Enriched Over_Sampling Techniques for Improving Classification of Imbalanced Big Data

Sachin Subhash Patil
Computer Science and Engineering
R.I.T., Rajaramnagar
Islampur - Sangli, India
sachin.patil@ritndia.edu

Shefali Pratap Sonavane
Information Technology
W.C.E., Sangli
Sangli, India
shefali.sonavane@walchandsangli.ac.in

Abstract— **Big Data generated in exabytes per year has become a watchword of today's research. They are exceptionally afar from the capability of commonly used software tools and also beyond the handling possibility of the single machine architecture. Facing this challenge has activated a requisite to reexamine the data management options. The new avenues of NoSQL Big Data compared to the traditional forms has insisted on adapting experimental beds, helping to discover large unknown values from enormous data sets. Also, outmoded management systems and statistical packages express trouble handling Big Data. In numerous real applications, handling of imbalanced data sets is the fact of precedence. The classification of data sets having imbalanced class distribution has produced a notable drawback in performance obtained by the most standard classifier learning algorithms. Assuming balanced class distribution and equal misclassification costs lead to poor results. In a real-world domain, the classification methods of multi-class imbalance problem need more attention compared to the two-class problem. A methodology is presented for binary/multi-class imbalanced data sets with improved over_sampling (O.S.) techniques to enhance classification. The methods are broadly classified into two categories: non-clustered and cluster based advanced approach compared to prior work on O.S. techniques. The balanced data are subsequently analyzed for classification using various classifiers. Proposed techniques are performed using mapreduce environment on Apache Hadoop, using various data sets from UCI/KEEL repository. F-measures and ROC area are used to measure the performance of this classification.**

Keywords- imbalanced data sets; Big Data; over_sampling techniques; data level approach

I. Introduction

Big Data is a catchphrase of today's research which is basically dependent on huge digital data generated in zettabytes per year. There is no dearth of data in today's enterprise, but the spotlight is to focus on integration, exploitation and analysis of information. The study of some performance techniques is needed to harness the efficient handling of Big Data streams.

As per resources [1], the size of digital data in 2011 is roughly 1.8 Zettabytes (1.8 trillion gigabytes) which is estimated to be supported by networking infrastructure having to manage 50 times more information by the year 2020 [2]. Concentric considerations of efficiency, economics and privacy should carefully be planned. Big Data challenges induced by traditional data generation, consumption and analytics are handled efficiently. But, recently in sighted characteristics of Big Data has shown vital trends of access, mobility, utilization as well as ecosystem capabilities [3,5,7]. Nowadays, mining knowledge from huge varied data for better decision making is a challenge [4].

Furthermore, many real world applications present classes which have an insignificant number of samples as compared to other classes. This situation is called as a class imbalance problem. Usually, the insignificant samples are the main focus of study; hence it is necessary to classify them correctly. In machine learning research, learning from imbalanced data sets is an issue that has attracted a lot of attention. The statistical learning methods are suited for balanced data sets and may ignore the negligible samples which are important. Almost classifiers work with rationally balanced data sets in contrast to real-world applications. The learning algorithms try to discover the preeminent result boundaries, which are difficult to represent in imbalance data sets. Additionally, the skewed data distribution is unaddressed by various classifier learning methods. That‘s why it is required to consider the features of the problem and solve it correctly.

In this paper, the various enhanced O.S. techniques used to deal with binary-class/multi-class imbalanced data problem are presented. These approaches are evaluated on the basis of effectiveness in the precise classification of each instance of each class. Various classifiers (Random Forest (R.F.) [6], Naïve Bayes) based on their scalability, robustness are used in order to perform classification. Non-clustered and clustering based O.S. techniques are experimented using the mapreduce based framework. In an imbalanced data domain, let us consider the cancer patient detection problem, in which the percentage of cancer detected patients compared to the normal ones is very low, as 3%. The classifier intends to make a perfect decision that a given details of the patient feature as normal gains the 97% of accuracy. Even if the occurrence of cancer detected patient is restricted to 3% of all the records, it may outcome in huge medical fatalities. In accordance, the efficacy in the classification of imbalanced data sets can basically be

978-1-5090-6318-5/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataService.2017.19

IEEE computer society

evaluated using two measures: F- Measure for true rates and ROC area [8].

II. Classification of Imbalanced Data sets

In mining, the categorization of input into pre-labeled classes is done on certain similarity aspects. The categories may either be of multi-class or two class. The analysis of various techniques for multi-class imbalanced data problem is required to be focused on many real world problems such as medical diagnosis [9], fraud detection, finances, risk management, network intrusion, E-mail foldering, Software Defect Detection [10]. The classification of imbalanced data sets poses problems where class distributions having a number of examples in one class are outnumbered by other classes [11]. The results from rare classes have been identified as one of the main challenges in data mining. The positive (minority) class is the class of interest from the learning point of view and has a greater impact when misclassified [6]. The global search measure which does not takes into account the variance between the statistics of instances of each class leading to challenging for learning from imbalanced data sets. During model construction, the specific rules used for identification of minority class instances are mostly ignored in the presence of more general rules, which are used to identify the instances of majority class.

Several techniques are available to address the classification of imbalanced data [12]. These techniques are categorized into various groups [6]:

1) Data Level Approach:

An original imbalanced data set is modified to get a balanced data set and further analyzed by standard machine learning algorithms to get the required results.

2) Algorithm Level Approach:

An existing algorithm is modified to launch procedures that can deal with imbalanced data.

3) Cost-Sensitive Approach:

Both, the data level and the algorithm level approaches are combined to get accuracy and reduce misclassification costs.

The techniques discussed, deal with the data level approach in detail. Furthermore, data level approaches are divided into various groups: O.S., Undersampling and Hybrid technique [13]. The Over_sampling and Undersampling techniques have some drawbacks such as, in O.S., noisy data may get replicated and in Undersampling, important data may get lost due to random selection scheme [6].

Synthetic Minority Oversampling Technique (SMOTE) algorithm [14] is an O.S. technique, used as a powerful solution to solve imbalanced data set problem by adding synthetic minority class samples to original data set to achieve the balanced data set [6].

In SMOTE algorithm, minority class is oversampled by duplicating samples from minority class. Depending on the O.S. required, numbers of the nearest neighbors are chosen randomly [6]. The synthetic data is then generated based on feature space likeliness prevails between existing samples of a minority class. Consider S as the set of all samples and Smin as the set of all minority instances. For subset $S_{min} \epsilon S$, consider k nearest neighbors for each sample $x_i \epsilon X$. The K-nearest neighbors are the K elements whose Euclidean distance between itself and x_i has the smallest weight along the n-dimensional feature space of X. The samples are generated as simply as, randomly selecting one of the K-nearest neighbors and multiplying the corresponding Euclidean distance with a random number between [0,1]. Finally, this value is added to x_i.

$$x_{i\_new} = x_{i\_old} + (x_{i\_KNN} - x_i)*\delta \qquad (1)$$

Where, $x_i \epsilon S_{min}$ is a sample from minority class used to generate synthetic data. x_{i_KNN} is the nearest neighbor for x_i and δ is a random number between [0,1]. The generated synthetic data is a point on line fragment between x_i under consideration and x_{i_KNN} k-nearest neighbors for x_i.

The following Fig.1.a shows imbalanced class distribution, where the circles represent the majority class and stars represent the minority class. The K-nearest neighbor is set to K=5. The Fig.1.b shows synthetic data generation on the line segment joining x_i and x_{i_KNN} and it is spotlighted by diamond. Finally, synthetic data are added to the original data sets in order to balance it.

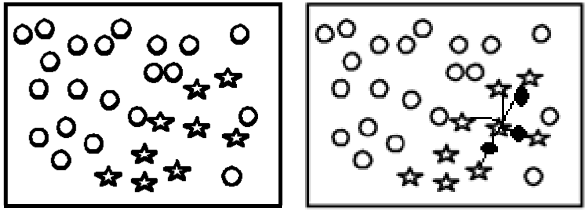

Fig. 1.a Imbalanced Class Distribution Fig. 1. b Generation of synthetic data, k=5

Though SMOTE is a popular technique in the imbalanced domain, it has some drawbacks including over-generalization, only applicable for the binary class problem, over-sampling rate varying with the data set. To avoid these drawbacks, some approaches are defined such as Borderline-SMOTE and Adaptive Synthetic Sampling for generalization. Evolutionary algorithms and sampling methods are used to deal with the class imbalance problem [15]. SMOTE+GLMBoost [16] and NRBoundary-SMOTE are based on the neighborhood Rough Set Model [17] and are used to solve class the imbalance problem. The ensemble methods like AdaBoost, RUSBoost and SMOTEBoost are coupled with SMOTE to solve imbalanced data problems [9]. All these approaches are focused on the two-class problem. In [18], the author proposed a solution for multi-class problem based on fuzzy rule classification. Ensembles of decision trees (R.F.) have been the most successful general-purpose classification algorithm in modern times [6]. R.F. was proposed by Leo Breiman in 2002 to improve the classification of a data set having a small training data set and large testing data set i.e. R.F. is suited for large number attributes and a small number of observations [12]. R.F. is the scalable, fast and durable approach for classification of high-dimensional data and can handle continuous data, categorical data and time-to-event data [19].

III. METHODOLOGY

A. Architecture

The planned research is grounded in experimental analysis, which comprises a range of exploratory statistical and quantitative methods. The study comprises the challenge of working with imbalanced Big Data sets (I.B.D.) using statistical methods. It incorporates the procedure of how to obtain data, process it, store it and convert it into a format suitable for analysis, model the experimental design and interpret the final results.

The theoretical concept deals with the development of some techniques to enhance the traditional approaches to handle streaming high-velocity data mining aspects in mapreduce execution environment. Moreover, the imbalanced classification of minority data versus majority samples is to be studied to provide a well-balanced data.

The exploration of Big Data includes multiple distinct phases as shown in the Fig. 2. The huge streams of data are required to handle, accept, record and store. The 'storage and retrieval' is a major concern of performance. Moreover, streaming inputs with heterogeneity are to be addressed with approximation techniques to provide some useful early insights. The inputs are further processed with enhanced O.S. techniques (Non-clustered/ clustered based techniques) for creating balanced data set. The outputs are finally analyzed for improved accuracy.

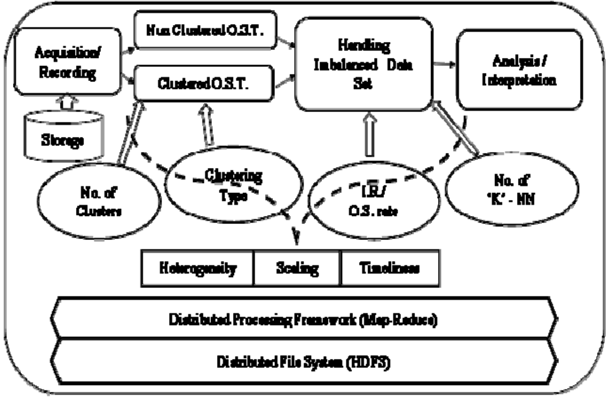

Fig. 2 Overall System Architecture

B. Enriched O.S. Techniques for Imbalanced Data set

An approach for O.S. of two-class or multi-class imbalanced data is addressed. The techniques can be applied in the non-clustered or clustered form on I.B.D [26].

1) Addressing Multi-Class Imbalanced Data Sets:

One v/s One (OVO) and One v/s All (OVA) are the insightful techniques for handling multi-class imbalanced data set comprising numerous overheads.

In the proposed technique viz. Lowest v/s Highest (LVH), is a unique method sufficing almost of the disadvantages of OVO and OVA technique. This technique helps to reduce computation in addition to improving classification performance.

LVH - The idea quoted is more robust compared to the above two techniques for multi-class data set balancing.

- Compares the individual lowest minority class (all classes below a certain threshold, i.e. 35-40%) v/s one highest majority class only.
- O.S. of minority classes is carried out one by one, forming the final data set.
- Avoid duplication cum reduced computation.
- Reduces synthetic samples generations and provides more realistic interpolated samples compared to all other minority set underlying.
- Evade overshooting of other majority sub-classes after O.S.
- Reduces computation of test samples for final classification.
- Comply the highest majority class indirectly, conforming all remaining classes within.

A, B, C is assumed to be majority classes, of which A is the highest majority class. Further, a, b, c is assumed to be as minority classes and a is the lowest minority class. The method starts with the comparison of lowest minority class (a) with only the highest majority class (A). The O.S. rate equals to the size of highest majority class. The minority class under consideration is Over_sampled. Correspondingly, the next remaining minority classes (b and c) are Over_sampled one-by-one. This method will surely help to enhance the accuracy of the model.

The three methods discussed below can be used with LVH for balancing of multi-class data sets to improve classification results.

2) Non-cluster based O.S. Techniques:

- <u>Technique1</u> - **ME**re **M**ean **M**inority **Over_Sampling T**echnique **(MEMMOT)**:

This technique is enhanced SMOTE methodology. A SMOTE provides a disadvantage of duplicating majority/minority samples. In MEMMOT, following procedure avoids almost a duplication of interpolated instances.

Let the data set be D_i, D_{maj} – majority class instances z_m (m = 1,2,….,m) and D_{min} – minority class instances x_n (n = 1,2,….n).

Find safe levels of all instances before processing such that, the safe level of particular instance is the count of minority instances in its K-NN [20]. Further, for each minority instance x_n (for 100% O.S. rate):

Algorithm: **MEMMOT**
Input: a set of all instances D_i
Output: a set of all synthetic positive instances D_o

```
1. $D_o = \emptyset$
2. Repeat {
3. for each positive instance $x_n$ in $D_i$ {
4. compute k nearest neighbors ($N_k$) for $x_n$ in $D_i$
5. Clear $D_s$ and $D_{si}$
6. for each k nearest neighbors ($N_k$) {
7. generate a random a number between 0 and 1, call
   it g
8. for each features of selected $N_k$ and $x_n$ {
9. dissimilarity = $N_k$[feature] - $x_n$[feature]
10. synthetic[feature] = $x_n$[feature] + dissimilarity * g
11. }
12. $D_s = D_s \cup \{$ synthetic $\}$
13. }
14. $D_{si} = \{$ Averaging of $D_s \}$
15. Check the generated instance $D_{si}$ for duplication
    among the present data set.
    if Yes : reduce the lowest safe level nearest
    neighbor instance from the current K-NN.
    Reduce the interpolated instance of the respective
    lowest safe level nearest neighbor instance from
    the current interpolated instances $D_s$. Repeat step
    14
    else : $D_o = D_o \cup \{ D_{si} \}$
16. }
17. }
18. Until O.S. rate
19. return $D_o$
```

For O.S. rate above 100%:

Reduce the lowest safe level nearest neighbor instance from the current K-NN. Reduce the interpolated instance of the respectively lowest safe level nearest neighbor instance from the current interpolated instances. Repeat step 14 to comply the O.S. rate. The value of K should satisfy the condition as-

$$K > \% \text{ O.S.} / 100$$

OR

Repeatedly use the current over sampled set for further over sampling based on MEMMOT, till the satisfaction of O.S. rate.

OR

i. Based on safe levels or random basis – select 50% samples out of first 100% over sampled instances and remaining 50% from the original set. Use this combined set for next O.S. generations with MEMMOT.
ii. For the more O.S rate, based on safe levels or random basis, select 33% samples from each – Original, First 100% and Second 100% over sampled sets. Use this combined set for next O.S. generations with MEMMOT.
iii. Continue step ii. with reduced selection ratios of 25%, 12.5%, 6.25% and so on from original and over sampled sets….till the O.S. rate is satisfied.

For O.S. rate below 100%, select the interpolated samples either randomly or on the basis of the high safe level, which complies the required O.S. rate. In either case, if the technique fails to comply imbalance ratio (I.R.) than under-sampling based on clustering [21] can be used to reduce majority classes to match the number of minority instances.

The above method helps to provide more generalized synthetic minority instances with low repetition rates and improves classification accuracy.

- Technique 2 - **M**inority **M**ajority **M**ix mean **O**ver_Sampling **T**echnique **(MMMmOT)**:

This technique is a unique advancement of SMOTE methodology.

The technique (MMMmOT) considers K-NN minority as well as majority samples for further O.S. of interpolated instance. It helps to provide a more generalized interpolated sample, less duplication faults with overcoming boundary-line samples.

Find safe levels of all instances before processing [20]. Further, for each minority instance x_n (n = 1, 2,….n and for 100% O.S. rate):

Algorithm: **MMMmOT**
Input: a set of all instances D_i
Output: a set of all synthetic positive instances D_o

```
1. $D_o = \emptyset$
2. Repeat {
3. for each positive instance $x_n$ in $D_i$ {
4. compute k nearest neighbors ($N_k$) for $x_n$ in $D_i$
5. Clear $D_s$ , $D_{snr}$ , $D_{smk}$ and $D_{si}$
6. Check for K-NN instances set, either all instances
   are minority or majority or minority-majority mix
7. if all instances are of minority class
8. for each k nearest neighbors ($N_k$) {
9. generate a random a number between 0 and 1, call
   it g
10. for each features of selected $N_k$ and $x_n$ {
11. Dissimilarity = $N_k$ [feature] - Xian [feature]
12. synthetic[feature] = $x_n$[feature] + dissimilarity * g
13. }
14. $D_s = D_s \cup \{$ synthetic $\}$
15. }
16. $D_{si} = \{$ Averaging of $D_s \}$
17. Check the generated instance $D_{si}$ for duplication
    among the present data set.
    if Yes : reduce the lowest safe level nearest
    neighbor instance from the current K-NN.
    Reduce the interpolated instance of the respective
    lowest safe level nearest neighbor instance from
    the current interpolated instances $D_s$. Repeat step
    16
    else : $D_o = D_o \cup \{ D_{si} \}$
18. }
19. else if all instances are of majority class {
20. select an instance from K-NN on a random basis,
    $K_{nr}$.
21. find the nearest minority sample to the current
    majority sample($K_{nr}$) under consideration, $N_{mk}$.
22. generate a random a number between 0 and 1,
    call it g
23. for each features of selected $K_{nr}$ and $x_n$ {
24. dissimilarity = $K_{nr}$[feature] - $x_n$[feature]
25. synthetic[feature] = $x_n$[feature] + dissimilarity * g
26. }
27. $D_{snr} = D_{snr} \cup \{$ synthetic $\}$
28. for each features of selected $N_{mk}$ and $x_n$ {
29. dissimilarity = $N_{mk}$[feature] - $x_n$[feature]
30. synthetic[feature] = $x_n$[feature] + dissimilarity * g
31. }
32. $D_{smk} = D_{smk} \cup \{$ synthetic $\}$
33. $D_{si} = \{$ Averaging of $D_{snr}$ and $D_{smk}$ $\}$
34. Check the generated instance $D_{si}$ for duplication
    among the present data set.
    if Yes : select the next nearest minority sample
    ($N_{mk}$) to the current majority sample ($K_{nr}$) under
    consideration. Repeat from step 22
    else : $D_o = D_o \cup \{ D_{si} \}$
35. }
36. else {
37. select an instance from K-NN on a random basis
38. if the selected instance is of minority class, $N_k$ {
39. generate a random a number between 0 and 1,
    call it g
40. for each features of selected $N_k$ and $x_n$ {
41. dissimilarity = $N_k$[feature] - $x_n$[feature]
42. synthetic[feature] = $x_n$[feature] + dissimilarity * g
43. }
44. $D_s = D_s \cup \{$ synthetic $\}$
45. Check the generated instance $D_s$ for duplication
    among the present data set.
    if Yes : remove the instance under consideration.
    Repeat step. Repeat from step 37
    else : $D_o = D_o \cup \{ D_{si} \}$
46. }
47. else{
48. in reference to the current majority sample ($K_{nr}$)
    under consideration, find a highest safe level
    minority sample from in hand minority samples
    in K-NN set, $N_{mk}$
49. generate a random a number between 0 and 1,
    call it g
50. for each features of selected $K_{nr}$ and $x_n$ {
51. dissimilarity = $K_{nr}$[feature] - $x_n$[feature]
52. synthetic[feature] = $x_n$[feature] + dissimilarity * g
53. }
54. $D_{snr} = D_{snr} \cup \{$ synthetic $\}$
55. for each features of selected $N_{mk}$ and $x_n$ {
56. Dissimilarity = $N_{mk}$ [feature] - Xian [feature]
57. synthetic[feature] = $x_n$[feature] + dissimilarity * g
58. }
59. $D_{smk} = D_{smk} \cup \{$ synthetic $\}$
60. $D_{si} = \{$ Averaging of $D_{snr}$ and $D_{smk}$ $\}$
61. Check the generated instance $D_{si}$ for duplication
    among the present data set.
    if Yes : remove the instance under consideration.
    Repeat step. Repeat from step 37
    else : $D_o = D_o \cup \{ D_{si} \}$
62. }
63. }
64. Until O.S. rate
65. return $D_o$
```

For O.S. rate above 100%:

Reduce the lowest safe level instance from the current k-NN set. Further, step 6 to 65 can repeatedly be used for all remaining instances to comply the O.S. rate.

The value of K should satisfy the condition as-

$$K >= \% \text{ O.S. rate} / 100$$

OR

Repeatedly use the current over sampled set for further over sampling based on MMMmOT, till the satisfaction of O.S. rate.

OR

i. Based on safe levels or random basis – select 50% samples out of the first 100 % over sampled instances and remaining 50% from the original set. Use this combined set for next O.S. generations with MMMmOT.
ii. For more O.S rate, based on safe levels or random basis, select 33% samples from each – Original, First 100% and Second 100% over sampled sets. Use this combined set for next O.S. generations with MMMmOT.
iii. Continue step ii. with reduced selection ratios of 25%, 12.5%, 6.25% and so on from original and over sampled sets….till the O.S. rate is satisfied.

For O.S. rate below 100%, select the interpolated samples either randomly or on the basis of the high safe level, which complies the required O.S. rate. In either case, if the technique fails to comply I.R. than under-sampling based on clustering [21] can be used to reduce majority classes to match the number of minority instances.

3) *Clustering based O.S. Techniques:*

In contrast to above non-clustered O.S. techniques for two-class/multi-class, clustering based techniques work with the unsupervised approach and further over sample the minority class data sets for required balanced objective. For two/multi-class data sets (LVH based), Clustering Minority Examples

(CMEOT) technique is used. Various clustering algorithms can be used for clustering with the enhanced technique.

- Technique 3 - **C**lustering **M**inority **E**xamples **O**ver_Sampling **T**echnique **(CMEOT)**:

This technique is a pure cluster based technique. The technique involves only the instances of minority classes for synthetic samples generation. The means of clusters basically seem to synthetic instances. The technique helps to provide the same objective as like DBSMOTE [25] of enriching centroids based O.S.

Find safe levels of all instances before processing [20]. Further, for each minority instance x_n (n = 1, 2,….,n and for 100% O.S. rate):

```
Algorithm: CMEOT
Input: a set of all instances $D_i$
Output: a set of all synthetic positive instances $D_o$
1. $D_o = \emptyset$
2. Repeat {
3. cluster only minority data set using any clustering algorithm (basically K-Means and K<N) i.e. $C_1, C_2, \ldots C_k$ clusters.
4. each mean (Centroid, $C'_1, C'_2, \ldots C'_k$ ) seems to be a new interpolated synthetic sample.
5. check individual $C'_1, C'_2, \ldots C'_k$ for duplication among the present data set.
   if Yes : reduce the remove that respective centroid. Repeat from step 3
   else : $D_o = D_o \cup \{ C' \}$
6. }
7. Until O.S. rate
8. return $D_o$
```

For achieving the O.S. rate:

- Add the centroids obtained in iteration to previous minority data set forming new set and continue Step 3 to 8 (Keeping cluster no. and initial seeds same as previous iteration)

OR

- Either repeat the Step 3 to 8 by reducing an element within the original data set based on lowest safe level, till data set_size > K (Keeping cluster no. same as previous iteration but the initial seeds will be different)

The above methods provide generalized synthetic instances with low repetition rates and helps improving classification.

In either case, if the technique fails to comply I.R. than Under-Sampling Based on Clustering [21] can be used to reduce majority classes to match the number of minority instances.

In [22], an online fault detection algorithm based on incremental clustering is studied. [23], addresses the issue of ordinal classification methods in imbalanced data sets. A novel class detection in data streams with efficient string based methodology is deliberated in [24].

C. Structure of I.B.D. techniques:

The overall proposed structure of I.B.D. handling techniques are shown in Fig. 3. O.S. techniques are used to handle large imbalanced data sets.

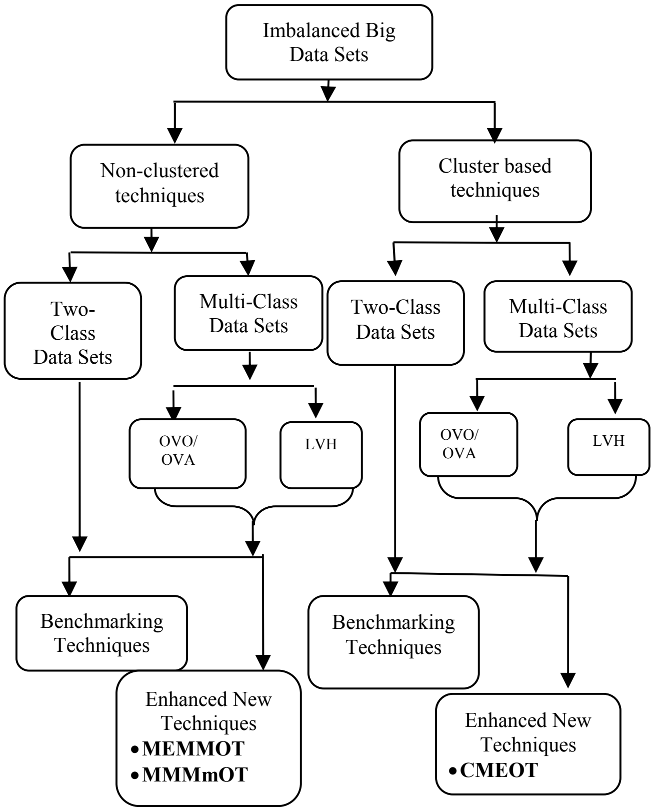

Fig. 3. Structure of I.B.D. techniques

IV. CONCEPTUAL FLOW OF EXPERIMENTATION

The conceptual flow of analysis consists of following steps:

1. Identify the imbalanced training set and form an R.F. tree of it. Use testing set to check for initial bare results for comparison at last. Parallely, this data set can be clustered and checked for cluster cohesiveness.
2. Carry out O.S. of two-class or multi-class imbalanced training data sets, either using the un-clustered simplistic term or on a clustering basis.
3. Perform – R.F. on same.
4. Use Testing set – for model prediction and accuracy testing.
5. Update R.F., if the I.R. goes above 1.5 (+10%) OR error rate goes above a certain threshold (recall). <If the error rate goes above a certain threshold – then re-correct the data set with prior known classes>
6. The new corrected data can further be used to improve R.F. tree using Step 2 and 3. Parallely this data can be analyzed for cluster cohesiveness and similarity cohesiveness.Continuously, collect real-time incoming

data set for further prediction and analysis through R.F. tree. Repeat step 5 to 7 for this real-time data set.

7. Analyzes the classifier performance on said measurements by respective methods. Progress to the conclusion.
8. Optionally – can repeat Step 7 and 8 for the infinite sequence of time to improve the classifier accuracy as per the requirement.

Fig. 4. explains the logical flow of experimental work and analyze for necessary results.

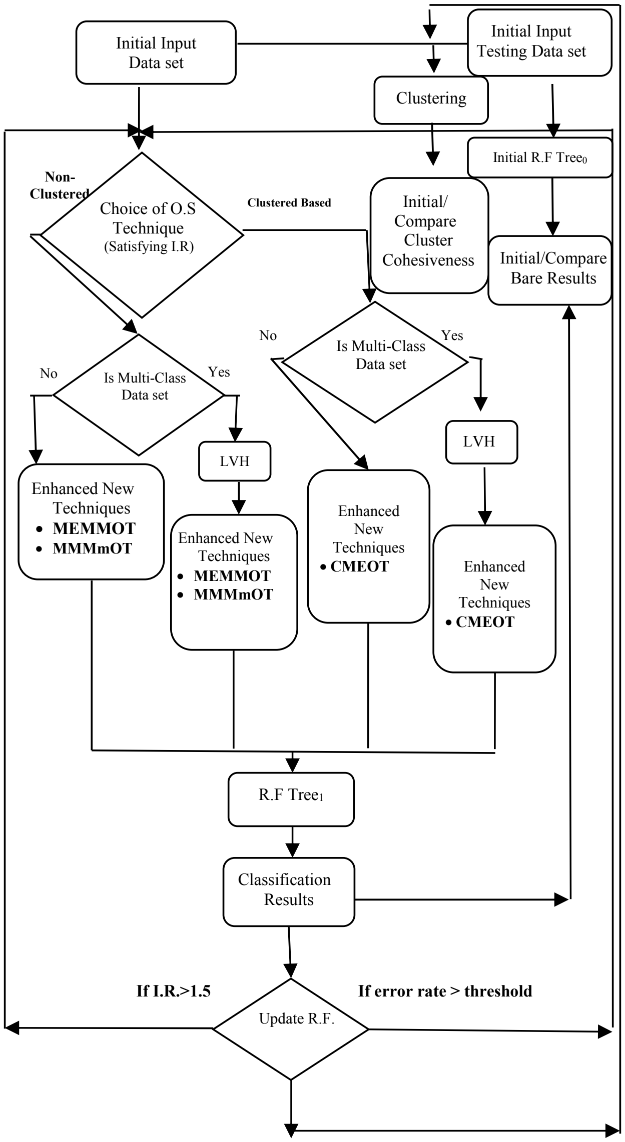

Fig.4. Conceptual Flow of Experimentation

V. EXPERIMENTAL FRAMEWORK

In this section, the details are presented regarding the experimentation setup and evaluation analysis used to compare the performance of different approaches.

A. Evaluation Parameters:

Experiments can be performed on Apache Hadoop and Apache Spark using diverse data sets from the UCI/KEEL repository. F-measures, ROC area can be planned to measure the performance of this classification.

The proposed major steps for implementation are as follows:

1. To convert the data set from CSV format into Hadoop Sequence File format consisting of key/value pairs.
2. To map the data set in OVO/OVA/LVH model.
3. To implement O.S. algorithms to convert imbalanced data set to balanced form.
4. To analyze data set using various classifiers for given performance metrics as an F-measure, ROC area of varying data size, a number of data nodes.

The effectiveness is to be evaluated using given two measures that are able to efficaciously rate the success in imbalanced classification.

– **F-measure:** Metric used to assess the quality of classifiers in the imbalanced domain is F-measure and it is given as:

$$\text{F-measure} = \frac{2\,R_i P_i}{R_i + P_i} \quad for\ all\ i = 1, 2, \ldots.k \qquad (2)$$

P_i - precision of ith class.

- Precision = True Positive / (True Positive +False Positive)

- predicted positive cases that were correctly classified.

- Sensitivity (recall) = True Positive / (True Positive +False Negative)
- Specificity = True Negative / (False Positive + True Negative)

- **ROC area:** Area under a graphical plot that illustrates the presentation of a binary classifier system as its discernment threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

B. Data Sets:

HBase is an open source, distributed, column-oriented database modeled after Google's BigTable. HBase is a distributed Key / Value store built on top of Hadoop. HBase shines with large amounts of data and read/write concurrency. The data sets are acquired for further analysis in HBase.

Two categories of data sets are selected for experimentation. The first category consists of five standard multivariate data sets. They comprise less attributes, I.R. and examples compared to the second category (binary) of data sets. Basically, the standard data sets are collected from KEEL (Knowledge Extraction based on Evolutionary Learning) database repository for experimentation as below:

TABLE I. CHARACTERISTICS OF DATA SET

Category	**Data set**	**#ATTR**	**#IR**	**#EX**	**#CL**
Multivariate Structured Data Sets **(M.V.)**	cleveland	14	1.17	302	4
	ecoli	8	10.58	336	8
	glass	10	3.19	214	7
	led7digit	7	10.97	443	10
	yeast	8	8.56	1484	10
Binary Structured Data Sets **(B)**	KEGG Metabolic Reaction Network (Undirected)	29	64.55	65554	2
	Nomao	120	2.12	34465	2
	Skin Segmentation	4	3.81	245057	2

Table I. summarizes the details of selected data sets which includes documented name of the data set, number of attributes (#ATTR), I.R. (#IR), number of instances (#EX) and number of classes (#CL).

C. Experimental Study:

The experiments are performed on three node hadoop cluster. Each node has Intel 2.5 GHz i3 processor having 4 GB RAM. The heap size is set to 1024 MB for all mappers/reducers.

Notations for Table III - VI. and Fig. 5 - 8 are given in TABLE II. :

TABLE II. NOTATIONS

C-1 - RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1	**D-7** – Nomao
C-2 - NaiveBayes	**D-8** – Skin
C-3 - AdaBoostM1-P100-S1-I10-Wweka.classifiers.trees. DecisionStump	**A** – SMOTE
D-1 - cleveland	**B -** SafeLevel SMOTE
D-2 - ecoli	**C-** ROS
D-3 - glass	**D-** MEMMOT
D-4 - led7digit	**E-** MMMmOT
D-5 - yeast	**F-** CMEOT
D-6 - KEGG	--

For experimentation, 10-fold cross validation partitioning scheme is used. The experiments are carried out on two class/multi-class data sets. The results for all data sets are compared between base techniques (SMOTE/ SafeLevel SMOTE/ ROS) and new proposed techniques (MEMMOT/ MMMmOT/ CMEOT) on evaluation parameters viz. F-measure and ROC area. They are described respectively in Table III. - VI., underlying three different classifiers (R.F./NaiveBayes/AdaBoostM1). The significant results are indicated in boldface.

The results obtained show that both, F-measure and ROC area overall average values give better results for all proposed methods (MEMMOT/ MMMmOT/ CMEOT), indicating improved classification.

The results of R.F. are more promising compared to other two classifiers viz. NaïveBayes and AdaBoostM1.

TABLE III. COMPARING F-MEASURE VALUE FOR DATA SET (M.V.) WITH BASE METHOD AND PROPOSED METHOD (R.F./NAIVEBAYES/ADABOOSTM1)

Classifier	**Data set**	**Over Sampling Techniques**					
		A	**B**	**C**	**D**	**E**	**F**
C-1	D-1	0.878	0.899	0.884	0.916	**0.922**	0.909
	D-2	0.908	0.914	0.91	0.956	**0.962**	0.945
	D-3	0.913	0.928	0.913	**0.948**	0.946	0.944
	D-4	0.902	0.916	0.91	0.934	0.943	**0.949**
	D-5	0.899	0.905	0.899	0.918	0.929	**0.93**
C-2	D-1	0.844	0.855	0.843	**0.899**	**0.899**	0.887
	D-2	0.869	0.873	0.871	0.927	**0.931**	0.928
	D-3	0.892	0.884	0.873	0.929	**0.933**	0.932
	D-4	0.873	0.888	0.883	0.922	**0.934**	**0.934**
	D-5	0.869	0.876	0.876	0.909	0.913	**0.919**
C-3	D-1	0.829	0.833	0.827	0.914	**0.922**	0.918
	D-2	0.933	0.918	0.928	0.951	**0.962**	0.958
	D-3	0.921	0.928	0.93	0.937	**0.955**	0.951
	D-4	0.907	0.914	0.909	0.934	**0.94**	0.935
	D-5	0.91	0.913	0.91	0.919	**0.923**	0.922
Overall Average		0.889	0.896	0.891	0.927	**0.934**	0.931

TABLE IV. COMPARING ROC AREA VALUE FOR DATA SET (M.V.) WITH BASE METHOD AND PROPOSED METHOD (R.F. /NAIVEBAYES/ADABOOSTM1)

Classifier	**Data set**	**Over Sampling Techniques**					
		A	**B**	**C**	**D**	**E**	**F**
C-1	D-1	0.928	0.934	0.914	0.951	**0.952**	**0.952**
	D-2	0.941	0.944	0.941	0.969	**0.972**	0.970
	D-3	0.923	0.931	0.930	0.952	0.957	**0.959**
	D-4	0.922	0.926	0.919	0.944	**0.953**	0.949
	D-5	0.939	0.941	0.939	0.948	**0.959**	0.953
C-2	D-1	0.924	0.935	0.93	**0.959**	**0.959**	0.957
	D-2	0.901	0.913	0.91	0.957	**0.963**	**0.963**
	D-3	0.929	0.931	0.931	0.959	**0.963**	0.961
	D-4	0.903	0.911	0.91	0.942	**0.954**	0.951
	D-5	0.911	0.922	0.92	0.959	**0.961**	0.953
C-3	D-1	0.899	0.903	0.899	0.934	**0.941**	0.94
	D-2	0.953	0.955	0.955	**0.969**	0.968	**0.969**
	D-3	0.941	0.958	0.953	0.967	**0.969**	0.961
	D-4	0.937	0.944	0.941	0.954	**0.964**	0.955
	D-5	0.943	0.949	0.948	0.959	0.963	**0.965**
Overall Average		0.927	0.933	0.929	0.954	**0.959**	0.957

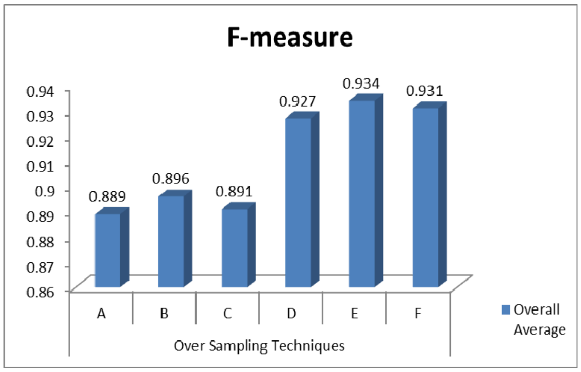

Fig. 5

The graph in Fig. 5, represents the overall average F-measure values of all techniques for respective data sets (M.V.) underlying three classifiers viz. R.F., NaïveBayes and AdaBoostM1 respectively.

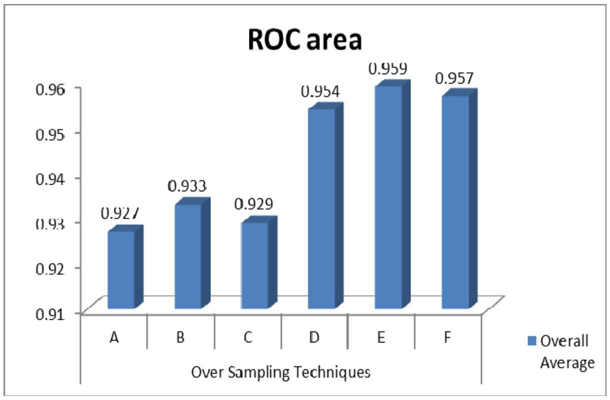

Fig. 6

The graph in Fig. 6, describes the overall average ROC area values of all techniques for respective data sets (M.V.) underlying three classifiers viz. R.F., Naïve Bayes and AdaBoostM1 respectively.

The technique MMMmOT performs better on multivariate data sets compared to other two new proposed techniques viz. MMMmOT and CMEOT.

TABLE V. COMPARING F-MEASURE VALUE FOR DATA SET (B) WITH BASE METHOD AND PROPOSED METHOD (R.F./ADABOOSTM1)

Classifier	Data set	Over Sampling Techniques					
		A	B	C	D	E	F
C-1	D-6	0.899	0.913	0.907	0.931	0.933	0.948
	D-7	0.92	0.934	0.927	0.948	0.949	0.953
	D-8	0.945	0.949	0.949	0.958	0.967	0.978
C-3	D-6	0.876	0.898	0.90	0.921	0.923	0.935
	D-7	0.899	0.914	0.907	0.928	0.929	0.941
	D-8	0.922	0.939	0.939	0.944	0.947	0.958
Overall Average		0.91	0.925	0.923	0.938	0.941	0.952

TABLE VI. COMPARING ROC AREA VALUE FOR DATA SET (B) WITH BASE METHOD AND PROPOSED METHOD (R.F. / ADABOOSTM1)

Classifier	Data set	Over Sampling Techniques					
		A	B	C	D	E	F
C-1	D-6	0.925	0.933	0.931	0.941	0.953	0.958
	D-7	0.945	0.948	0.949	0.958	0.959	0.965
	D-8	0.969	0.971	0.979	0.979	0.983	0.984
C-3	D-6	0.912	0.926	0.929	0.952	0.953	0.965
	D-7	0.931	0.934	0.947	0.953	0.959	0.967
	D-8	0.952	0.959	0.953	0.964	0.967	0.968
Overall Average		0.939	0.945	0.948	0.957	0.962	0.967

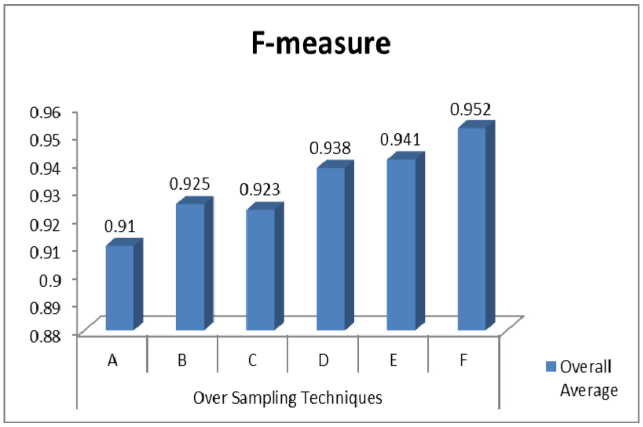

Fig. 7

The graph in Fig. 7, depicts the overall average F-measure values of all techniques for respective data sets (B) underlying two classifiers viz. R.F. and AdaBoostM1 respectively.

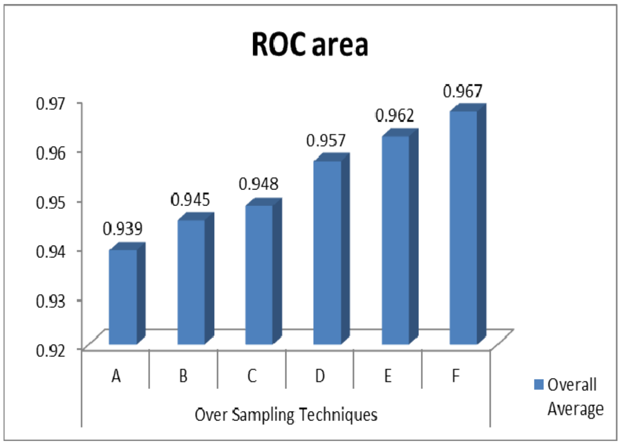

Fig. 8

The graph in Fig. 8, represent the overall average ROC area values of all techniques for respective data sets (B) underlying two classifiers viz. R.F. and AdaBoostM1 respectively.

The technique CMEOT performs better on binary data sets, handling big data sets in size, I.R., number of attributes compared to other two new proposed techniques viz. MMMEOT and MMMmOT.

VI. CONCLUSION

In this study, the enhanced data preprocessing techniques for two-class and multi-class imbalanced data have been presented using non-clustered/clustered based O.S. techniques. Various classifiers are used as base classifiers. Traditional data mining techniques are unable to survive with requirements urged by Big Data; hence, the mapreduce framework under Hadoop environment is used to deal with it.

The system quality testing benchmark may be indexed in terms of the parameters like accuracy, AOC area, G-Mean and F-measure. Experimental analysis is carried out using various data sets of the UCI/KEEL repository. The proposed three methods, namely MEMMOT, MMMmOT and CMEOT outperform the existing methods for selected data sets. The results indicate that proposed methods show an improved score of F-measure and ROC area, compared to the base techniques improving classification. At the same time, concerns raised out of the intrinsic data characteristics like small disjuncts, lack of density, overlapping and impact of borderline instances are addressed. The issues related to data set shift and changing O.S. rate needs to be further addressed in depth.

ACKNOWLEDGMENT

This work had been supported by Walchand College of Engineering, Sangli (MH – India, 416415) under TEQIP-II.

REFERENCES

[1] A. Gandomi and M. Haider, "Beyond the hype: big data concepts, methods, and analytics," International Journal of Information Management, vol. 35, no. 2, pp. 137–144, 2015.

[2] W. Zhao, H. Ma, and Q. He., "Parallel k-means clustering based on mapreduce," CloudCom, pp. 674-679, 2009.

[3] D. Agrawal et al., "Challenges and Opportunity with Big Data," Community White Paper, pp. 01-16, 2012.

[4] X. Wu et al., "Data Mining with Big Data," IEEE Trans. Knowledge Data Engg., vol. 26, no. 1, pp. 97–107, 2014.

[5] X.-W. Chen et al., "Big data deep learning: Challenges and perspectives," IEEE Access Practical Innovations: open solutions, vol. 2, pp. 514 -525, 2014.

[6] M. A. Nadaf, S. S. Patil, "Performance Evaluation of Categorizing Technical Support Requests Using Advanced K-Means Algorithm," IEEE International Advance Computing Conference (IACC), pp. 409-414, 2015.

[7] "Big Data: Challenges and Opportunities, Infosys Labs Briefings - Infosys Labs," http://www.infosys. com/infosys-labs/publications/Documents/bigdata-challenges-opportunities.pdf.

[8] R. C. Bhagat, S. S. Patil, "Enhanced SMOTE algorithm for classification of imbalanced bigdata using Random Forest," IEEE International Advance Computing Conference (IACC), pp. 403-408, 2015.

[9] B. Chumphol, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safelevel- synthetic minority over-sampling technique for handling the class imbalanced problem," AKDD Springer Berlin Heidelberg, pp. 475-482, 2009.

[10] P. Byoung-Jun, S. Oh, and W. Pedrycz, "The design of polynomial function-based neural network predictors for detection of software defects," Elsevier: Journal of Information Sciences, pp. 40-57, 2013.

[11] N. Chawla, L. Aleksandar, L. Hall, and K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," PKDD Springer Berlin Heidelberg, pp. 107-119, 2003.

[12] R. Sara, V. Lopez, J. Benitez, and F. Herrera, "On the use of MapReduce for imbalanced big data using Random Forest," Elsevier: Journal of Information Sciences, pp. 112-137, 2014.

[13] H. Jiang, Y. Chen, and Z. Qiao, "Scaling up mapreduce-based big data processing on multi-GPU systems," SpingerLink Clust. Comput., vol. 18, no. 1, pp. 369–383, 2015.

[14] N. 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.

[15] S. Garcia et al., "Evolutionary-based selection of generalized instances for imbalanced classification," Elsevier: Journal of Knowl. Based Syst., pp. 3-12, 2012.

[16] H. Xiang, Y. Yang, and S. Zhao, "Local clustering ensemble learning method based on improved AdaBoost for rare class analysis," Journal of Computational Information Systems, Vol. 8, no. 4, pp. 1783-1790, 2012.

[17] H. Feng, and L. Hang, "A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE," Hindawi Mathematical Problems in Engineering, 2013.

[18] F. Alberto, M. Jesus, and F. Herrera, "Multi-class imbalanced data-sets with linguistic fuzzy rule based classification systems based on pairwise learning," Springer IPMU, pp. 89–98, 2010.

[19] J. Hanl, Y. Liul, and X. Sunl, "A Scalable Random Forest Algorithm Based on MapReduce," IEEE, pp.849-852, 2013.

[20] W. A. Rivera, O. Asparouhov, "Safe Level OUPS for Improving Target Concept Learning in Imbalanced Data Sets," Proceedings of the IEEE Southeast Con., pp. 1-8, 2015.

[21] S. Yen and Y. Lee, "Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset," ICIC 2006, LNCIS 344, pp. 731 – 740, 2006.

[22] J. Kwak, T. Lee, C. Kim, "An Incremental Clustering-Based Fault Detection Algorithm for Class-Imbalanced Process Data," IEEE Transactions on Semiconductor Manufacturing, pp. 318-328, 2015.

[23] S. Kim, H. Kim, Y. Namkoong, "Ordinal Classification of Imbalanced Data with Application in Emergency and Disaster Information Services," IEEE Intelligent Systems, pp. 50-56, 2016.

[24] M. Chandak, "Role of big-data in classification and novel class detection in data streams," Spinger Journal of Big Data, pp. 1-9, 2016.

[25] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, "DBSMOTE: Density-Based Synthetic Minority Over-Sampling TEchnique," Spinger Journal of Applied Intelligence, pp. 664-684, 2012.

[26] S. Patil, S. Sonavane, "Enhanced Over_Sampling Techniques for Handling Imbalanced Big Data Set Classification," book chapter for the book title: Data Science and Big Data: An Environment of Computational Intelligence published by Springer-Verlag, unpublised.