. [7] used Multi-objective Differential Evolution to weigh each classifier. Weighting was done by measuring the results of class prediction (a single classifier) towards an instance. The F1 measure, precision, and recall parameters on each classifier were used as a reference to determine the weight of each classifier toward the final voting decision. Testing was conducted on public data sentiment analysis and showed results where MODE-Based Weighted Voting obtained higher accuracy than other Stacking and Weighted Voting methods. The use of weights based on F1 measure values was also carried out by Bashir et al. [8] by utilizing the Majority Voting method combined with Bagging (BagMOOV) to predict and analyze heart disease. The classification methods used in this research were Naive Bayes, Linear Regression, Quadratic Discriminant Analysis, Instance-Based Learner, and SVM, and weighting was done using Multi-Objective Optimization. The F1 measure value was used as an objective function to measure the weight of each classifier in producing the final decision in predicting a class of data. This proposed model produced an accuracy value of 84.16%, a sensitivity of 93.26%, a specificity of 96.70%, and an F-measure of 82.15%.

Furthermore, handling the imbalanced data problem requires data-level processing. Data-level processing is a mechanism to address the problem of imbalanced learning using sampling methods. Training instances are modified in such a way as to produce a more balanced class distribution that allows classifiers to perform similarly to standard classification. One of the most frequently used data oversampling methods in the data-level approach is the Synthetic Minority Oversampling Technique (SMOTE) method. The SMOTE method proposed by Chawla et al. [9] was an algorithm that performed oversampling of the minority class by taking several random samples from that class and creating synthetic data along the interpolation line from the sample data to the nearest minority data point with as many neighbors as k. The amount of synthetic data could be determined with a predetermined sampling rate parameter. The advantage of the SMOTE oversampling method compared to the random oversampling method was its ability to produce synthetic data that did not cause overfitting in the classifier. Even though SMOTE achieved a better distribution of synthetic data than random oversampling, when used on data distribution with a reasonably high level of variance, SMOTE could obtain not as good results as they should be or might even be counterproductive in many cases, especially at very high imbalance ratios, causing overfitting and conflict region between class labels. It was because SMOTE presents several drawbacks related to blind oversampling, where the generation of new (minority) positive samples only considered proximity and did not observe the surrounding area of the new synthetic data to be generated.

However, in the last two decades, the development of the SMOTE method has evolved to improve some of the weaknesses of the initial SMOTE method. Research conducted by Fernández et al. [10] revealed many challenges and deficiencies in the SMOTE method, namely the presence of overlapping, small disjuncts, and noise when creating new synthetic data. The conditions above made it difficult for the classifier to find a decision boundary and added complexity to finding the optimal solution. Research conducted by Fernandez et al. [10] summarized that modifications to the development of the SMOTE method were generally divided into several approaches, namely the initial sample, integrating with undersampling, type of interpolation, operation with dimensionality changes, adaptive generation, relabeling, and filtering noise. One approach to modifying the SMOTE method was to consider the feasibility of the initial sample from the synthetic data generation process. Incorrect samples would make the distribution of data more complicated after the oversampling process was complete [11–13]. Based on the SMOTE method, research by Bunkhumpornpat et al. [12] proposed a model named Safe-Level SMOTE. This method ensured that each data point was in the safe category before the oversampling process was carried out using SMOTE.

Each synthetic data point would be placed close to the safe-level area so that all new data would be created only in safe-level regions. Research conducted by Maciejewski & Stefanowski [14] tries to fix some of the weaknesses in Safe Level SMOTE. Modifying how to calculate the safe level ratio for majority-class neighbors was done by considering the value of the local neighborhood in selecting sample data. The selection of initial sample data by grouping data based on

local characteristics and adding cleaning steps with the Roughset method has been carried out by Borowska & Stepaniuk [15]. This research also used modified versatile SMOTE, which put data in three processing modes: no-safe, high-complexity, and low-complexity. The k-farthest neighbors approach to determining the neighbors of the selected sample data was proposed by Gosain & Sardana [16]. This concept contrasts with most SMOTE development methods, which seek the closest distance value from the data neighbor. This research assumed that using the farthest neighbors would make the decision area in the minority class wider to facilitate the learning process. The Diversity and Separable Metrics in Over-Sampling Technique (DSMOTE) model was proposed by Mahmoudi et al. [17]. The main idea of this research was to use diversity and separable measure information to determine sample data points in the minority class. The measurement of the diversity level combined the Euclidean distance value and the geometric mean value of the sample data [18]. In addition, the type of SMOTE modification was carried out based on the amount of synthetic data generation based on the difficulty level of the sample data to be studied by the classifier.

This approach was pioneered by research conducted by He et al. [19], who proposed the ADYSN (Adaptive Synthetic Sampling Approach) method. Making synthetic data based on the difficulty level of the sample data to be studied. The more the Nearest Neighbors of the sample data were determined to be the majority class, the more new synthetic data were created in that area to strengthen minority data. Research [20] proposed the SMOTE-D method. This method performs the dynamic creation of new synthetic data for each sample in the minority class. The amount of synthetic data depends on the distance of the sample data to the k-Nearest Neighbors (k=5). The farther the sample data was from the nearest neighbors, the more synthetic data would be generated. Data distribution calculation using the standard deviation of the sample data distance to each k-Nearest Neighbors. According to research [13], misclassified data was prone to occur in the borderline area between classes. Then the BORDERLINE-SMOTE method was proposed, which detected the minority class sample points that fall into the DANGER category because they were in the borderline area. Synthetic data was made only in the border area so that the classifier was expected could improve the ability to determine the decision boundary between classes in the dataset. The next approach was to combine SMOTE with various noise detection algorithms, such as Local Outlier Factor (LOF) [21], Rough Set Theory [15, 22–25], and Iterative-Partitioning Filter (IPF) [26]. Outlier detection was carried out in the same way to ensure that the sample data selected in the oversampling process was quality data and not in the noise category.

Considering noise or outlier removal in datasets that have the risk of reducing the meaning of the data, especially in data on real-world problem cases, such as medical diagnoses, Pradipta et al. [27] proposed a modified SMOTE algorithm called R-SMOTE. After oversampling, the overlapping region problem in the dataset was the main focus to solve in this R-SMOTE algorithm. Overlapping regions occurred because of interpolation between the minority sample points and the nearest minority data points in different class regions. The higher the level of overlap, the more difficult it was for the classifier to find the boundary line for each label in the dataset. R-SMOTE did not remove the majority of the data, which was classified as noise, but focused more on avoiding the occurrence of interpolation towards the noise data point. A new synthetic sample was created with a safe radius boundary, namely the boundary with the closest majority data. Each piece of data was interpolated in one direction and to all areas within that radius. The R-SMOTE method had been applied to medical case data, namely the umbilical cord [28, 29] and electroencephalography [30].

However, the R-SMOTE method still has weaknesses, namely on datasets with an amount of noise and small disjunct spread over several data distribution areas. This condition causes a decrease in the performance level of the machine learning algorithm. Small disjunct is a condition where noise forms a set of small clusters scattered in the dataset. In addition, in R-SMOTE, overfit conditions may occur when the synthetic data generated is extensive and relatively close to one another. The small safety radius distance causes this closeness in one data generation cycle. In addition, when directly applied to imbalanced datasets, ensemble learning methods do not solve the underlying problem in the classifiers themselves with skewed class distributions. For this reason, they need to be combined with other techniques to tackle class imbalance problems. Based on this problem, the main contribution of this research was to create a synthetic data formation model by dividing the primary dataset into several parts or called Bag. The proposed model was a combination of two approaches: data-level processing and ensemble learning. The R-SMOTE method was modified by working on sub-datasets so that each classifier would be supplied with training data based on the oversampling results in each sub-dataset. This modification process combined the R-SMOTE method into the Bagging algorithm. With the sub-dataset that has been oversampled using R-SMOTE, each base classifier would easier understand the data characteristics even though there were some noises and small disjuncts in the dataset. The majority voting method was applied to each base classifier formed on each oversampled subset data.

2- Integrated Radius-SMOTE with Bagging Algorithm (IRS-BAG)

The proposed IRS-BAG model was a development model of the SMOTE [9] and Radius-SMOTE [27] methods to overcome and withstand imbalanced data conditions and the existence of noise data. In the SMOTE or Radius-SMOTE algorithms, the process of creating synthetic data was carried out on the entire dataset as a whole. This concept was still very susceptible to overfitting when learning was carried out by the classifier. In data samples between the minority and majority instances that were very close together, the Radius-SMOTE method would accumulate new data in a smaller area where this caused a higher overfit risk. Figure 1. illustrates the differences in the data distribution resulting from oversampling using the SMOTE, Radius-SMOTE, and IRS-BAG SMOTE methods. From the illustration, the IRS-BAG

model oversamples the new sub-data sample created from the bootstrapping process. Synthetic data did not seem to overlap with other regions, and the data distribution did not accumulate in very small areas. It was indeed able to improve the performance of the classification process by each base classifier. Figure 2 shows the workflow of the process IRS-Bag Model. The first step in the IRS-BAG method was dividing the original dataset into several sub-data using the bootstrapping technique, which was part of the bagging method. This process produced several sub-sample datasets where the data inside was part of the primary dataset. The next process was oversampling the minority data contained in each sub-sample dataset.

The Radius-SMOTE method was one of the development methods from SMOTE, which focused on improving the distribution of synthetic data to minimize overlapping regions between classes and reduce the creation of new data noise. The condition of overlapping regions and the existence of noise data greatly defected the ability of the base classifier to determine patterns and decision boundaries for each class in the learning process. The Radius-SMOTE method essentially limited the area where synthetic data was generated by using a safe radius parameter taken from the distance from the minority sample point to the nearest majority data point. Furthermore, each sub-sample dataset resulting from oversampling was used as input in the learning process of each base classifier. In this research, the number of performance evaluations was carried out on four base classifiers, namely Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree (C.45). The decision-making for the final class output was carried out by the averaging voting method where the weight value for decision making was the same between each. The final class would be the output class with the highest voting value.

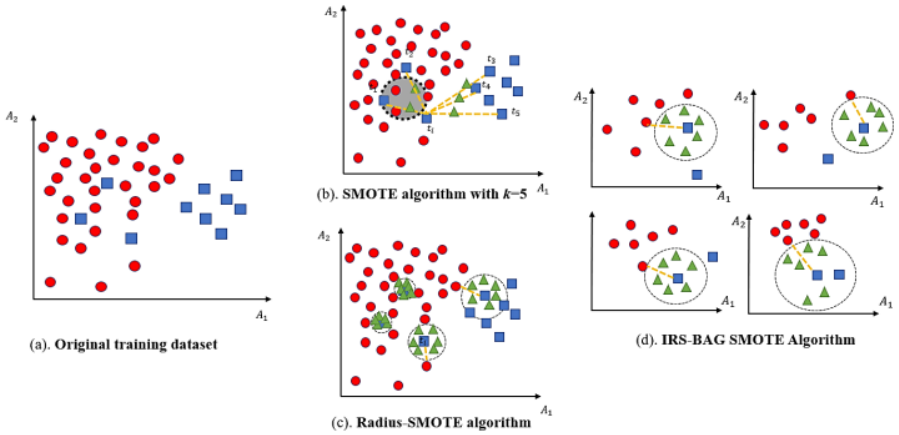

Figure 1. Illustration of sample distribution on Oversampling SMOTE, Radius SMOTE, and IRS-BAG SMOTE Algorithms

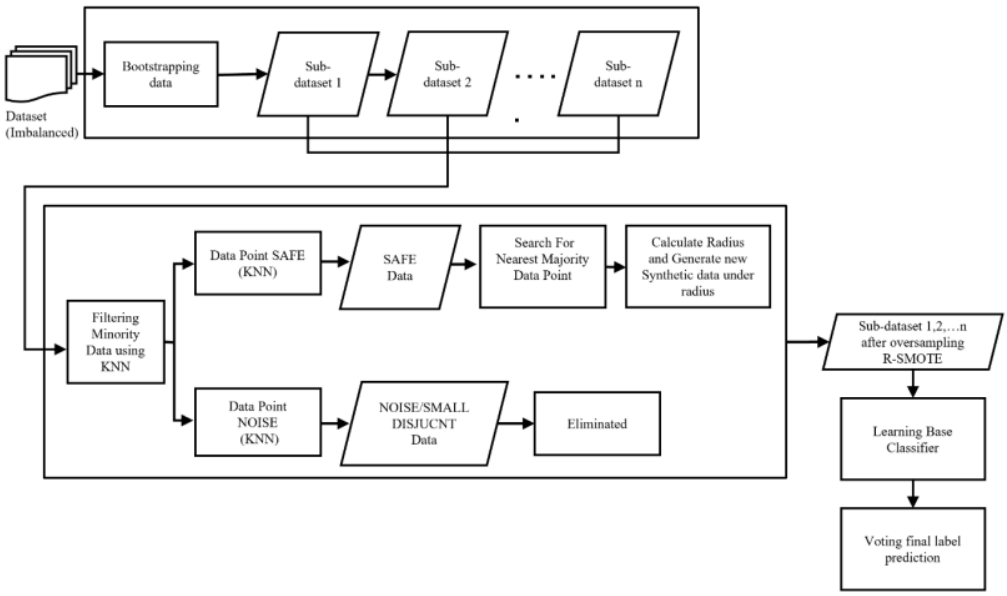

Figure 2. Workflow of IRS-BAG SMOTE Algorithm

The detailed algorithm flows are shown with the pseudocode in Table 1. In general, the stages of this algorithm consisted of bootstrapping, oversampling Radius-SMOTE, and final label voting. In the bootstrapping process for each instance in the training dataset S determined the number of data subsets J to be formed. In each training data subset S_t filled the data with random sample replacement taken from dataset S according to the bootstrap size n. The following process created synthetic data using the Radius-SMOTE method on each S_t sub-dataset. In each minority data, P_{subnum} in the sub-dataset calculated the Euclidian distance to all minority data and classified it based on the number of nearest neighbors set. If the classification result of the P_{subnum} was minority data, it was categorized into $P_{subSAFE}$. Conversely, if the result was a majority category, it was stored in $P_{subDANGER}$. Data sampling was taken randomly for Radius-SMOTE oversampling on the safe data category stored in $P_{subSAFE}$. After the sample data was determined, calculated its distance to all the data in the train data subset. Furthermore, find the smallest distance value from the sample data to the majority data in the training subset. Then the closest distance result *X_dist* was used as a circle-safe radius to form new synthetic data. The last step after the oversampling process was the learning process of each base classifier on each sub-dataset. When the model had been formed, the voting method from each base classifier was used to determine the final class label of the testing data.

Table 1. IRS-BAG Algorithm

Input: S: Training Set ; N : Majority instances; P : Minority instance; N_{sub} : Majority instances in subset data ; P_{sub} : Minority instance in subset data; T: Number of iterations; n: Bootstrap size; I: Weak learner; P_{subnum}: number of minority instance in subset data; N_{subnum}: number of majority instance in subset data ; J : Number of subset data; I: Weak learner; S_t: *Subset trainning set*

Output: final label prediction

Bootstrap sampling

```
for s = 1 to
    for t = 1 to T do
    S_t ← RandomSampleReplacemnet (n, S)
```

Oversampling Radius-SMOTE for Each Subset Data

```
for t =1 to P_subnum
    Compute KNN algorithm
    Classifying  P_sub
    P_subSAFE  ← P_sub classified by KNN as Minority Instances
    P_subDANGER ← P_sub classifier by KNN as Majority Instances
    P_subSAFESAMPLE ← randomly select form P_subSAFE
end for
for 1 to N_subnum
    N_subdist ← Compute distance form all majority instances to P_subSAFESAMPLE using
Euclidean distance.
    N_subMIN ← Majority instance with minimum distances from P_subSAFESAMPLE
end for
X_dist ← distance N_subMIN to P_subSAFESAMPLE
```

Calculate new synthesize instance with under radius distance *X_dist*

```
P_subNEW =  P_subSAFESAMPLE + (rand (0,1) × X_dist)
P_subNEW  = P_sub+P_subNEW
End for
```

Learning on Subset Data

```
for k= 1,….,I
    h_t= I(. P_subNEW)  # train a base learner I from Subset dataset after
oversampling Radius-SMOTE
add h_t to the ensemble, ε ← ε ∪ h_t
End
```

Voting

```
Ensemble Combination: Simple Majority Voting - Given unlabeled instance x
Evaluate the ensemble ε = {h_1,……h_t} on x
Let V_t,c = 1 if h_t chooses class ω_c , and 0, otherwise.
Obtain the total vote received by each class.
```

$$V_c = \sum_{t=1}^{T} V_{t,c}, c = 1, \ldots . C$$

Output: Class with highest V_c

The following explains each step of the IRS-BAG process: bagging, oversampling Radius-SMOTE, and Voting Classification.

2-1-Radius SMOTE Algorithm

If analyzed, the problems of overlapping, noise, and small disjunct were the result of selecting random sample data for minority class data. Noise data in the minority class had the risk of producing new noise data, which caused regional conflicts between each class. The Radius-SMOTE was inspired by the work carried out by Han et al. [13] and Bunkhumpornpat et al. [12], which conducted filtering on the sample data so that the selection of data samples was not done randomly. It ensured the data was created with the right sample selection. This proposed modification of the SMOTE model began by dividing the data points in the minority class into three categories, namely SAFE, NOISE, and SMALL DISJUNCT. Data selection or filtering was carried out using the K-NN method based on the data's location and the data's neighbors to other classes. Each minority dataset would be selected with a parameter value of k in the KNN method, which was set to 5. Then, minority data correctly classified as minority data by the KNN method became data in the SAFE category. Conversely, minority data classified as majority would be those with the NOISE/SMALL DISJUNCT category.

The next step after the sample data was divided into two categories, namely SAFE and NOISE, where was the process of creating new synthetic data. Making new synthetic data was done on data with the SAFE category. As in the SMOTE method, synthetic data was made by identifying the closest minority data points and drawing interpolation lines between them. Determining the number of closest data points in the SMOTE method used the KNN method approach with parameter k as the number of closest data points. As previously explained, using the k parameter was very risky to produce new synthetic data, resulting in overlapping between the minority class and the majority class. Therefore, this research proposed to use the radius parameter. The radius was obtained by finding the closest distance to the majority data point from the sample and using it as the radius value. All new data points were created within that radius.

For data formation within the radius, the circle equation was used as in Equation 1, with an example of a two-dimensional vector.

$$\left\|\underset{\sim}{b} - \underset{\sim}{p}\right\| \le r^2$$
$$\left\|\underset{\sim}{a} - \underset{\sim}{p}\right\| \le r^2 \quad (1)$$

$$\sum_{i=1}^{n}\left(b_{ij} - P_{ij}\right)^2 \le r^2 \quad (2)$$

$$r^2 = \sum_{j=1}^{n}\left(p_j - t_j\right)^2 \quad (3)$$

where p a is the center point of the circle (minority sample point) with $(p_1, p_2, p_3, \ldots p_n)$, and t $(t_1, t_2, t_3, \ldots t_n)$, is the majority point closest to the center of the circle, b_i is the new data point under the radius with $(b_1, b_2, b_3, \ldots b_n)$ with i =1…n then r^2 is the distance between p with t as in Equation 3. An illustration of this proposed model can be seen in Figure 1. Then each minority sample is calculated its distance from the majority class. Calculation of the distance using the Euclidean distance method. The nearest majority data point is the one that has the minimum distance to the overall distance to the minority data points as in Equation 4.

$$r_{ij} = min \sum_{i=1}^{n}\sum_{j=1}^{n}\sqrt{\left(p_j - t_i\right)^2} \quad (4)$$

where r_{ij} is the smallest distance between the minority data to j against the majority data to i. After the majority data points are found, synthetic data formation is carried out on the interpolation of the two points. The formation of synthetic data is carried out in two directions of the line, namely r_{ij} and $-r_{ij}$ with the Equations on 5 and 6.

$$aij = p_j + (rand(0,1) \times (r_{ij} - p_j)) \quad (5)$$

$$bij = p_j + (rand(0,1) \times (p_j - r_{ij})) \quad (6)$$

Limiting the area of creating this new data reduced the occurrence of overlapping data, as happened in the SMOTE method. In this research, Radius-SMOTE was not directly implemented on the entire training dataset but on the sub-dataset resulting from the bootstrapping process, as shown in Figure 3. A dataset with a high imbalance ratio and a low variance value was at risk of overfitting due to data crowding in a narrow area. In addition, one way to improve classifier performance was with the ensemble learning technique, which involved multiple classifiers trained on different subsets of training data.

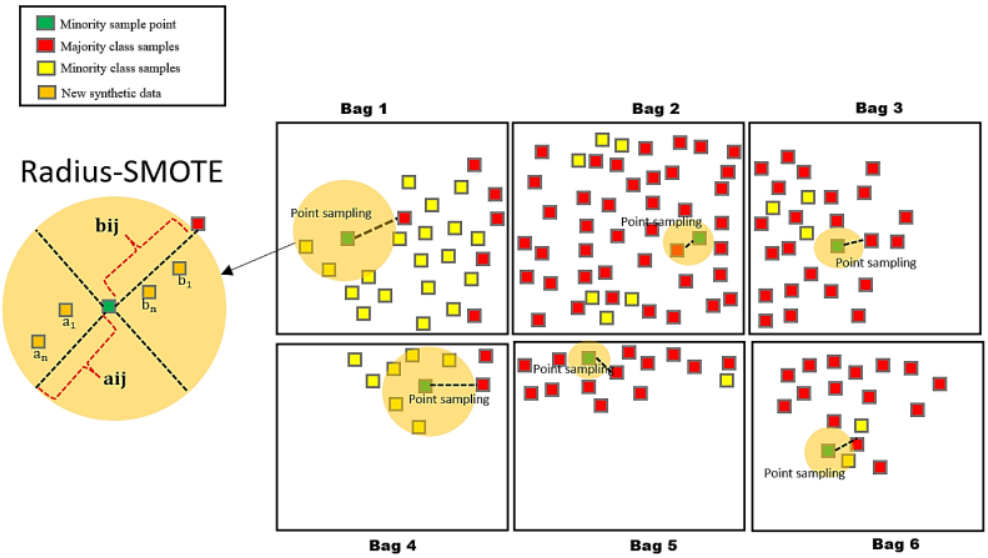

Figure 3. Radius-SMOTE in every bag/subset training set

2-2-Bagging Algorithm

Breiman [31] introduced a concept called bootstrap aggregating to form an ensemble model. This learning ensemble was formed from the results of training several classifiers with different bootstrapped replicas of the original training dataset. Therefore, a subsample dataset was formed by randomly drawing with replacement instances from the original dataset to conduct training for each base classifier. The most usual practice was maintaining the original dataset size, meaning that approximately 63.2% of the instances would be presented in each bag (with some of them appearing more than once). Given a training dataset S of cardinality N, bagging simply trains T independent classifiers, each trained by sampling, with replacement, N instances from S. Because each model was formed from different sample data, the resulting decision trees would be different. Therefore, decision-making on bagging applied the principle of aggregating, namely voting on the classification results of the entire model. Bagging diversity was obtained from the resampling procedure by training each classifier with different data subsets. The assuming base classifier used was of a weak type, the resulting model should differ due to the changes in the data. Weighted majority voting was usually used by using confidence given by each classifier in the prediction. One of the advantages of this bagging method was its simplicity and reduced variance since the effect of voting was similar to that of averaging regression, where the overfitting reduction became easier to observe. The pseudo-code of the bagging algorithm is shown in Table 2.

Table 2. Bagging Algorithm

Input: Training data S; supervised learning algorithm, BaseClassifier, integer T specifying ensemble size; percent R to create bootstrapped trainning data.

Do $t = 1, \ldots, T$

1. Take a bootstrapped replica S_t by randomly drawing R % of S.
2. Call BaseClassifier with S_t and receive the hypothesis (classifer) h_t.
3. Add h_t to the ensemble, $\varepsilon \leftarrow \varepsilon \cup h_t$.

End

Ensemble Combination: Simple Majority Voting – Given unlabeled instance x

20. Evaluate the ensemble $\varepsilon = \{h_1, \ldots \ldots h_t\}$ on x
21. Let $V_{t,c} = 1$ if h_t chooses class ω_c , and 0, otherwise.
22. Obtain total vote received by each class

$$V_c = \sum_{t=1}^{T} V_{t,c}, c = 1, \ldots . . C$$

Output: Class with highest V_c

3- Experimental Setting

3-1-Experimental Data

This section describes the characteristics of the acquired datasets used in this research. The experiments were carried out using 13 different imbalanced datasets from different application areas on binary and multiclass classification problems. The dataset used had a different number of features and a different imbalance ratio. Dataset obtained from UCI Machine Learning Repository and KEEL Repository. Table 3 shows the characteristics of the dataset used in the

experiment. In the table, there was information on the imbalanced ratio value, which represented the value of the ratio between negative and positive classes, the number of features in each dataset, the number of data or instances in each dataset, the number of comparisons of positive and negative instances in percent size. The dataset was selected by considering the imbalanced ratio values from relatively low to high.

Table 3. Dataset Characteristic for Experimentation

No	Name	Imbalanced Ratio (IR)	Features	Instances	Positive Instances (%)	Negative Instances (%)
1	03subcl5-600-5-70-BI	5	2	600	16.67	83.30
2	04clover5z-600-5-70-BI	5	2	600	16.67	83.30
3	ecoli-0-1-3-7_vs_2-6	39.14	7	281	2.49	97.51
4	glass1	1.82	9	214	35.46	64.54
5	new thyroid	4.84	5	215	17.12	82.88
6	paw02a-600-5-70-BI	5	2	600	16.67	83.30
7	wine	1.5	13	178	40.00	60.00
8	yeast-1-4-5-8_vs_7	22.10	8	693	4.330	95.67
9	Umbilical Cord	18.87	5	151	5.300	94.70
10	Breast	2.36	9	286	29.12	70.38
11	Haberman	2.78	3	306	26.39	73.61
12	Pima	1.87	8	768	34.86	65.14
13	Bupa	1.38	6	345	42.19	57.81

3-2-Evaluation Metrics

The performance of the classification model would be tested using three metrics, including accuracy, precision, recall, and F-Measure. In machine learning classification tasks, confusion matrix parameters which were True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), were the main parameters from which other performance metrics, such as Precision, Recall and F1 scores were computed. Accuracy measured the amount of data correctly classified according to the ground truth label divided by the total data used for testing. Precision was the rate of correct predictions among all samples predicted to belong to the minority class. It indicated how many of the positive predictions were correct, whereas recall means the proportion of minority class samples labeled as positive. Table 4 shows formulas for measuring accuracy, precision, recall, and F-Measure.

Table 4. Performance Metrics

No	Metrics	Expression
1	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
2	Precision	$\frac{TP}{TP+FP}$
3	Recall	$\frac{TP}{TP+FN}$
4	F-Measure	$F_\beta = \frac{1}{\beta \times \frac{1}{precision} + (1-\beta) \times \frac{1}{recall}}$

Then the F-measure parameter was the *harmonic mean* of precision and recall. β was the value of the F-measure ranging from 0 to 1. The greater the value β then the testing model prioritized the results of precision and vice versa.

4- Experimental Result and Analysis

4-1-Experiment on Combination of Different Base Classifier and SMOTE variant

The IRS-BAG model was tested with the first scheme on three types of base classifiers or estimators, namely support vector machines (SVM), Decision trees, and K-Nearest Neighbors (KNN), which were also combined with several states of art from the development of the SMOTE method. Their default settings in Python libraries had used all of these classifiers, and none of their hyperparameters had been optimized. The SMOTE variants used were ADAYSN-SMOTE [19], Tomek-link [32], SMOTE-IPF [26], Borderline-SMOTE [13], and Safe-level SMOTE [12]. The K-fold cross validation method was used in each classifier with the specified number of folds being 10.

Table 5 shows the experimental results on IRS-BAG with the SVM base classifier. The performance shown by the proposed model was marked in bold font with the highest value on each dataset. The proposed model performed was measured based on accuracy, precision, recall, and F-Measure for each dataset used. The proposed method using IRS-BAG outperforms in 7 out of 11 datasets compared to other SMOTE methods, namely in 04clover5z-600-5-70-BI, ecoli-0-1-3-7_vs_2-6, breast, new thyroid, Pima, Umbilical Cord, and yeast-1-4-5-8_vs_7 datasets. These results showed an increase in performance for the accuracy of the bagging method with the proposed method (IRS-BAG) on the 04clover5z-600-5-70-BI dataset of 10%. Then, there was an increase of 10% in the new thyroid dataset and 0.6% in the Pima and Umbilical Cord datasets. The increase in performance in terms of precision on the 04clover5z-600-5-70-BI dataset by 24%, on the ecoli-0-1-3-7_vs_2-6 dataset by 50%, on the Pima dataset by 12%, and on the Umbilical Cord dataset by 0.5%. While the performance improvement in terms of recall on the 03subcl5-600-5-70-BI and 04clover5z-600-5-70-BI datasets by 30%, on the ecoli-0-1-3-7_vs_2-6 dataset by 50%, and on the Umbilical Cord and yeast datasets by 2%.

Table 5. IRS-BAG Classification result using SVM based classifier compare with varians of previous SMOTE methods

Metrics	Dataset	Bagging	Bagging+ SMOTE [9]	Bagging + ADAYSN [19]	Bagging + IPF [26]	Bagging + TomekLink [32]	Bagging + Borderline [13]	Bagging + Safe Level [12]	IRS-BAG
Accuracy	03subcl5-600-5-70-BI	0.82	0.8	0.76	0.79	0.83	0.78	0.78	**0.84**
	04clover5z-600-5-70-BI	0.81	0.87	0.87	0.87	0.91	0.89	0.82	**0.91**
	bupa	0.71	0.76	0.76	0.52	**0.79**	0.69	0.57	0.76
	ecoli-0-1-3-7_vs_2-6	0.97	0.99	**0.99**	0.96	**0.99**	**0.99**	0.97	**0.99**
	glass1	0.74	0.77	**0.84**	0.46	0.5	0.46	0.57	0.55
	haberman	**0.81**	0.71	0.6	0.65	0.61	0.57	0.67	0.71
	breast	**0.71**	0.61	0.52	0.61	0.63	0.59	0.62	**0.71**
	new thyroid	0.81	0.87	0.87	**0.91**	0.86	**0.91**	0.79	**0.91**
	paw02a-600-5-70-BI	0.85	0.88	0.91	0.89	**0.93**	0.91	0.84	0.88
	pima	0.81	0.83	0.82	0.86	0.82	0.82	0.75	**0.87**
	Umbilical Cord	0.91	0.92	0.91	0.87	0.91	0.96	0.93	**0.97**
	wine	0.87	**0.97**	0.9	0.91	0.89	0.91	0.87	0.92
	yeast-1-4-5-8_vs_7	0.89	0.81	0.79	86	0.87	**0.91**	0.81	**0.91**
Precision	03subcl5-600-5-70-BI	0.41	0.85	0.8	0.84	**0.87**	0.82	0.81	**0.87**
	04clover5z-600-5-70-BI	0.67	0.87	0.87	0.87	**0.91**	0.89	0.82	**0.91**
	bupa	0.71	**0.77**	**0.77**	0.32	0.8	0.7	0.6	0.69
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	0.99	0.99	0.49	**0.99**
	glass1	0.74	0.77	**0.85**	0.23	0.5	0.23	0.54	0.47
	haberman	**0.90**	0.73	0.6	0.68	0.65	0.59	0.69	0.72
	breast	0.68	0.64	0.56	0.63	0.65	0.62	0.69	**0.71**
	new thyroid	**0.93**	0.91	0.89	**0.93**	0.91	0.92	0.92	0.92
	paw02a-600-5-70-BI	0.75	0.88	0.91	0.89	**0.93**	0.92	0.84	0.89
	pima	0.77	0.83	0.82	0.86	0.81	0.83	0.75	**0.89**
	Umbilical Cord	0.9	0.92	0.93	0.82	0.91	0.93	0.91	**0.95**
	wine	0.85	**0.95**	0.89	0.89	0.89	0.87	0.83	0.91
	yeast-1-4-5-8_vs_7	0.86	0.79	0.77	0.84	0.87	0.85	0.82	**0.89**
Recall	03subcl5-600-5-70-BI	0.50	0.8	0.76	0.79	0.83	0.79	0.77	**0.84**
	04clover5z-600-5-70-BI	0.64	0.87	0.87	0.87	0.91	0.89	0.82	**0.90**
	bupa	0.71	0.76	0.76	0.37	0.8	0.7	0.58	0.56
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.49	**0.99**
	glass1	0.7	0.77	**0.83**	0.5	0.5	0.5	0.5	0.5
	haberman	0.53	0.69	0.6	0.64	0.59	0.58	0.64	**0.74**
	breast	0.65	0.61	0.54	0.62	0.62	0.58	0.62	0.54
	new thyroid	0.59	0.86	0.87	**0.91**	0.85	**0.91**	0.57	**0.91**
	paw02a-600-5-70-BI	0.75	0.88	0.91	0.89	**0.93**	0.92	0.84	0.88
	pima	0.77	0.83	0.82	0.86	0.81	0.82	0.75	**0.84**
	Umbilical Cord	0.92	0.91	0.91	0.79	0.89	0.93	0.91	**0.94**
	wine	0.84	**0.93**	0.88	0.88	0.89	0.87	0.83	0.89
	yeast-1-4-5-8_vs_7	0.85	0.78	0.74	0.82	0.87	0.85	0.82	**0.87**

	03subcl5-600-5-70-BI	0.45	0.79	0.75	0.78	0.82	0.78	0.77	**0.84**
	04clover5z-600-5-70-BI	0.65	0.87	0.87	0.87	0.91	0.89	0.82	**0.9**
	bupa	0.71	0.76	0.76	0.34	0.79	0.69	0.55	0.55
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.49	**0.99**
	glass1	0.71	0.77	**0.83**	0.32	0.5	0.32	0.36	0.5
	haberman	0.5	0.68	0.6	0.62	0.56	0.56	0.63	**0.71**
F-Measure	breast	**0.65**	0.58	0.48	0.59	0.61	0.55	0.59	0.5
	new thyroid	0.66	0.87	0.87	**0.91**	0.86	**0.91**	0.63	**0.91**
	paw02a-600-5-70-BI	0.75	0.88	0.91	0.91	**0.93**	0.92	0.84	0.89
	pima	0.77	0.83	0.82	**0.86**	0.81	82	0.75	0.85
	Umbilical Cord	0.92	0.91	0.91	0.79	0.89	0.93	0.91	**0.94**
	wine	0.84	**0.93**	0.88	0.88	0.89	0.87	0.83	0.89
	yeast-1-4-5-8_vs_7	0.85	0.78	0.74	0.82	**0.87**	0.85	0.82	**0.87**

Then the following experiment was to test the Decision Tree C.45 as a classifier. Table 6 shows the experimental results with accuracy, precision, recall, and F-Measure parameters. The IRS-BAG method produced the best performance on nine datasets compared to other methods. In the Bagging method with a based classifier using a Decision Tree, the use of oversampling increased the resulting performance. The proposed IRS-BAG method was able to dominate with superiority in the nine datasets used compared to the previous methods.

Table 6. IRS-BAG classification result using Decision Tree based classifier compare with variants of previous SMOTE Methods

Metric	Dataset	Bagging	Bagging+ SMOTE [9]	Bagging + ADAYSN [19]	Bagging + IPF [26]	Bagging + TomekLink [32]	Bagging + Borderline [13]	Bagging + Safe Level [12]	IRS-BAG
	03subcl5-600-5-70-BI	0.82	0.85	0.84	0.85	0.86	0.86	0.82	**0.89**
	04clover5z-600-5-70-BI	0.81	0.87	0.87	0.87	**0.91**	0.89	0.82	**0.91**
	bupa	0.71	0.76	0.77	0.52	0.79	0.81	0.70	**0.89**
	ecoli-0-1-3-7_vs_2-6	0.97	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.97	**0.99**
	glass1	0.74	0.77	0.84	0.78	**0.89**	0.77	0.67	0.85
	haberman	0.70	0.74	0.72	0.74	0.76	0.75	0.58	**0.83**
Accuracy	breast	0.65	0.83	0.78	0.75	0.81	0.80	0.64	**0.89**
	new thyroid	1.00	0.96	0.97	0.97	0.97	0.95	0.93	**0.98**
	paw02a-600-5-70-BI	0.85	0.88	0.91	0.89	0.93	**0.91**	0.85	0.88
	pima	0.80	0.83	0.82	0.86	**0.87**	0.82	0.75	**0.87**
	Umbilical Cord	0.98	0.97	0.99	0.99	0.97	0.99	0.92	**1.00**
	wine	0.96	**1.00**	0.96	0.94	0.99	0.96	0.94	0.95
	yeast-1-4-5-8_vs_7	0.94	0.95	0.95	0.96	0.95	0.97	0.90	**0.98**
	03subcl5-600-5-70-BI	0.69	0.86	0.84	0.87	0.87	0.86	0.82	**0.89**
	04clover5z-600-5-70-BI	0.67	0.87	0.87	0.88	**0.91**	0.89	0.82	**0.91**
	bupa	0.71	0.77	0.77	0.32	0.80	0.81	0.70	**0.91**
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.49	**0.99**
	glass1	0.74	0.77	0.85	0.80	**0.89**	0.77	0.67	0.85
	haberman	0.56	0.73	0.75	0.75	0.76	0.79	0.58	**0.84**
Precision	breast	0.60	0.83	0.79	0.76	0.81	0.80	0.64	**0.92**
	new thyroid	**1.00**	0.98	0.97	0.98	0.97	0.96	0.93	0.98
	paw02a-600-5-70-BI	0.75	0.89	**0.91**	0.89	0.93	0.92	0.84	0.88
	pima	0.77	0.83	0.82	0.86	0.87	0.83	0.75	**0.90**
	Umbilical Cord	0.66	0.99	**1.00**	**1.00**	0.99	**1.00**	0.60	**1.00**
	wine	0.95	**1.00**	0.96	0.96	0.98	0.96	0.93	0.93
	yeast-1-4-5-8_vs_7	0.48	0.95	0.97	0.97	0.95	0.98	0.86	**0.98**

	03subcl5-600-5-70-BI	0.67	0.85	0.84	0.85	0.86	0.86	0.81	**0.89**
	04clover5z-600-5-70-BI	0.67	0.87	0.87	0.87	**0.91**	0.89	0.82	**0.91**
	bupa	0.70	0.77	0.77	0.37	0.80	**0.81**	0.70	0.80
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.49	**0.99**
	glass1	0.70	0.77	0.83	0.79	**0.89**	0.77	0.67	0.85
	haberman	0.56	0.73	0.72	0.74	0.76	0.75	0.59	**0.78**
Recall	breast	0.57	0.83	0.79	0.77	0.81	0.80	0.64	**0.84**
	new thyroid	**1.00**	0.98	0.97	0.98	0.97	0.96	0.90	0.98
	paw02a-600-5-70-BI	0.75	0.89	**0.91**	0.89	0.93	**0.91**	0.84	0.89
	pima	0.77	0.83	0.82	0.86	0.87	0.82	0.75	0.84
	Umbilical Cord	0.68	0.99	**1.00**	**1.00**	0.99	**1.00**	0.65	**1.00**
	wine	0.98	**1.00**	0.96	0.96	0.98	0.96	0.95	0.94
	yeast-1-4-5-8_vs_7	0.49	0.95	0.96	0.97	0.95	**0.98**	0.87	**0.98**
	03subcl5-600-5-70-BI	0.68	0.85	0.84	0.85	0.86	0.86	0.82	**0.89**
	04clover5z-600-5-70-BI	0.65	0.87	0.87	0.87	**0.91**	0.89	0.82	**0.91**
	bupa	0.71	0.76	0.77	0.34	0.79	0.81	0.70	**0.84**
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.49	**0.99**
	glass1	0.71	0.77	0.83	0.78	**0.89**	0.77	0.67	0.85
	haberman	0.56	0.73	0.72	0.74	0.76	0.76	0.58	**0.79**
F-Measure	breast	0.57	0.83	0.78	0.76	0.81	0.80	0.64	**0.87**
	new thyroid	**1.00**	0.98	0.97	0.97	0.97	0.96	0.91	0.98
	paw02a-600-5-70-BI	0.75	0.89	**0.91**	0.89	0.93	**0.91**	0.84	0.88
	pima	0.77	0.83	0.82	0.86	**0.87**	0.82	0.75	0.86
	Umbilical Cord	0.66	0.99	**1.00**	**1.00**	0.99	**1.00**	0.62	**1.00**
	wine	0.96	**1.00**	0.96	0.96	0.98	0.96	0.94	0.94
	yeast-1-4-5-8_vs_7	0.50	0.95	0.96	0.97	0.95	0.98	0.87	**0.98**

The most significant increase using the IRS-BAG method occurred in the Bupa dataset, with an initial accuracy compared to the Bagging method by 18%, the Haberman and breast datasets by 13%, the Pima dataset by 7%, the paw02a-600-5-70-BI, and yeast-1-4-5-8_vs_7 datasets by 4%, and in the Umbilical Cord dataset by 2%. Performance improvement in precision on 03subcl5-600-5-70-BI and 04clover5z-600-5-70-BI datasets by 30%. In addition, on Bupa dataset by 20%, the ecoli-0-1-3-7_vs_2-6 dataset by 50%, on Haberman dataset by 28%, the breast dataset by 32%, the Pima dataset by 13%, the Umbilical dataset by 34% and yeast-1-4-5-8_vs_7 dataset by 50%. While the performance improvement in terms of recall on the 03subcl5-600-5-70-BI dataset by 22%, on the 04clover5z-600-5-70-BI dataset by 24%, ecoli-0-1-3-7_vs_2-6 by 50%, on the breast dataset by 27%, ecoli-0-1-3-7_vs_2-6 dataset by 50%, on Haberman dataset by 20%, breast dataset by 32%, Umbilical dataset by 32% and yeast-1-4-5-8_vs_7 dataset by 49%

Table 7 shows the test results using the KNN classifier, where the IRS-BAG method can achieve the best result of accuracy, precision, recall, and F-Measure on five datasets. The most significant performance improvements occurred in the 03subcl5-600-5-70-BI, 04clover5z-600-5-70-BI, Bupa, Haberman, and Breast datasets with an average increase in accuracy, precision, and recall of ±21%.

The most significant increase using the IRS-BAG method occurred in the paw02a-600-5-70-BI with the initial accuracy compared to the Bagging method by 4%, the Bupa dataset by 29%, the Haberman dataset by 21%, the breast dataset by 27%, and in the Umbilical Cord dataset by 7%. Performance improvement in terms of precision on the paw02a-600-5-70-BI dataset with the initial precision compared to the Bagging method by 25%, 04clover5z-600-5-70-BI dataset by 28%, Haberman dataset by 35%, breast dataset by 25%, breast dataset by 27%, and Umbilical Cord dataset by 40%. Improved performance in terms of recall on the paw02a-600-5-70-BI dataset with the initial precision compared to the Bagging method by 30%, glass1 dataset by 15%, breast dataset by 34%, yeast-1-4-5-8_vs_7 dataset by 49%, breast dataset by 27%, and Umbilical Cord dataset by 32%.

The use of these three-based classifiers in the experiments that have been carried out showed that IRS-BAG could provide a significant increase in performance compared to other methods. These results provided evidence of the effectiveness of the Radius-SMOTE method in suppressing the overlapping of data resulting from the oversampling process so that it was easier for the classifier to determine the decision boundary of each class in the dataset.

Table 7. IRS-BAG classification result using KNN based classifier compare with variants of previous SMOTE methods

Metric	Dataset	Bagging	Bagging+ SMOTE [9]	Bagging + ADAYSN [19]	Bagging + IPF [26]	Bagging + TomekLink [32]	Bagging + Borderline [13]	Bagging + Safe Level [12]	IRS-BAG
Accuracy	03subcl5-600-5-70-BI	0.79	0.78	0.81	0.86	0.82	0.85	0.80	**0.87**
	04clover5z-600-5-70-BI	0.71	0.82	0.82	0.80	**0.91**	0.83	0.82	0.88
	bupa	0.69	0.71	0.73	0.51	0.74	0.81	0.69	**0.88**
	ecoli-0-1-3-7_vs_2-6	0.98	0.98	0.98	0.95	0.98	**1.00**	0.96	0.98
	glass1	0.72	0.75	0.81	0.71	**0.85**	0.71	0.62	0.82
	haberman	0.60	0.72	0.72	0.67	0.71	0.72	0.51	**0.81**
	breast	0.61	0.80	0.72	0.73	0.74	0.79	0.61	**0.88**
	new thyroid	**1.00**	0.92	0.93	0.94	0.98	0.94	0.94	0.98
	paw02a-600-5-70-BI	0.81	0.83	0.88	**0.89**	0.93	0.90	0.83	0.87
	pima	0.77	0.81	0.82	0.83	**0.88**	0.83	0.72	0.81
	Umbilical Cord	0.93	0.97	0.94	0.98	0.98	0.99	0.97	**1.00**
	wine	0.90	**1.00**	0.95	0.95	0.98	0.95	0.92	0.92
	yeast-1-4-5-8_vs_7	0.91	0.83	0.89	0.91	0.92	0.93	0.83	0.94
Precision	03subcl5-600-5-70-BI	0.60	0.85	0.81	0.82	0.84	0.77	0.82	**0.85**
	04clover5z-600-5-70-BI	0.61	0.87	0.81	0.88	0.91	0.82	0.77	**0.89**
	bupa	0.70	0.73	0.72	0.32	0.76	0.79	0.67	**0.88**
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.95	0.98	0.92	0.46	0.92
	glass1	0.73	0.73	0.81	0.81	**0.87**	0.73	0.63	0.85
	haberman	0.46	0.71	0.74	0.75	0.71	0.75	0.55	**0.81**
	breast	0.57	0.81	0.73	0.74	0.77	0.81	0.63	**0.82**
	new thyroid	**1.00**	0.93	0.92	0.98	0.91	0.93	0.88	0.98
	paw02a-600-5-70-BI	0.77	0.74	0.92	0.88	0.92	**0.95**	0.83	0.81
	pima	0.77	0.82	0.88	0.84	0.88	**0.89**	0.75	0.85
	Umbilical Cord	0.60	0.95	0.94	0.95	0.99	0.93	0.60	**1.00**
	wine	0.99	**1.00**	0.92	0.87	0.83	0.82	0.93	0.93
	yeast-1-4-5-8_vs_7	0.48	0.90	0.97	0.97	**0.98**	0.94	0.86	0.97
Recall	03subcl5-600-5-70-BI	0.57	**0.87**	0.79	**0.87**	0.78	0.83	0.81	**0.87**
	04clover5z-600-5-70-BI	0.65	0.84	0.85	0.87	**0.93**	0.88	0.82	0.90
	bupa	0.60	0.72	0.70	0.34	0.71	0.81	0.70	0.77
	ecoli-0-1-3-7_vs_2-6	0.49	**1.00**	**1.00**	0.96	0.98	0.99	0.49	0.91
	glass1	0.70	0.72	0.80	0.79	0.92	0.73	0.56	**0.85**
	haberman	0.53	0.71	0.71	0.75	0.76	0.75	0.57	0.76
	breast	0.56	0.89	0.79	0.77	**0.90**	0.81	0.78	**0.90**
	new thyroid	**1.00**	0.98	0.88	0.96	0.91	0.98	0.90	0.92
	paw02a-600-5-70-BI	0.72	0.87	0.91	**0.92**	0.87	0.91	0.82	0.88
	pima	0.78	0.81	0.83	0.88	**0.89**	0.81	0.75	0.84
	Umbilical Cord	0.68	0.89	0.99	**1.00**	0.97	0.96	0.61	**1.00**
	wine	0.98	**1.00**	0.96	0.96	0.98	0.96	0.95	0.94
	yeast-1-4-5-8_vs_7	0.49	0.95	0.96	0.97	0.95	**0.98**	0.87	**0.98**
F-Measure	03subcl5-600-5-70-BI	0.65	0.84	0.84	0.86	0.86	0.85	0.82	**0.89**
	04clover5z-600-5-70-BI	0.64	0.88	**0.90**	0.83	**0.90**	0.88	0.79	0.89
	bupa	0.65	0.71	**0.79**	0.46	0.75	0.77	0.69	**0.79**
	ecoli-0-1-3-7_vs_2-6	0.49	**0.99**	**0.99**	0.96	**0.99**	**0.99**	0.49	**0.99**
	glass1	0.71	0.77	0.83	0.78	**0.89**	0.77	0.67	0.85
	haberman	0.56	0.73	0.72	0.74	0.76	0.76	0.58	**0.79**
	breast	0.57	0.83	0.78	0.76	0.81	0.80	0.64	**0.87**
	new thyroid	**1.00**	0.98	0.97	0.97	0.97	0.96	0.91	0.98
	paw02a-600-5-70-BI	0.75	0.89	**0.91**	0.89	0.93	**0.91**	0.84	0.88
	pima	0.77	0.83	0.82	0.86	**0.87**	0.82	0.75	0.86
	Umbilical Cord	0.66	0.99	**1.00**	**1.00**	0.99	1.00	0.62	**1.00**
	wine	0.96	**1.00**	0.96	0.96	0.98	0.96	0.94	0.94
	yeast-1-4-5-8_vs_7	0.50	0.95	0.96	0.97	0.95	**0.98**	0.87	**0.98**

To visually see the results of oversampling from the proposed method compared to other methods used, visualization was carried out using a scatter plot as shown in Figure 4 below. The dataset used was named as circle where the dataset was conditioned in such a way as to approach the imbalanced dataset condition. The red dots were the distribution of minority data, the blue dots were the majority data, and the green dots were synthetic data resulting from the oversampling method.

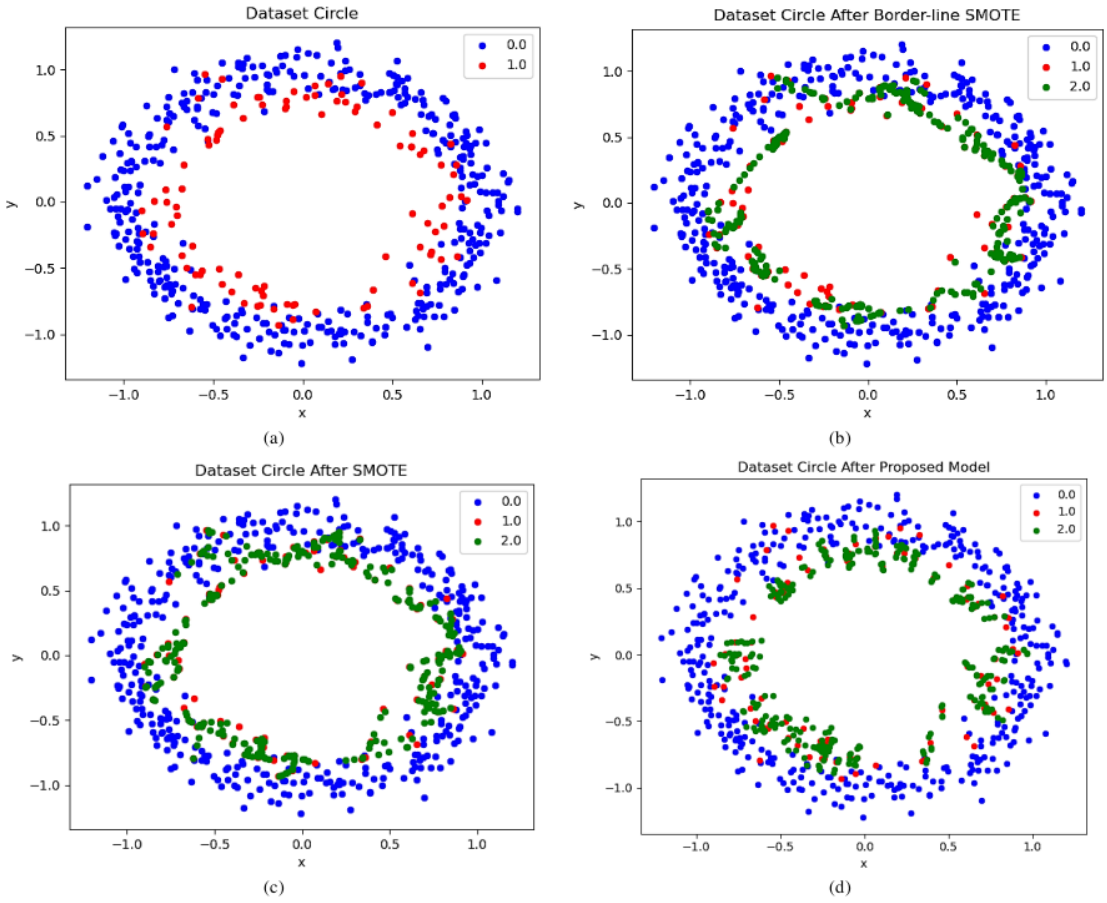

Figure 4. The distribution of synthetic data results from the oversampling method of SMOTE, Borderline-SMOTE, and IRS-BAG

In the figure above, it can be seen that the distribution of synthetic data occurs in each oversampling method. The green dots on the scatter plot indicated the distribution of synthetic data resulting from oversampling. In the SMOTE method, it can be seen that the results of oversampling make the overlapping region between the minority and majority classes more clearly visible. Likewise, in Borderline SMOTE, some of the sampling data were in the majority area, which makes the overlapping conditions more widespread. This condition certainly affected the results of the classifier's performance in determining the decision boundary between the two classes. This proposed IRS-BAG method showed the results of limiting synthetic data to a safer area, just as the Radius-SMOTE concept works based on a safe radius distance. The categorization of data points into SAFE and NOISE means that some minority data points located in the majority area were not selected for sampling. It made synthetic data not created in that region. The emphasis on increasing the number of minority data occurred between the selected sampling points and the closest majority data.

5- Conclusion

This research presented a model for imbalanced dataset classification. Fundamentally, the proposed model was oversampling minority data using the Radius-SMOTE method, which was performed on each sample subset generated by the Bagging algorithm. This research applied the bagging technique before oversampling to avoid accumulating synthetic data with very similar characteristics. Thus, it had the potential to cause overfitting in the learning process by the classifier. The combination of the Radius-SMOTE and Bagging methods was named the IRS-BAG model, which used three base classifiers in the trial: SVM, KNN, and the Decision Tree algorithm. Oversampling the subset of the

sample dataset was done by selecting sampling data in the SAFE category, where the same class dominated the surrounding data. Then the creation of synthetic data was limited to the radius distance obtained from the distance of the sampling data to the closest majority data. Furthermore, the oversampling results became the final dataset used by the classifier to find patterns in each of these classes.

Based on the binary and multiclass datasets used in the experiments, the experiment results using three different classifiers proved that all classifiers had gained a notable improvement when combined with the proposed IRS-BAG model compared with the previous state-of-the-art oversampling methods.

6- Declarations

6-1-Author Contributions

Conceptualization, L.Y. and G.A.P.; methodology, L.Y., G.A.P., and D.P.H.; software, X.X.; validation, G.A.P. and R.R.H.; formal analysis, G.A.P. and P.D.W.A.; investigation, L.Y., G.A.P., P.D.W.A., D.H., and R.R.H.; resources, G.A.P. and P.D.W.A.; data curation, G.A.P. and P.D.W.A.; writing—original draft preparation, G.A.P. and L.Y.; writing—review and editing, G.A.P. and P.D.W.A. All authors have read and agreed to the published version of the manuscript.

6-2-Data Availability Statement

The data used in this research is accessible from UCI and KEEL repository dataset. This dataset is freely accessible at https://archive.ics.uci.edu/ml/datasets.php and https://sci2s.ugr.es/keel/datasets.php.

6-3-Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6-4-Institutional Review Board Statement

Not applicable.

6-5-Informed Consent Statement

Not applicable.

6-6-Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Learning from Imbalanced Data Sets. Springer, Cham, Switzerland. doi:10.1007/978-3-319-98074-4.

[2] Ren, J., Wang, Y., Cheung, Y. ming, Gao, X. Z., & Guo, X. (2023). Grouping-based Oversampling in Kernel Space for Imbalanced Data Classification. Pattern Recognition, 133, 108992. doi:10.1016/j.patcog.2022.108992.

[3] Ganaie, M. A., & Tanveer, M. (2022). KNN weighted reduced universum twin SVM for class imbalance learning. Knowledge-Based Systems, 245, 108578. doi:10.1016/j.knosys.2022.108578.

[4] Anyanwu, G. O., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2023). RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network. Ad Hoc Networks, 140, 103026. doi:10.1016/j.adhoc.2022.103026.

[5] Petinrin, O. O., Saeed, F., & Al-Hadhrami, T. (2017). Voting-based ensemble method for prediction of bioactive molecules. 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA). doi:10.1109/ickea.2017.8169913.

[6] Smith, M. R., & Martinez, T. (2018). The robustness of majority voting compared to filtering misclassified instances in supervised classification tasks. Artificial Intelligence Review, 49(1), 105–130. doi:10.1007/s10462-016-9518-2.

[7] Onan, A., Korukoğlu, S., & Bulut, H. (2016). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Systems with Applications, 62, 1–16. doi:10.1016/j.eswa.2016.06.005.

[8] Bashir, S., Qamar, U., & Khan, F. H. (2015). Heterogeneous classifiers fusion for dynamic breast cancer diagnosis using weighted vote-based ensemble. Quality and Quantity, 49(5), 2061–2076. doi:10.1007/s11135-014-0090-z.

[9] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. doi:10.1613/jair.953.

[10] Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. Journal of Artificial Intelligence Research, 61, 863–905. doi:10.1613/jair.1.11192.

[11] Hoffmann, C. H. (2022). Intelligence in Light of Perspectivalism: Lessons from Octopus Intelligence and Artificial Intelligence. Journal of Human, Earth, and Future, 3(3), 288-298. doi:10.28991/HEF-2022-03-03-03.

[12] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. Advances in Knowledge Discovery and Data Mining. PAKDD 2009. Lecture Notes in Computer Science, Vol. 5476. Springer, Berlin, Germany. doi:10.1007/978-3-642-01307-2_43.

[13] Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, 3644. Springer, Berlin, Germany. doi:10.1007/11538059_91.

[14] Maciejewski, T., & Stefanowski, J. (2011). Local neighbourhood extension of SMOTE for mining imbalanced data. 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). doi:10.1109/cidm.2011.5949434.

[15] Borowska, K., & Stepaniuk, J. (2016). Imbalanced Data Classification: A Novel Re-sampling Approach Combining Versatile Improved SMOTE and Rough Sets. Computer Information Systems and Industrial Management. CISIM 2016. Lecture Notes in Computer Science, 9842. Springer, Cham, Switzerland. doi:10.1007/978-3-319-45378-1_4.

[16] Gosain, A., & Sardana, S. (2019). Farthest SMOTE: A Modified SMOTE Approach. Computational Intelligence in Data Mining. Advances in Intelligent Systems and Computing, 711, Springer, Singapore. doi:10.1007/978-981-10-8055-5_28.

[17] Mahmoudi, S., Moradi, P., Akhlaghian, F., & Moradi, R. (2014). Diversity and separable metrics in over-sampling technique for imbalanced data classification. 4th International Conference on Computer and Knowledge Engineering (ICCKE-2014). doi:10.1109/iccke.2014.6993409.

[18] Wang, G. (2018). D-self-SMOTE: New method for customer credit risk prediction based on self-training and smote. ICIC Express Letters, Part B: Applications, 9(3), 241-246.

[19] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence, Hong Kong. doi:10.1109/ijcnn.2008.4633969.

[20] Torres, F. R., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2016). SMOTE-D a Deterministic Version of SMOTE. Pattern Recognition. MCPR 2016. Lecture Notes in Computer Science, 9703. Springer, Cham, Switzerland. doi:10.1007/978-3-319-39393-3_18.

[21] Asniar, Maulidevi, N. U., & Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification. Journal of King Saud University - Computer and Information Sciences, 34(6), 3413–3423. doi:10.1016/j.jksuci.2021.01.014.

[22] Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Cornelis, C., & Herrera, F. (2012). SMOTE-FRST: a new resampling method using fuzzy rough set theory. Uncertainty modeling in knowledge engineering and decision making. World Scientific, Singapore.

[23] Hu, F., & Li, H. (2013). A novel boundary oversampling algorithm based on neighborhood rough set model: NRSBoundary-SMOTE. Mathematical Problems in Engineering, 2013. doi:10.1155/2013/694809.

[24] Ramentol, E., Caballero, Y., Bello, R., & Herrera, F. (2012). SMOTE-RSB: A hybrid preprocessing approach based on oversampling and under sampling for high imbalanced data-sets using SMOTE and rough sets theory. Knowledge and Information Systems, 33(2), 245–265. doi:10.1007/s10115-011-0465-6.

[25] Ramentol, E., Gondres, I., Lajes, S., Bello, R., Caballero, Y., Cornelis, C., & Herrera, F. (2016). Fuzzy-rough imbalanced learning for the diagnosis of High Voltage Circuit Breaker maintenance: The SMOTE-FRST-2T algorithm. Engineering Applications of Artificial Intelligence, 48, 134–139. doi:10.1016/j.engappai.2015.10.009.

[26] Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences, 291, 184–203. doi:10.1016/j.ins.2014.08.051.

[27] Pradipta, G. A., Wardoyo, R., Musdholifah, A., & Sanjaya, I. N. H. (2021). Radius-SMOTE: A New Oversampling Technique of Minority Samples Based on Radius Distance for Learning from Imbalanced Data. IEEE Access, 9, 74763–74777. doi:10.1109/ACCESS.2021.3080316.

[28] Pradipta, G. A., Wardoyo, R., Musdholifah, A., & Sanjaya, I. N. H. (2022). Machine learning model for umbilical cord classification using combination coiling index and texture feature based on 2-D Doppler ultrasound images. Health Informatics Journal, 28(1), 1–19. doi:10.1177/14604582221084211.

[29] Pradipta, G. A., Wardoyo, R., Musdholifah, A., & Sanjaya, I. N. H. (2020). Improving classification performance of fetal umbilical cord using combination of SMOTE method and multiclassifier voting in imbalanced data and small dataset. International Journal of Intelligent Engineering and Systems, 13(5), 441–454. doi:10.22266/ijies2020.1031.39.

[30] Wardoyo, R., Wirawan, I. M. A., & Pradipta, I. G. A. (2022). Oversampling Approach Using Radius-SMOTE for Imbalance Electroencephalography Datasets. Emerging Science Journal, 6(2), 382–398. doi:10.28991/ESJ-2022-06-02-013.

[31] Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123–140. doi:10.1007/bf00058655.

[32] Tomek, I. (1976). An Experiment with the Edited Nearest-Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics, SMC-6(6), 448–452. doi:10.1109/tsmc.1976.4309523.

IRS-BAG-Integrated Radius-SMOTE Algorithm with Bagging Ensemble Learning Model for Imbalanced Data Set Classification

ORIGINALITY REPORT

6%
SIMILARITY INDEX

PRIMARY SOURCES

1	vdoc.pub Internet	304 words — 3%
2	Retantyo Wardoyo, Aina Musdholifah, Gede Angga Pradipta, I Nyoman Hariyasa Sanjaya. "Weighted Majority Voting by Statistical Performance Analysis on Ensemble Multiclassifier", 2020 Fifth International Conference on Informatics and Computing (ICIC), 2020 Crossref	168 words — 2%
3	www.ijournalse.org Internet	165 words — 2%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 2%
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF