International Review on Computers and Software (I.RE.CO.S.), Vol. 11, N. 12
ISSN 1828-6003 December 2016

RANDSHUFF: an Algorithm to Handle Imbalance Class for Qualitative Data

Tora Fahrudin[1], Joko Lianto Buliali[2], Chastine Fatichah[3]

Abstract – *Class imbalance is a case in which the proportion of training data between one class and another is not balanced, the larger data are called "major class", conversely known as the "minor class". It is believed that accuracy of data mining algorithms can be affected by an imbalance problem. Nowadays, researchers distinguish three main factors of class imbalance that affect the accuracy of data mining algorithm such as overlap, small disjuncts and outliers. A general solution to the problem is the modification of data level or algorithm level. To overcome imbalance problems, we propose a new algorithm called RANDSHUFF(Random Shuffle Oversampling Techniques for Qualitative Data), oversampling synthetic data generation for qualitative data type. RANDSHUFF algorithm uses the concept of neighborhood with IVDM (Interpolated Value Difference Metric) distance calculation and crossovers of the original attribute values and their neighbor's attribute values using the random shuffle technique. Our experimental results showed that RANDSHUFF, combined with Borderline and ADASYN concepts, provides the best results against seven imbalanced public qualitative data type (best minor class Recall on hepatitis, breast cancer and German data and best F-Measure of minor class on hepatitis, abalone and German data).*

Keywords*: Class Imbalance, Oversampling, Synthetic Data, IVDM, RANDSHUFF, Qualitative Data*

Nomenclature

k	The initial number of nearest neighbors
Syn	New synthetic data
gap	Random number between 0 and 1
diff	Difference between one nearest neighbor and original data
w_a	The interval width *w* of attribute *a* value
max_a	Maximum value of attribute *a*
min_a	Minimum value of attribute *a*
$discretize_a(x)$	The discretized value of a continuous value *x* for attribute *a*
s	Integer value for binning continues values into *s* equal-width intervals
$P_{a,x,c}$	Probability class *c* for given input value *x* on attribute *a*
$P_{a,y,c}$	Probability class *c* for given input value *y* on attribute *a*
$vdm_a(x,y)$	Value difference metric for attribute *a* between *x* value and *y* value
c	Class *c*
$P(a_i\|b_i)$	Posterior probabilities of *a* given *b* values
$N_{a,x,c}$	Frequency of value *x* for attribute *a* and class *c* from training data
$N_{a,x}$	Frequency of value *x* for attribute *a* from training data
$N_{a,y,c}$	Frequency of value *y* for attribute *a* and class *c* from training data
$N_{a,y}$	Frequency of value *y* for attribute *a* from training data
$x_i^{chromosome\ length}$	The offspring of data *x* for attribute *I* to *chromosome length*
$Recall\ (-)$	Recall value for minor class
$Precision\ (-)$	Precision value for minor class
$FMeasure\ (-)$	F-Measure value for minor class
TP	True Positive, major class data which correct predict to major class
TN	True Negative, minor class data which correct predict to minor class
FN	False Negative, major class data which fail predict to minor class
FP	False Positive, minor class data which fail predict to major class
β	Harmonization value of Recall and Precision

I. Introduction

Class imbalance is a case in which the proportion of training data between one class and another is not balanced (*significant skewed class distribution*), the class with a larger amount of training data is called "major class", conversely known as the "minor class". Usually, the imbalance ratio between "minor class" and "major

 DOI: 10.15866/irecos.v11i12.10956

class" is about 0.0001-30%. It is believed that accuracy of data mining algorithms can be affected by imbalance problems, which are mostly biased towards "major class" [1][2]. The amount of "minor class" data in imbalance cases are relatively rare compared with normal cases, such as phone fraud, banks fraud, a rare disease in medicine, network intrusion, detection of oil spills from satellite images and so on [3].

Nowadays, researchers distinguish three main factors of class imbalance problems that affect the accuracy of data mining algorithms such as overlap, small disjuncts and outliers[3]. A general solution to overcome imbalance problems is the modification of data (data level)or the adaptation algorithm (algorithm level)[4][5].Compared to the adaptation algorithm, using the modification of data solution provides the advantages of using an independent classifier[6]. Between the two solutions, modification of data more widely studied than the adaptation algorithm[3].

Modification of data is generally divided into two techniques: oversampling and under sampling[7].The most well-known oversampling with additional data synthetic techniques is SMOTE (Synthetic Minority Oversampling Technique)[8]. Although SMOTE can be used to generate synthetic data with qualitative type, but it was basically developed for quantitative data, as seen in the experiments. For this reason, the focus is the development of RANDSHUFF, a new variant of SMOTE to overcome imbalance problems in the qualitative data domain.

RANDSHUFF uses the neighborhood distance computation using the IVDM technique (Interpolated Value Difference Metric), which, according to [9], provides the best results for qualitative dominant data types. While for generating new values for synthetic data based on a crossover method of the original attribute values and their neighbor's attribute values the random shuffle technique is used.

RANDSHUFF uses boundary distance and boundary attributes to guarantee the distance of new synthetic data not more than $\frac{1}{2}$ from its original distance. The success of Borderline and ADASYN concepts in generating synthetic data only on specific area was also used to improve accuracy.

The remainder of this paper is organized as follows: Section 2 explains the related work or literature about oversampling with synthetic data, distance function and cross over concepts. Section 3 describes the proposed algorithm. Section 4 shows the about experimental setting and results from 7 public data with qualitative dominant data type and discussions. Finally, section 5 provides conclusion and the possible future work.

II. Related Work

II.1. Oversampling and Synthetic Data

Oversampling increases the amount of "minor class" training data, while under sampling is reduces the amount of "major class" training data. Based on the sampling ratio of each original data and the presence of synthetic data, the oversampling method was split in 2 parts:

1) Based on the sampling ratio of each original data
   - Same ratio oversampling: increases minor class training data by copying each minor class data with the same ratio until achieving balance distribution.
   - Random oversampling: increases minor class training data by using sampling with random replacement.
2) Presence of synthetic data
   - Without synthetic data: increases minor classtraining data by oversampling original data only.
   - With additional synthetic data: increases minor classtraining data by adding additional synthetic data.

In this study, the oversampling was chosen because under sampling may cause loss of potential data [10]. In the case of oversampling by using a copy of the original data, the minority class decision region becomes very specific, and will cause over fitting [11], and may cause a "lack of data" problem [12][13], that is the background of the emergence of creating a synthetic data idea, which is believed to overcome the lack of "information" problem on the training data. A synthetic data technique in oversampling was popularized by Chawla, et al in SMOTE. The two core concepts of SMOTE can be explained as follows:

1) A method to find *k*-nearest neighbors
2) A method to generate synthetic data based on one of the *k*-nearest neighbors and on the original data
   - *Quantitative* (Quan):compute *diff* and gap. The new synthetic value will be determined by formula (1):

$$Syn[rec][atr] = data[rec][atr] + gap \times diff \qquad (1)$$

   - *Qualitative* (Qual):the new synthetic value will be determined by the majority vote of the feature vectors from its *k*-nearest neighbors.

Some domains of the SMOTE are:

1) Combination with other available methods (Hybrid Method) such as:
   - Ensemble: Boosting (SMOTEBoost) [14], Bagging (SMOTEBagging) [15], Random Subspace (RSM+SMOTE) [16].
   - Under sampling: Rough Set (SMOTE-RSB) [17], Editing Nearest Neighbor (SMOTE+ENN) [18], Tomek Link (SMOTE+TomekLink) [18], Wilson's Editing (SMOTE+WE) [19]
   - Clustering: Density Based (DBSMOTE) [20]
2) Combination with area selection methods of synthetic data generation process (safe area or unsafe area).*Safe* area means the location of data in an homogeneous area. Meanwhile *unsafe* area means the location of data in heterogeneous area. The *unsafe* area was

divided into 2 area, borderline and noise [21]. The development of SMOTE using selection methods proceeds as follows:

- Generation process was done either in *safe* or *unsafe* area: Random SMOTE [5].
- Generation process was done only in *safe* area: Safe-Level-SMOTE [22].
- Generation process was done only in *borderline* area (location of the data is in the border area between major class and minor class): SMOTE Borderline [23].
- Generation process was done in *borderline* and *noise* area: ADASYN [24].

To the best of the author's knowledge, many studies of oversampling with additional synthetic data focused on the quantitative data type. Table I provides the number of unique datasets used in each algorithm based on the amount of their quantitative and qualitative attributes.

TABLE I
LITERATURE METHOD OF OVERSAMPLING WITH DATA SYNTHETIC

Algorithm	Quan	Qual	Quan+Qual	Source
SMOTE [11]	8	0	1	UCI, et al
Random-SMOTE [5]	7	0	3	UCI
SMOTEBoost[14]	3	0	1	UCI, KDD Cup
SMOTE-RSB [17]	7	0	2	UCI
SMOTE+(ENN/Tomek Link) [18]	10	2	3	UCI
SMOTE Borderline [23]	4	0	0	UCI
ADASYN [24]	3	0	2	UCI
Safe-Level-SMOTE [22]	2	0	0	UCI
Databoost-IM [25]	10	3	4	UCI

Databoost-IM is one variant of oversampling with synthetic data algorithm which was developed by Guo, et al. The algorithm has two main concepts that synthetic data generation and Class frequency balancing[25]. A synthetic data generation concept of Databoost-IM scrambles value for each attribute in the original training data independently (to keep synthetic data distribution equal to the original data distribution).

The new synthetic training data retains the original distribution both for nominal and continuous data[25].

II.2. Distance Function for Qualitative Data

In accordance with the above explanation, SMOTE uses two core concepts, finding the nearest neighbors using *k*-NN with a certain distance function and generating new synthetic data. Associated with the distance function, study [9]showed that IVDM had the best performance in computing the distance for data dominated by qualitative attributes.

IVDM uses the statistical probability $P_{a,x,c}$[9]. IVDM requires a non-parametric probability density estimation to produce $P_{a,x,c}$ value[26]. Handling of qualitative and quantitative attributes at IVDM method can be explained as follows:

1) *Quantitative:* Discretized into *s* equal-width interval

$$w_a = \frac{|max_a - min_a|}{s} \tag{2}$$

$$discretize_a(x) = \begin{cases} x, \ if\ a\ discrete \\ s, if\ x = max_a \\ \left\lfloor \frac{(x - min_a)}{w_a} \right\rfloor + 1 \end{cases} \tag{3}$$

2) *Qualitative:* compute $P_{a,x,c}$ and $P_{a,y,c}$

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| P_{a,x,c} - P_{a,y,c} \right|^2 \tag{4}$$

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^2 \tag{5}$$

II.3. Cross Over in Genetic Algorithm

The crossovers method applied to a Genetic Algorithm was "a fundamental mechanism of genetic re-arrangement for both real organisms and genetic algorithms" [27]. That method provides a space solution by combining strings [27]. The principle of the crossovers is to exchange bits chromosomes in a pair and combines them to produce a new individual. This exchange involves a certain probability value (P_c), the determination of the probability value is one of the keys to the success of the genetic algorithm[28].

Figure 1 shows a simple illustration of *cross over*.

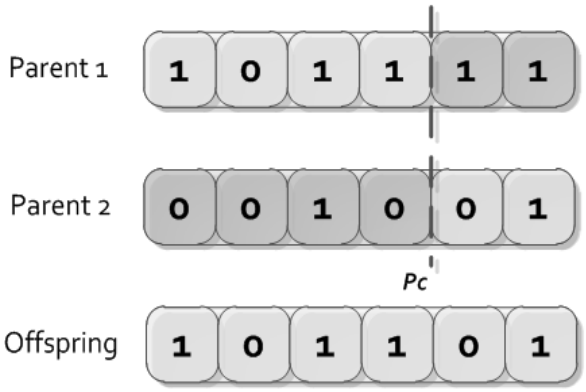


Fig. 1. Cross over simple Illustration

Based on[29], *cross over methods* can be divided into 4 types:

1) Single point cross over

The crossover was done based on a random number from k = 1 to the length of chromosomes. The new individual is formed from k=1 to random numbers from parent1 chromosomes, and the rest of chromosomes from the random number to the length of chromosomes from parent2.

2) Two-point cross over

There are two random numbers, rand1 and rand2, where rand1 <rand2. New individual chromosomes formed by taking k = 1 to rand1 and rand2 to length of chromosomes from parent1, and k = rand1 until rand2 of chromosomes from parent2.

3) Uniform cross over / Discrete crossover

Chromosomes with a value $<P_c$ taken from its first parent and conversely taken from second parent.

4) Flat cross over

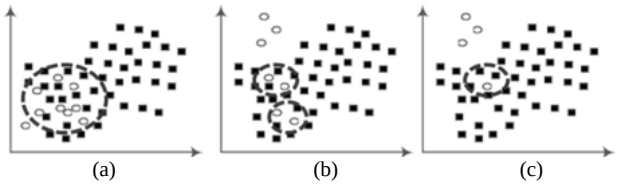
For $Parent1 = (x_{1,1}, \ldots\ldots, x_{1,n})$ and $Parent2 = (x_{2,1}, \ldots\ldots, x_{2,n})$ also random value $r = (r_1, \ldots\ldots, r_n)$. New individual is computed with this formula:

$$x_i^{chromosome\ length} = r_i x_{1,i} + (1 - r_i) x_{2,i} \qquad (6)$$

III. The Proposed Algorithm

Inspired by the success of SMOTE in oversampling with additional synthetic data and also IVDM's method to find the *k*-nearest neighbors dominated by qualitative data type, a RANDSHUFF algorithm was develop, an oversampling technique with synthetic data generation for qualitative data type.

RANDSHUFF also considers three main issues that are common in case of imbalance such as overlap, small disjuncts and outliers [3] (which are shown in Figures 2).



(a) (b) (c)

Figs. 2. Illustration of overlap (a), small disjuncts (b), outlier (c)

RANDSHUFF algorithm was developed with Who_NearestNeighbors and Is_Keep_Correlated parameters input to overcome overlap issue and also to combine with Borderline and ADASYN concepts to overcome small disjuncts and outlier issues.

Overall, RANDSHUFF algorithm requires 6 input parameters: training data (D), the number of neighbors (k), the choice of neighborhoods that will be used as the basis to generate synthetic data (Who_NearestNeighbors), while keeping correlated attribute (IKA), the number of attributes (A), and the order of attribute for label class (A_Class).

Figure 3 illustrates how Who_NearestNeighbors parameter works. For k = 5, at the initial stage, the determination of one neighbor is randomly selected from the *k*-nearest neighbor based on smaller average of IVDM between *k*-nearest neighbors from "major class" and*k*-nearest neighbors from "minor class".

After an average IVDM distance value was obtained for both classes, the smaller value will be selected as a basis random point to generate synthetic data.

Algorithm RANDSHUFFOversampling

```
1. Input: D (Original Training Set), k (Number of Nearest Neighbors),
2.       Who_NearestNeighbors (Average_on_MajorMinor / Minor_Only),
3.       IKA(Is Keep Correlated Attributes True/False),
4.       A (Number of Attributes), A_Class (Attributes Class Order)
5. Output : DS (Original Training + Synthetic Data Set)
6. Process :
7. Nmaj = Count of Major Data on Training Set
8. Nmin = Count of Minor Data on Training Set
9.    N      = Count of Training Set
10.   Ratio = (Nmaj/Nmin) – 1
11.   Arr_Dt[][] = Array of Original Training Set D
12.   Arr_Syn[][] = Array of Synthetic Data
13.   Arr_Corr_Atr[] = Array of Correlated Attributes
14.   Read_data(D) to Arr_Dt[][]
15.   Copy data from Arr_Dt[][] to Arr_Syn[][]
16.   if IKA = true then
17.   find correlated attributes
18.    save correlated attributes index to  Arr_Corr_Atr[]
19.   end if
20.   SI = N+1 (Synthetic Index) /*initialize index for synthetic data */
21.   for i=1 to N do
22.    if Arr_Dt[i][] = Minor Class then
23.     ifWho_NearestNeighbors= Average_on_MajorMinorthen
24.     find knn_major data and compute average of distance for i
25.          find knn_minor data and compute average of distance for i
26.     if average to knn major data > average to knn minor datathen
27.           /* data-i closer to minor nearest neighbors */
28.     forj=1 to Ratio do
29.     create_synthetic_data(i, knn_minor, IKA,SI)
30.              SI = SI + 1
31.          else
32.           /* data-i closer to major nearest neighbors */
33.     forj=1 to Ratio do
34.     create_synthetic_data (i, knn_major, IKA,SI)
35.              SI = SI + 1
36.         end if
37.      else
38.     for j=1 to Ratio do
39.           /* use synthetic from minor nearest neighbors only */
40.     create_synthetic_data (i, knn minor, IKA,SI)
41.             SI = SI + 1
42.     end for
43.      end if
44.    end if
45.   end for
      /* procedure to create synthetic data using randshuff method */
46.   procedure create_synthetic_data(i, knn_data,IKA,SI)
47.     Arr_IVDM [] = Array IVDM value between original data and random
        result of data
48.     vrandshuff[] = Array for random shuffle result
49.     /* initialize order of attributes */
50.     forh=1 to A do
51.     vrandshuff [h] = h
52.     end for
53.     random.shuffle(vrandshuff) /* shuffle order of attributes */
54.     /* find candidate for cross over */
55.     random instance of knn_data , call it R
56.     sum_ivdm = 0
57.     for h=1 to A do
        Arr_IVDM[h] = calculate IVDM(Arr_Dt[i][h], Arr_Dt[R][h])
        sum_ivdm = sum_ivdm + Arr_IVDM[h]
58.     end for
59.     boundary_distance = 1/2 × sum_ivdm
60.     boundary_attribute= 1/2 × A
61.     /* initialize array synthetic data from original */
62.     Copy data from Arr_Dt[i][] to Arr_Syn[SI][]
63.     w = 0 , vsumivdm = 0
64.     while w < 1 do
65.     for h  = 1 toboundary_attributedo
66.     vsumivdm = vsumivdm + Arr_IVDM[vrandshuff[h]]
67.     if sum_ivdm>boundary_distance then
68.              w = w + 1
69.           else
70.             /* overridedata value with R neighbor value data */
71.     Arr_Syn[SI][vrandshuff[h]] = Arr_Dt[R][vrandshuff[h]]
72.           end if
73.       end for
74.     end while
75.     /* check is keep correlated attribute or not */
76.     if IKA=true then
77.     /* override correlated attributes with the original data */
78.      for o=1 to length(Arr_Corr_Atr[]) do
79.      Arr_Syn[SI][Arr_Corr_Atr[o]] = Arr_Dt[i][Arr_Corr_Atr[o]]
```

```
80.    end for
81.    end if
82.    /*override the class label using minor class label*/
83.    Arr_Syn[SI][A_Class] = Arr_Dt[i][A_Class]
84.    Add_data(Arr_Syn[SI][] synthetic data) to DS
85.    end procedure
```

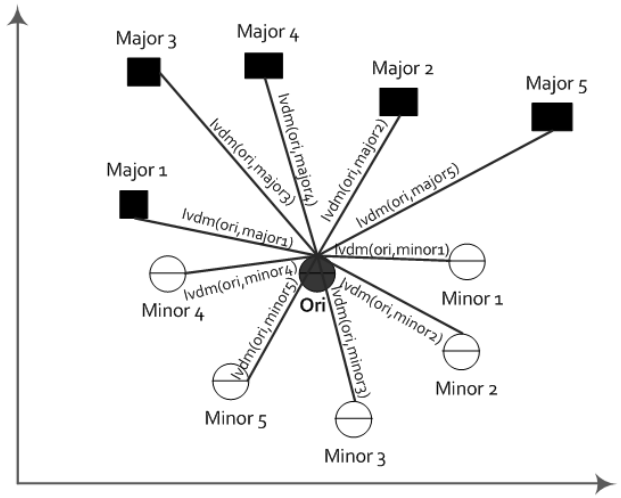


Fig. 3. The calculation of the average IVDM values from original minor data to both of the *k*-nearest neighbors for "major" and "minor" classes

Figure 4 shows that the average value of the five nearest neighbors minor class IVDM is smaller than the majority class, so data generating process will use set of the five nearest neighbors from the minor class as the basis of synthetic data generation. In other cases, if the average of IVDM resulting from major class is smaller than minor class, then the synthetic data generating process will use set of the five nearest neighbors from major class as the basis of synthetic data generation.

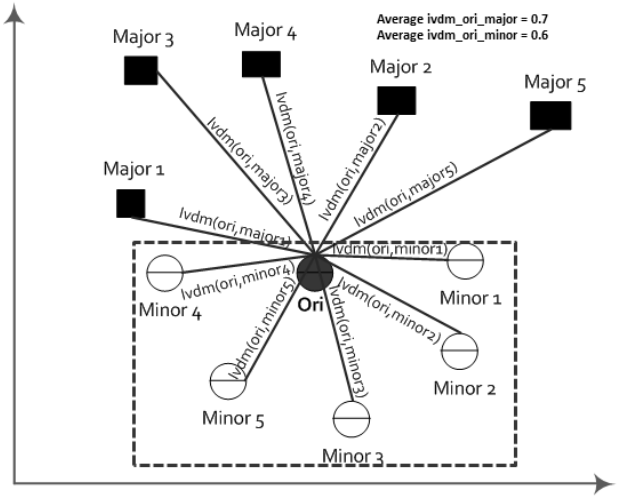


Fig. 4. Synthetic data generation area

Table II below illustrates the generation of synthetic data with 2 resource data, original data and random results from five nearest neighbor's data (in this case minor3 is the random result). IVDM values between original data and minor3 data as follows: 0.1 for $ivdmOri_{A1}Minor3_{A1}$, 0.2 for $ivdmOri_{A2}Minor3_{A2}$, 0.2 for $ivdmOri_{A3}Minor3_{A3}$, 0.1 for $ivdmOri_{A4}Minor3_{A4}$, and0 for $ivdmOri_{A5}Minor3A_5$. The boundary_distance can be computed as $(\frac{1}{2x} \times (0.1 + 0.2 + 0.2 + 0.1 + 0) = 0.3)$, while the boundary attribute value is calculated as$(\frac{1}{2} \times 5 = 2.5)$. The Boundary distance and boundary attribute explained that synthetic data are built with 2 possible boundaries, one is the$\frac{1}{2} \times$ IVDM distance value of the original data and random data, and the second is the$\frac{1}{2} \times$ number of attributes. It guarantees that the IVDM distance of synthetic data from its original data is not more than ½.

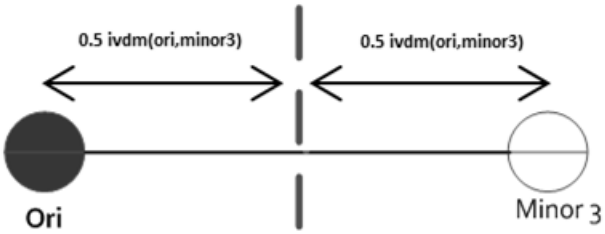


Fig. 5. IVDM maximum boundary distance between the original and its synthetic data

TABLE II
PRELIMINARY DATA AS AN ILLUSTRATION OF THE RANDSHUFF

A1	A2	A3	A4	A5	Label	Notes
18	>20JT	S1	GOL1	PROP1	Minor	Minor3
25	1-3JT	S2	GOL2	PROP1	Minor	Origin

RANDSHUFF randomizes the order of attributes. Those randoming results will be used to exchange values (*crossover*) between the attribute value of original data and the attribute value of one nearest neighbor. Suppose that random shuffle give results A4, A2, A1, A5 and A3. Then, A4 and A2 were selected because the sum of $ivdmX_4Y_4$ and $ivdmX_2Y_2$ reaches the distance boundary (<= 0.3) and also the number of attributes is 2 which reach attributes boundary (<= 2.5). We can generate a new synthetic data straightforward by only exchanging two pair of attribute values from original data and one of nearest neighbor random result as we see in Table III

TABLE III
NEW SYNTHETIC DATA RESULT BASED ON CROSS OVER METHOD FROM ORIGINAL DATA AND ONE OF NEAREST NEIGHBOR RANDOM RESULT

A1	A2	A3	A4	A5	Label	Notes
18	**>20JT**	S1	**GOL1**	PROP1	Minor	Minor3
25	1-3JT	S2	GOL2	PROP1	Minor	Origin
25	**>20JT**	S2	**GOL1**	PROP1	Minor	Synthetic

Meanwhile, to accommodate a "good data", it is necessary to set the Is_Kept_Correlated parameter attribute as true. That means that correlated attribute value was preserved by using the attribute value of the original data. For example, it was found that the attribute correlated is A4.

Therefore, synthetic data generation uses a combination of particular attribute values from the original training data (A1, A3, A5 and the class labels and attributes that correlate A4) and the rest of the attribute value from its random data (A2) (Table IV).

TABLE IV
NEW SYNTHETIC DATA WITH KEEPING CORRELATED ATTRIBUTE

A1	A2	A3	A4	A5	Label	Notes
18	**>20JT**	S1	**GOL1**	PROP1	Minor	Minor3
25	1-3JT	S2	*GOL2*	PROP1	Minor	Origin
25	**>20JT**	S2	*GOL2*	PROP1	Minor	Synthetic

IV. Experimental Setting and Results

Experiment on RANDSHUFF were conducted on UCI data sets that are often used in the imbalance case literature (which contains at least one qualitative attribute). Our RANDSHUFF result was compared with other state of the art oversampling result(such as same portion oversampling, random oversampling, SMOTE IVDM, SMOTE IVDM Borderline, SMOTE IVDM ADASYN, and DataIM (DataboostIM without boosting)).

In presenting the results in the experiment below, the notation of "V1" and "V2"was used to distinguish a different parameter input of RANDSHUFF algorithm. "V1" notation explains that RANDSHUFF was run using parameter IKA = false (without maintaining attributes are correlated). While, the "V2" notation explains that the RANDSHUFF was run using parameter IKA = true (correlated attributes are maintained).

The additional parameter "minor" explains that the RANDSHUFF is executed using the parameters Who_NearestNeighbors = Minor_Only. "Borderline" or "ADASYN" additional parameters showed that the algorithm combined with Borderline or ADASYN concepts to select a minor data region would be processed.

We use a distribution ratio of major and minor as 50:50, which is based on [30], that distribution gives best accuracy in C4.5 algorithm. The nearest neighbors parameter *k* is set to 5, which is the most accepted value in most of the imbalance class literature[12].

We also used C4.5 algorithm which is known as J48 in Weka as base classifier with "prune" and "unprune" settings. To determine that the attributes are correlated (Is_Keep_Correlated_Attributes parameter in our algorithm), the Correlation-based Feature Selection (CFS) of Weka was used. CFS is a method of selecting features by considering the correlation of each of the features with its attributes predictor[31].

IV.1. Data Set Description

7 data sets obtained from the UCI were used to evaluate the performance of our proposed algorithm. For Vowel, and Primary-tumor data sets, one class was selected and the rest merged into a single class with the intention of making binary classes in order to comply with our research focus. In the binary imbalance case, the relationship between classes is well-defined: one is considered as the majority class, and the rest become minority classes[32].

A full description and preprocessing training data that has been done can be seen in Tables V and VI below.

To handle missing values and outlier values (beyond the limit) were found in several datasets above, then the data preprocessing was done and the missing values were replaced using mode and mean values.

TABLE V
DATA SET DESCRIPTION

Data	Maj/Min	Qual/Quan	Safe	Border	Noise
Hepatitis	123/32	13/6	8	21	3
Vowel	900/90	2/10	90	0	0
Sick	3002/221	22/6	0	167	54
Abalone	689/42	1/7	0	17	25
PrimaryTumor	325/14	17/0	0	3	11
BreastCancer	201/85	9/0	27	47	11
German	700/300	13/7	119	140	41

TABLE VI
CLASS LABEL AND PREPROCESS FOR EACH DATA SET

Data	Derived Class	Preprocess
Hepatitis	-	Rep miss val
Vowel	hYd, non_hYd	-
Sick	-	Del TBG attr due to high miss val, del some row with outlier value
Abalone	-	-
Primary Tumor	colon, non_colon	Rep miss val
Breast-Cancer	-	Rep miss val
German	-	-

IV.2. Assessment Metrics

In this paper, the performance of algorithms was evaluated by three parameters (Recall, Precision and F-Measure).

Those parameters usually was used in an imbalance problem with binary class [12].These three parameters are calculated using the confusion matrix as shown in Table VII.

TABLE VII
CONFUSION MATRIX

Actual	**(+) Predicted**	**(-) Predicted**
(+)	TP	FN
(-)	FP	TN

We use the notation (+) to define the major classes and (-) for the minor class. Furthermore, Napierala[3]emphasized that accuracy in minor class is more important than in major class. So, based on that literature, the above three parameters were evaluated for minor class only (-).

As of F-Measure, typically the value of $\beta = 1$ [7], it showed that Recall and Precision occupy the same interests:

$$Recall\ (-) = \frac{TN}{FP + TN} \quad (7)$$

$$Precision\ (-) = \frac{TN}{TN + FN} \quad (8)$$

$$FMeasure\ (-) = \frac{(1 + \beta^2) \times Recall(-) \times Precision(-)}{\beta^2 \times Recall(-) \times Precision(-)} \quad (9)$$

IV.3. Experimental Results

Table VIII gives average result values of minor class performance on Recall, Precision and F-Measure parameters. Table IX provides an explanation about the best three summary result of Recall. Table X provides an explanation about the best three summary result of Precision. Table XI provides an explanation about the best three summary results of F-Measure performance. Table XII provides an average performance increase compared to base classifier using C4.5 algorithm.

IV.4. Analysis and Discussions

Regarding the experimental results in Section 4, the following analysis results were obtained:

1) In vowel data,90 training data in minor class are in a safe position. This makes Borderline and ADASYN useless, since there are no synthetic data to develop.

TABLE VIII
AVERAGE RESULTS OF MINOR CLASS

Data	Algorithms	Prune Results			Unprune Results		
		Recall (-)	Precision (-)	F-Measure (-)	Recall (-)	Precision (-)	F-Measure (-)
Hepatitis	C4.5	31.18%	36.08%	32.57%	39.32%	41.91%	37.04%
	Same Portion Oversampling	56.00%	57.22%	50.73%	56.00%	58.69%	51.88%
	Random Oversampling	52.67%	51.94%	50.07%	52.67%	52.90%	50.18%
	DataIM	69.34%[2]	56.47%	56.83%	67.87%[2]	57.99%	57.71%[2]
	SMOTE IVDM	45.19%	51.26%	42.99%	54.53%	65.59%[2]	54.20%
	SMOTE IVDM Borderline	45.99%	49.59%	43.44%	55.33%	63.39%	52.76%
	SMOTE IVDM ADASYN	49.99%	55.19%	49.04%	51.33%	**68.28%[1]**	53.55%
	Randshuff V1	61.20%	46.29%	49.72%	58.00%	45.36%	48.43%
	Randshuff V1 Borderline	66.93%[3]	49.06%	52.78%	66.00%[3]	49.54%	54.29%
	Randshuff V1 ADASYN	**73.33%[1]**	44.15%	52.58%	**72.26%[1]**	48.08%	55.21%
	Randshuff V1 Minor	64.93%	**62.82%[1]**	**57.78%[2]**	62.26%	62.34%	56.93%[3]
	Randshuff V1 Minor Borderline	61.46%	57.43%	54.78%	60.66%	58.99%	55.69%
	Randshuff V1 Minor ADASYN	58.66%	53.63%	52.38%	58.66%	54.01%	53.16%
	Randshuff V2	56.13%	50.41%	49.21%	56.26%	53.90%	50.40%
	Randshuff V2 Borderline	45.20%	39.41%	40.21%	50.00%	45.67%	45.30%
	Randshuff V2 ADASYN	57.33%	57.00%	51.89%	57.60%	60.90%	54.10%
	Randshuff V2 Minor	66.13%	61.56%[2]	**59.95%[1]**	64.26%	63.99%[3]	**59.80%[1]**
	Randshuff V2 Minor Borderline	41.60%	38.66%	37.91%	44.53%	44.59%	41.80%
	Randshuff V2 Minor ADASYN	58.00%	60.42%[3]	55.15%[3]	56.26%	61.50%	54.30%
Vowel	C4.5	72.20%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Same Portion Oversampling	80.00%	**89.92%[1]**	**81.34%[1]**	80.00%	89.92%[2]	**81.34%[1]**
	Random Oversampling	77.78%	88.09%[2]	78.85%[2]	76.89%	88.90%[3]	78.19%[2]
	DataIM	**94.22%[1]**	57.79%	69.91%	**94.22%[1]**	60.14%	71.61%
	SMOTE IVDM	73.11%	75.33%	70.92%	74.00%	77.68%	72.26%
	SMOTE IVDM Borderline	72.20%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	SMOTE IVDM ADASYN	72.20%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V1	84.66%[2]	69.14%	72.28%	84.88%[2]	68.86%	72.54%
	Randshuff V1 Borderline	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V1 ADASYN	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V1 Minor	84.00%[3]	67.65%	71.42%	84.44%[3]	67.17%	71.33%
	Randshuff V1 Minor Borderline	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V1 Minor ADASYN	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V2	72.67%	82.09%	72.84%	72.23%	83.48%	73.26%
	Randshuff V2 Borderline	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V2 ADASYN	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V2 Minor	74.67%	82.21%[3]	74.08%[3]	74.67%	83.47%	74.48%[3]
	Randshuff V2 Minor Borderline	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
	Randshuff V2 Minor ADASYN	72.22%	88.09%[2]	72.83%	66.66%	**90.00%[1]**	70.67%
Sick	C4.5	90.07%	**91.14%[1]**	90.51%[3]	90.08%	**90.70%[1]**	90.26%[3]
	Same Portion Oversampling	92.82%	89.11%[3]	90.84%[2]	90.55%	90.51%[2]	**90.39%[1]**
	Random Oversampling	92.51%	89.52%[2]	**90.85%[1]**	90.61%	90.46%[3]	90.36%[2]
	DataIM	92.85%	67.10%	77.63%	**94.58%[1]**	61.37%	74.26%
	SMOTE IVDM	86.64%	85.13%	85.51%	91.01%	84.39%	87.37%
	SMOTE IVDM Borderline	87.76%	88.78%	87.93%	91.87%	85.30%	88.15%
	SMOTE IVDM ADASYN	87.19%	84.69%	85.51%	90.37%	84.24%	86.94%
	Randshuff V1	91.77%	57.10%	70.12%	89.18%	54.02%	67.05%
	Randshuff V1 Borderline	85.65%	62.65%	72.22%	88.47%	63.93%	74.10%
	Randshuff V1 ADASYN	93.57%	54.75%	68.90%	91.53%	53.14%	67.04%
	Randshuff V1 Minor	93.85%	77.70%	84.78%	92.05%	77.18%	83.70%
	Randshuff V1 Minor Borderline	90.80%	81.83%	85.85%	90.89%	79.60%	84.56%

Data	Algorithms	Prune Results			Unprune Results		
		Recall (-)	Precision (-)	F-Measure (-)	Recall (-)	Precision (-)	F-Measure (-)
Abalone	Randshuff V1 Minor ADASYN	95.04%[3]	75.50%	84.02%	91.41%	75.85%	82.69%
	Randshuff V2	93.94%	82.55%	87.65%	92.43%[3]	82.37%	86.72%
	Randshuff V2 Borderline	89.60%	83.16%	86.01%	89.59%	83.37%	86.06%
	Randshuff V2 ADASYN	94.23%	80.89%	86.79%	92.60%[2]	80.54%	85.81%
	Randshuff V2 Minor	**95.56%[1]**	78.70%	86.18%	91.58%	79.14%	84.67%
	Randshuff V2 Minor Borderline	87.52%	82.94%	84.97%	87.81%	81.35%	84.07%
	Randshuff V2 Minor ADASYN	95.12%[2]	78.19%	85.67%	91.59%	79.27%	84.76%
	C4.5	14.17%	**38.30%[1]**	18.50%	14.17%	24.16%	16.19%
	Same Portion Oversampling	25.00%	18.38%	21.39%	25.83%	19.20%	21.17%
	Random Oversampling	33.16%	29.88%[3]	29.18%	30.33%	**29.69%[1]**	28.04%
	DataIM	34.33%	23.57%	26.14%	34.83%	21.65%	25.08%
	SMOTE IVDM	**55.00%[1]**	20.43%	29.44%[3]	**53.16%[1]**	19.68%	28.32%[3]
	SMOTE IVDM Borderline	30.83%	21.74%	24.76%	31.16%	20.43%	24.15%
	SMOTE IVDM ADASYN	54.00%[2]	19.70%	28.51%	51.33%[2]	19.94%	27.57%
	Randshuff V1	25.83%	18.48%	20.40%	27.16%	18.71%	20.29%
	Randshuff V1 Borderline	5.17%	8.79%	6.25%	8.17%	12.29%	9.42%
	Randshuff V1 ADASYN	27.33%	18.60%	21.22%	28.33%	17.85%	20.98%
	Randshuff V1 Minor	41.16%[3]	29.56%	31.91%[2]	40.66%	27.43%[3]	31.08%[2]
	Randshuff V1 Minor Borderline	16.33%	24.37%	18.71%	17.33%	23.80%	18.79%
	Randshuff V1 Minor ADASYN	22.76%	22.76%	25.78%	32.99%	22.65%	26.16%
	Randshuff V2	24.83%	19.35%	20.96%	27.33%	20.20%	22.35%
	Randshuff V2 Borderline	15.83%	22.96%	17.50%	14.33%	19.29%	15.68%
	Randshuff V2 ADASYN	32.33%	24.39%	26.63%	33.00%	22.57%	25.81%
	Randshuff V2 Minor	40.50%	30.60%[2]	**32.48%[1]**	41.50%[3]	29.09%[2]	**32.37%[1]**
	Randshuff V2 Minor Borderline	16.83%	25.28%	19.69%	16.83%	22.80%	18.72%
	Randshuff V2 Minor ADASYN	32.66%	25.28%	27.37%	33.66%	26.15%	27.63%
Primary Tumor	C4.5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Same Portion Oversampling	24.00%	11.67%[3]	13.97%	24.00%	17.00%[3]	**17.77%[1]**
	Random Oversampling	22.00%	11.02%	13.08%	24.00%	16.08%	17.17%[2]
	DataIM	**66.80%[1]**	8.49%	14.53%[3]	**38.40%[1]**	5.74%	9.41%
	SMOTE IVDM	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	SMOTE IVDM Borderline	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	SMOTE IVDM ADASYN	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Randshuff V1	32.40%	4.34%	7.23%	21.20%	2.82%	4.72%
	Randshuff V1 Borderline	30.00%	18.43%[2]	18.56%[2]	26.00%	17.09%[2]	16.50%[3]
	Randshuff V1 ADASYN	22.40%	3.08%	4.95%	20.80%	2.44%	4.13%
	Randshuff V1 Minor	34.00%	4.84%	8.09%	30.00%	3.64%	6.22%
	Randshuff V1 Minor Borderline	28.00%	4.75%	7.61%	25.60%	4.30%	6.90%
	Randshuff V1 Minor ADASYN	47.60%[2]	6.56%	10.97%	34.40%[3]	4.92%	8.09%
	Randshuff V2	32.80%	4.22%	7.04%	23.60%	3.40%	5.52%
	Randshuff V2 Borderline	32.00%	**18.63%[1]**	**18.95%[1]**	28.00%	17.49%[1]	17.17%[2]
	Randshuff V2 ADASYN	26.40%	3.44%	5.70%	21.60%	3.02%	4.89%
	Randshuff V2 Minor	29.60%	4.07%	6.77%	23.60%	2.98%	5.03%
	Randshuff V2 Minor Borderline	40.00%[3]	8.23%	12.56%	36.00%[2]	7.29%	10.99%
	Randshuff V2 Minor ADASYN	39.60%	4.90%	8.26%	32.00%	3.90%	6.55%
Breast Cancer	C4.5	25.58%	**66.67%[1]**	35.90%	42.60%	**47.04%[1]**	43.68%
	Same Portion Oversampling	43.85%	43.97%	43.00%	47.60%	42.23%	44.40%
	Random Oversampling	41.44%	44.74%	41.24%	46.94%	42.82%	44.03%
	DataIM	44.38%	51.00%[3]	45.31%[2]	54.44%	43.33%	47.39%[3]
	SMOTE IVDM	45.58%	51.25%[2]	**47.04%[1]**	50.58%	46.20%[3]	**47.75%[1]**
	SMOTE IVDM Borderline	41.63%	39.63%	39.15%	43.37%	41.76%	41.61%
	SMOTE IVDM ADASYN	41.35%	43.24%	40.32%	43.37%	46.43%[2]	42.44%
	Randshuff V1	39.90%	46.92%	41.16%	54.77%	42.91%	47.47%[2]
	Randshuff V1 Borderline	**58.45%[1]**	34.85%	43.09%	**58.49%[1]**	39.63%	46.53%
	Randshuff V1 ADASYN	53.53%[2]	35.53%	41.94%	55.03%[2]	38.71%	45.01%
	Randshuff V1 Minor	42.85%	44.48%	42.58%	53.02%	42.28%	46.51%
	Randshuff V1 Minor Borderline	44.19%	31.18%	35.75%	49.32%	36.73%	41.33%
	Randshuff V1 Minor ADASYN	47.48%	32.89%	38.26%	52.98%	37.32%	43.38%
	Randshuff V2	42.96%	49.41%	44.07%[3]	51.82%	45.80%	45.80%
	Randshuff V2 Borderline	51.68%[3]	36.92%	42.38%	51.49%	40.61%	44.92%
	Randshuff V2 ADASYN	50.21%	36.18%	41.51%	54.06%	38.87%	44.81%
	Randshuff V2 Minor	41.63%	46.15%	42.13%	50.65%	41.43%	45.09%
	Randshuff V2 Minor Borderline	51.15%	35.53%	41.26%	50.55%	38.67%	43.44%
	Randshuff V2 Minor ADASYN	50.67%	36.57%	41.77%	55.02%[3]	39.45%	45.66%
German	C4.5	43.33%	52.58%[2]	47.27%	48.33%	45.94%	46.88%
	Same Portion Oversampling	30.33%	**55.05%[1]**	38.77%	42.33%	48.54%[2]	45.04%
	Random Oversampling	54.00%	47.88%	50.52%	49.60%	46.05%	47.55%
	DataIM	61.20%	48.76%	54.13%[3]	60.53%[2]	44.65%	51.28%
	SMOTE IVDM	54.06%	48.78%	50.99%	51.73%	45.24%	48.12%
	SMOTE IVDM Borderline	57.73%	48.48%	52.42%	53.80%	46.32%	49.61%
	SMOTE IVDM ADASYN	56.93%	47.31%	51.48%	53.53%	44.08%	48.22%

Data	Algorithms	Prune Results			Unprune Results		
		Recall (-)	Precision (-)	F-Measure (-)	Recall (-)	Precision (-)	F-Measure (-)
	Randshuff V1	62.20%	49.40%[3]	54.88%[2]	59.06%	46.39%[3]	51.82%[2]
	Randshuff V1 Borderline	**72.20%[1]**	45.74%	**55.84%[1]**	**64.80%[1]**	**64.80%[1]**	**52.36%[1]**
	Randshuff V1 ADASYN	65.40%[2]	44.44%	52.79%	57.80%	42.03%	48.50%
	Randshuff V1 Minor	59.86%	48.43%	53.36%	55.40%	45.94%	50.10%
	Randshuff V1 Minor Borderline	63.93%	46.25%	53.49%	59.40%[3]	45.75%	51.54%[3]
	Randshuff V1 Minor ADASYN	61.73%	44.88%	51.85%	58.06%	44.02%	49.91%
	Randshuff V2	60.86%	48.17%	53.62%	56.86%	45.50%	50.43%
	Randshuff V2 Borderline	65.26%[3]	45.94%	53.80%	58.80%	43.97%	50.15%
	Randshuff V2 ADASYN	63.93%	45.49%	52.98%	57.26%	43.83%	49.47%
	Randshuff V2 Minor	59.53%	47.98%	52.92%	57.00%	46.14%	50.79%
	Randshuff V2 Minor Borderline	60.46%	45.14%	51.53%	55.80%	44.32%	49.24%
	Randshuff V2 Minor ADASYN	58.60%	43.89%	50.00%	55.80%	44.23%	49.17%

TABLE IX
BEST THREE RECALL (-) ACHIEVEMENTS SUMMARY

Data	Prune			Unprune		
	First	Second	Third	First	Second	Third
Hepatitis	V1 ADASYN	DataIM	V1 Borderline	V1 ADASYN	DataIM	V1 Borderline
Vowel	DataIM	V1	V1 Minor	DataIM	V1	V1 Minor
Sick	V2 Minor	V2 Minor ADASYN	V1 Minor ADASYN	DataIM	V2 ADASYN	V2
Abalone	SMOTE IVDM	SMOTE IVDM ADASYN	V1 Minor	SMOTE IVDM	SMOTE IVDM ADASYN	V2 Minor
Primary Tumor	DataIM	V1 Minor ADASYN	V2 Minor Borderline	DataIM	V2 Minor Borderline	V1 Minor ADASYN
Breast Cancer	V1 Borderline	V1 ADASYN	V2 Borderline	V1 Borderline	V1 ADASYN	V2 Minor ADASYN
German	V1 Borderline	V1 ADASYN	V2 Borderline	V1 Borderline	DataIM	V1 Minor Borderline

TABLE X
BEST THREE PRECISION (-) ACHIEVEMENTS SUMMARY

Data	Prune			Unprune		
	First	Second	Third	First	Second	Third
Hepatitis	V1 Minor	V2 Minor	V2 Minor ADASYN	SMOTE IVDM ADASYN	SMOTE IVDM	V2 Minor
Vowel	Oversampling	C4.5	V2 Minor	C4.5	Same Portion Oversampling	Random Oversampling
Sick	C4.5	Random Oversampling	Same Portion Oversampling	C4.5	Same Portion Oversampling	Random Oversampling
Abalone	C4.5	V2 Minor	Random Oversampling	Random Oversampling	V2 Minor	V1 Minor
Primary Tumor	V2 Borderline	V1 Borderline	Same Portion Oversampling	V2 Borderline	V1 Borderline	Same Portion Oversampling
Breast Cancer	C4.5	SMOTE IVDM	DataIM	C4.5	SMOTE IVDM ADASYN	SMOTE IVDM
German	Same Portion Oversampling	C4.5	V1	V1 Borderline	Same Portion Oversampling	V1

TABLE XI
BEST THREE F-MEASURE (-) ACHIEVEMENTS SUMMARY

Data	Prune			Unprune		
	First	Second	Third	First	Second	Third
Hepatitis	V2 Minor	V1 Minor	V2 Minor ADASYN	V2 Minor	DataIM	V1 Minor
Vowel	Same Portion Oversampling	Random Oversampling	V2 Minor	Same Portion Oversampling	Random Oversampling	V2 Minor
Sick	Random Oversampling	Same Portion Oversampling	C4.5	Same Portion Oversampling	Random Oversampling	C4.5
Abalone	V2 Minor	V1 Minor	SMOTE IVDM	V2 Minor	V1 Minor	SMOTE IVDM
Primary Tumor	V2 Borderline	V1 Borderline	DataIM	Same Portion Oversampling	Random Oversampling , V2 Borderline	V1 Borderline
Breast Cancer	SMOTE IVDM	DataIM	V2	SMOTE IVDM	V1	DataIM
German	V1 Borderline	V1	DataIM	V1 Borderline	V1	V1 Minor Borderline

2) Data with additional synthetic data provide better results of Recall parameter compared to sampling using original data. This is indicated by the oversampling with the addition of synthetic data, to achieve the best 3 in all test data.
3) All oversampling methods (with and without additional synthetic data)may cause a reduction of Precision value as seen in Table XII, which is indicated by the negative value.
4) For the F-Measure results, the use of original and additional synthetic training data equally provides impartial results. It is indicated for hepatitis, abalone, breast cancer and German data, oversampling with additional synthetic training data method provides

better value than oversampling with original data. On the contrary, for the vowel and sick data, oversampling with original training data provides better value than oversampling with additional synthetic training data.

5) In certain data, as seen in Tables IX, X and XI, the generation of synthetic training data only in "Border" or "Noise" areas implemented on V1 or V2, seem to improve accuracy. Improvement of Recall occurs in hepatitis, primary tumor, breast cancer and German data. Improvement of Precision and F-Measure occurs in primary tumor data.
6) In certain data, the use of the parameter Who_NearestNeighbors = Minor_Only consistently deliver better results on Recall, Precision and F-Measure than Average_on_MajorMinor. V1 Minor provided better result than V1 on hepatitis, sick, abalone, and primary tumor. V2 Minor provided better results than V2 on hepatitis and abalone data.
7) As of "prune" and "unprune" performance of C4.5 algorithm, the experimental results of Recall and F-Measure in Table XII showed that "prune" performance gives a better average increase than "unprune".

TABLE XII
AVERAGE PERFORMANCE INCREASE (COMPARED WITH BASE CLASSIFIER)

Algorithms	Prune			Unprune		
	Recall (-)	Precision (-)	F-Measure (-)	Recall (-)	Precision (-)	F-Measure (-)
Same Portion Oversampling	10.78	**-1.07**[1]	6.07	9.31	**3.76**[1]	**6.75**[2]
Random Oversampling	13.86	**-1.40**[2]	**8.03**[2]	**13.86**[3]	-1.39	**8.03**[1]
DataIM	**26.66**[1]	-8.53	6.7	**18.94**[1]	-6.41	4.57
SMOTE IVDM	11.86	-5.81	4.19	9.924	-0.14	4.76
SMOTE IVDM Borderline	10.19	-5.38	4.56	5.41	-1.37	2.73
SMOTE IVDM ADASYN	12.16	-4.95	4.30	8.36	**1.89**[2]	3.52
V1	17.35	-11.60	2.60	11.60	-8.67	1.09
V1 Borderline	16.30	-9.32	3.43	8.37	-0.35	2.74
V1 ADASYN	17.62	-10.90	2.99	10.27	-5.24	1.98
V1 Minor	**20.59**[2]	-5.34	**7.48**[3]	**15.13**[2]	-1.97	5.87
V1 Minor Borderline	14.34	-5.57	4.49	8.02	-0.08	3.54
V1 Minor ADASYN	18.42	-6.94	5.50	11.19	-1.57	4.19
V2	15.38	-5.24	5.40	10.48	-0.73	4.25
V2 Borderline	13.61	-5.39	4.87	6.69	0.09	3.60
V2 ADASYN	17.16	-5.34	5.82	9.49	-0.003	4.41
V2 Minor	**18.73**[3]	**-3.08**[3]	**8.13**[1]	13.27	**0.93**[3]	**6.79**[3]
V2 Minor Borderline	13.32	-7.00	3.31	6.45	-1.53	2.03
V2 Minor ADASYN	18.62	-5.07	6.21	10.61	0.68	4.86
Average	***15.94***	***-5.99***	***5.22***	***10.41***	***-1.23***	***4.21***

V. Conclusion

From the experimental results, the author concluded as follows:

1) Oversampling with additional synthetic data is able to overcome the problem of "lack" of information on training data (as indicated by increase of Recall and F-Measure values). However, the drawback of oversampling with additional synthetic data is a reduction of Precision value.
2) RANDSHUFF provides an alternative in the development of an oversampling technique with additional synthetic data that have been tested on a qualitative dominant data type. RANDSHUFF provides competitive performance compared to other "state of the art" imbalance algorithms like SMOTE IVDM, Oversampling, and DataIM (indicated for better achievement of Recall and F-Measure values). Flexibility of RANDSHUFF can be set using the input parameters such as the type of neighbor class and maintenance of attribute values are correlated.
3) To overcome three main issues in imbalance problems (overlap, small disjuncts and outliers), Randshuff can be combined with Borderline or ADASYN concepts. The use of "neighborhood from minor class only" parameter provides consistent improvement for 3 parameters on some dataset.

Limitations of this study include the following:

1) Performance evaluation of this proposed algorithm has been accepted only in the domain of qualitative data on 7 public data sets. Different results may occur in other data sets.
2) The smaller number of attributes could lead to smaller variations of synthetic data, or in other words, data are likely to be the same as the original data.

A Future development of this study could be:

1) To combine our RANDSHUFF algorithm with an ensemble environment like Boosting and Bagging, such as the DataboostIM, SMOTE-Boosting or SMOTE-Bagging methods.
2) To test RANDSHUFF algorithm in multiclass imbalance problems which contain more complex situations because of the complexity of the decision boundary of different classes[12].

References

[1] R. Longadge, S. Dongre, and M. Latesh, "Class Imbalance Problem in Data Mining: Review," *Int. J. Comput. Sci. Netw.(IJCSN)*, vol. 2, no. 1, 2013.

[2] P. Thanathamathee and C. Lursinsap, "Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and AdaBoost techniques," *Pattern Recognit. Lett.*, vol. 34, no. 12, pp. 1339–1347, 2013.

[3] K. Napierała, "*Improving Rule Classifiers For Imbalanced Data*,"

Doctoral Dissertation, Faculty of Computer Science, Poznan University of Technology, Piotrowo 2, Poznan, Poland, 2012.
[4] D. Tomar and S. Agarwal, "A Survey on Pre-processing and Post-processing Techniques in Data Mining," *International Journal of Database Theory and Application*,vol. 7, no. 4, pp. 99–128, 2014.
[5] Y. Dong and X. Wang, "A new over-sampling approach: Random-SMOTE for learning from imbalanced data sets," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7091 LNAI, Springer Berlin Heidelberg, 2011, pp. 343–352.
[6] A. Fernández, V. López, M. Galar, M. José, and F. Herrera, "Knowledge-Based Systems Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," in *Knowledge-Based Systems*, vol. 42, 2013, pp. 97–110.
[7] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.
[8] N. Verbiest, E. Ramentol, C. Cornelis, and F. Herrera, "Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection," *Appl. Soft Comput. J.*, vol. 22, pp. 511–517, 2014.
[9] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *J. Artif. Intell. Res.*, vol. 6, pp. 1–34, 1997.
[10] A. M. Mahmood, "Class Imbalance Learning in Data Mining – A Survey," *Int. J. Commun. Technol. Soc. Netw. Serv.*, vol. 3, no. 2, pp. 17–36, 2015.
[11] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique,"*Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[12] L. Abdi and S. Hashemi, "To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, 2016.
[13] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning Deep Representation for Imbalanced Classification," *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.
[14] N. V Chawla, A. Lazarevic, L. O. Hall, and K. Bowyer, "SMOTEBoost: improving prediction of the minority class in boosting," *7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, PKDD-2003, pp. 107–119, 2003.
[15] F. S. Hanifah, "SMOTE Bagging Algorithm for Imbalanced Dataset in Logistic Regression Analysis (Case: Credit of Bank X)," *Appl. Math. Sci.*, vol. 9, no. 138, pp. 6857–6865, 2015.
[16] T. R. Hoens and N. V. Chawla, "Generating diverse ensembles to counter the problem of class imbalance," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6119 LNAI, no. PART 2, pp. 488–499, 2010.
[17] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB *: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, 2012.
[18] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *SIGKDD Explor. Newsl.(Special issue on learning from imbalanced datasets)*, vol. 6, no. 1, pp. 20–29, 2004.
[19] V. García, A. I. Marqués, and J. S. Sánchez, "Improving risk predictions by preprocessing imbalanced credit data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7664 LNCS, no. PART 2, pp. 68–75, 2012.
[20] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "DBSMOTE: Density-based synthetic minority over-sampling technique," *Appl. Intell.*, vol. 36, no. 3, pp. 664–684, 2012.
[21] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One Sided Selection," *Proc. Fourteenth Int. Conf. Mach. Learn.*, vol. 4, no. 1, pp. 179–186, 1997.
[22] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE : Safe-Level-Synthetic Minority Over-Sampling TEchnique",*Lect. Notes Comput Sci*, vol. 5476, pp. 475-482, Springer Berlin Heidelberg, 2009.
[23] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE : A New Over-Sampling Method in," in *International Conference on Intelligent Computing, ICIC 2005*, Hefei, China, August 23-26, 2005, Proceedings, Part I, Springer Berlin Heidelberg, 2005, pp. 878–887.
[24] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). S* on, 2008, no. 3, pp. 1322–1328.
[25] H. Guo and H. L. Viktor, "Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach," *ACM SIGKD Explor. Newsl. - Spec. issue Learn. from imbalanced datasets*, vol. 6, no. 1, pp. 30–39, 2004.
[26] H. Wang and W. Dubitzky, "A flexible and robust similarity measure based on contextual probability," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 27–32, 2005.
[27] J. H. Holland, Genetic Algorithms - Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand.*Scientific American*, pp. 66-72, 1992.
[28] W. Y. Lin, W. Y. Lee, and T. P. Hong, "Adapting crossover and mutation rates in genetic algorithms," *J. Inf. Sci. Eng.*, vol. 19, no. 5, pp. 889–903, 2003.
[29] J. Magalhães-Mendes, "A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem," *WSEAS Trans. Comput.*, vol. 12, no. 4, pp. 164–173, 2013.
[30] S. Visa and A. Ralescu, "Issues in mining imbalanced data sets-a review paper," in *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference*, 2005, no. August 2016, pp. 67–73.
[31] M. F. Naufal and S. Rochimah, "Software complexity metric-based defect classification using FARM with preprocessing step CFS and SMOTE a preliminary study," in 2015 *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2015, pp. 1–6.
[32] J. A. Sáez and B. Krawczyk, "Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets," *Pattern Recognit.*, vol. 57, pp. 164–178, 2016.

Authors' information

[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Jl Raya ITS, Keputih, Sukolilo, Surabaya, Indonesia. School of Applied Science, Telkom University Jl Telekomunikasi Terusan Buah Batu, Bandung Indonesia.
E-mails: tora15@mhs.if.its.ac.id
torafahrudin@telkomuniversity.ac.id

[2]Department of Informatics, Institut Teknologi Sepuluh Nopember, Jl Raya ITS, Keputih, Sukolilo, Surabaya, Indonesia.
E-mail: joko@cs.its.ac.id

[3]Department of Informatics, Institut Teknologi Sepuluh Nopember, Jl Raya ITS, Keputih, Sukolilo, Surabaya, Indonesia.
E-mail: chastine@cs.its.ac.id

Tora Fahrudin was born in Sukoharjo on October 23, 1985. He received ST. in informatics engineering from Sekolah Tinggi Teknologi Telkom (STT Telkom), Bandung, Indonesia. He received the M.T. in management telecomunication from Institut Teknologi Telkom, Bandung, Indonesia at the end of 2010. Currently he is entering the second year of doctorate student in computer science at Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests are Database and Data Mining. He has published some papers in the Indonesian domestic conference and one paper in Journal of Theoretical and Applied Information Technology. Tora Fahrudin is currently a member in International Association of Engineers (IAENG).

Joko Lianto Buliali was born in Surabaya on July 27, 1967. He received Ir. in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. He then received M.Sc. and Ph.D. in computation from University Of Manchester Institute of Science and Technology (UMIST), Manchester, England, in 1995 and 1998, respectively. Currently he is professor in modeling and simulation at Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. His research interests include system modeling and simulation, intelligent systems, business process engineering, object-oriented systems. Joko Lianto Buliali is currently a member in Association for Computing Machinery (ACM), a senior member in Society for Modeling and Simulation International (SCS), and a member in Asosiasi Pendidikan TinggiIlmu Komputer Indonesia (APTIKOM).

Chastine Fatichah was born in Pasuruan on December 20, 1975. She received S.Kom. in informatics engineering from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. She received the M.Kom. in computer science from Universitas Indonesia, Depok, Indonesia. She received her Ph.D. from Tokyo Institute of Technology, Japan in 2012. She is currently a lecturer at Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. Her research interests include medical image processing, soft computing, and data mining. Chastine Fatichah is currently a member in International Association of Engineers (IAENG).