Performance Analysis of Under-Sampling and Over-Sampling Techniques for Solving Class Imbalance Problem

G.Rekha[a], *Amit Kumar Tyagi*[b], *V. Krishna Reddy*[a]

[a,b]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502.*
[b]*Department of Computer Science and Engineering, Lingaya's Vidyapeeth, Faridabad - 121002, Haryana, India*
gillala.rekha@klh.edu.in, amitkrtyagi025@gmail.com, vkrishnareddy@kluniversity.in

ARTICLE INFO

Article History:
Received 12 January 19
Received in revised form 20 January 19
Accepted 21 February 19

Keywords:
Class Imbalance Problem,
Sampling Techniques,
Earth Mover's Distance (EMD),
Synthetic Minority Oversampling Techniques (SMOTE).

ABSTRACT

Most of the traditional classification algorithms assume their training data to be well-balanced in terms of class distribution. Real-world datasets, however, are imbalanced in nature thus degrade the performance of the traditional classifiers. An imbalance data-set typically make prediction accuracy difficult. Data pre-processing approaches discuss this issue by using random under-sampling or oversampling techniques. To solve this problem, many strategies are adopted to balance the class distribution at the data level. The data level methods balance the imbalance distribution between majority and minority classes using either oversampling or under-sampling techniques. In this paper, we present the performance analysis of under-sampling method and oversampling methods. The methods are implemented with 5 conventional classifiers like C4.5 Decision Tree (DT), k-Nearest Neighbor (k-NN), Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Naive Bayes (NB) on 15 real life data sets. The experimental results show comparative study of under-sampling and over sampling technique.

Peer review under responsibility of International Conference on Sustainable Computing in Science, Technology and Management.

1. Introduction

Today most of the real-world problems are imbalanced in nature. In data mining and machine learning, the classifier trained based on imbalanced data-sets usually effects the learning model. It has been drawn significant attention from researchers in data mining, pattern classification, and machine learning disciplines (Elrahman, & Abraham, 2013). The imbalanced problem occurs when the class distribution of a given data-set is unequal between the data classes. A class with a large number of instances is considered as majority class or negative class and the class with few instances is considered as minority class or positive class. The class with few samples (minority class) may be ignored as noise and lead to false detection when trained by the traditional classifier (López, Fernández, García, Palade, & Herrera, 2013). These problems exist in a real-world domain such as financial crisis prediction, fraud detection, medical analysis, text classification, risk management and informational retrieval (Ramyachitra, & Manikandan, 2014). The different approaches to solve the class imbalance problem are broadly classified into four types: data-level methods, algorithmic-level methods, cost-sensitive learning and ensemble methods (Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2012) and (Beyan, & Fisher, 2015)

- *Data level approach* also known as an external approach. It employs pre-processing to re-balance the class distribution of imbalanced data sets. The pre-processing is done either by under-sampling or over-sampling techniques to reduce the imbalance ratio in the data set.
- *Algorithm level approach* also known as an internal approach. It modifies the classification algorithm to bias the learning towards the minority class. These algorithms require knowledge to learn from the imbalance data distribution before training the classifier.
- *Cost-sensitive learning approach* combines both data level and algorithm approaches to incorporate different mis-classification cost for each class.
- *Ensemble method* uses the ensembles of classifiers. It increases the accuracy of a classifier by training different classifiers and combines their result to generate a single class label.

Moreover, class imbalance involves a series of difficulties in learning such as small sample size, class overlapping, and small disjuncts. Specifically, the pervasive approach in taking care of the class imbalance problems is to utilize data level techniques. The primary focus of data level method is on applying pre-processing techniques to the imbalanced data-sets to balance it before building the classifiers. In this method, the data pre-processing and classifier training tasks are independent of each other. In addition, author in (Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2012) conducted a study on various well-known approaches which combines both pre-processing techniques at data-level and classifier ensembles. The results show better performance with ensemble methods. Data pre-processing techniques are based on data sampling approaches performed before the construction of classification model. (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and (Hu, Liang, Ma, & He, 2009). The under-sampling technique is applied on majority class and aim at eliminating the instances by applying any of the under-sampling techniques and balance the distribution in terms of a minority class. Oversampling techniques are applied to minority class to alleviate the samples size to balance the distribution.

1.1. Imbalanced Data Problem in Binary Classification with respect to Oversampling and Under sampling

This section discusses about the problem of class imbalance (in classification) using Oversampling and Undersampling techniques. At data level, a pre-processing technique is applied to balance the imbalanced data sets. The common oversampling approach is Random Over-Sampling (ROS) technique. ROS is a relatively simpler than the other techniques like Synthetic Minority Oversampling Technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Using SMOTE the new positive/minority samples are generated by interpolating several minority samples that lies closer to each other. A variant of SMOTE have been proposed in the literature. Chen at al. (Chen, Guo, & Chen, 2010) proposed a novel oversampling strategy called Cluster Indexing (CE)-SMOTE to handle imbalanced datasets. In this work, clustering Consistency Index (CI) used to find out the cluster boundaries of minority samples by using k- means algorithm and then oversampled these minority samples to match the original dataset. Koto et al. (Koto, 2014) proposed three improvements of SMOTE method named as SMOTE-Out, SMOTE-Cosine and Selected-SMOTE.

A number of well-known under-sampling approaches have been published in the literature. A straightforward and well-known under-sampling method is the Random Under-Sampling (RUS) (Japkowicz, 2000), which randomly discards majority class samples from the data-set until the imbalance effect is significantly lessened. Hart et al. (Hart, 1968) proposed an under-sampling technique called Condensed Nearest Neighbor (C-NN). It works by discarding the far away samples of majority class from the decision borderline by considering such samples as less appropriate for learning using a 1-NN rule. Tomek Links (TL) (Tomek, 1976) is another popular technique for under-sampling. It will eliminate the majority samples which are noisy and borderline by treating them as risky. In (Yen, & Lee, 2009), the author proposed a clustering-based under-sampling technique to hold the class distributions after pre-processing technique for both the minority and majority class data samples. Garcia et al. (García, & Herrera, 2009) presented a set of Evolutionary Algorithms (EAs) under-sampling methods. The proposed method efficiently does a prototypes selection for the majority class using fitness function based on accuracy and reduction rates for a given classification algorithm. The author in (Wong, Leung, & Ling, 2014) projected under-sampling method using the fuzzy rule to select the majority class instances from huge skewed class distribution. The sensitivity-based under-sampling method using clustering has been proposed (Ng, Hu, Yeung, Yin, & Roli, 2015). In this method, the class having more number of instances are grouped to indicate the within class distribution thus selected on the basis of a stochastic sensitivity measure of both positive and negative classes. Principal Component Analysis (PCA) based under-sampling method was proposed (Fu, Zhang, Bai, & Sun, 2016) to remove the redundant samples from majority class by using a wide-range of estimation model. Ha et al. in (Ha, & Lee, 2016) proposed an under-sampling technique based on Genetic Algorithm (GA). The performance of an original classifier is maximized by adopting GA to reduce a loss function between the actual and under-sampled instances from the majority class. The author in (Devi, & Purkayastha, 2017) proposed an extension to Tomek link under-sampling technique. The proposed method eliminates the outlier, redundant and noisy instances in majority samples by considering them as least influence for estimation of class label accuracy.

Apart from the above techniques, the hybrid method combines different sampling-based approaches with the algorithmic methods. These techniques integrate various under-sampling and over-sampling with ensemble classifier. The classical examples are SMOTEBoost (Chawla, Lazarevic, Hall, & Bowyer, 2003), RUSBoost (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010), OverBagging (Wang, & Yao, 2009), and UnderBagging (Barandela, Valdovinos, & Sánchez, 2003, Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2012) suggested that ensemble-based algorithms provide better performance results than that attained by the using data pre-processing techniques trained on a single classifier.

Hence, this section discusses about introduction part related to oversampling and undersampling techniques. Together this, the organization of this work is planned as (for future sections): Section 2 discusses reason behind working related to this approach or class imbalance problem. Further, our proposed algorithm is discussed with undersampling and oversampling techniques in section 3. Later, experimental results are discussed in detail in section 4 and in last, this work is concluded with some future work/remarks in section 5.

2. Motivation

Class imbalance problem is “a serious problem in machine learning, where the total number of a class of data (positive/ minority class) is far less than the total number of another class of data (negative/ majority class)” (Rekha, Krishna Reddy, & Tyagi, 2018). In this problem, data set of one class is far less than other class. This problem is very interesting and received attention from several research communities in the past decades. So, solving this problem or balancing data set among class is a must require research. Re-sampling techniques such as undersampling, oversampling and hybrid method are used for generating synthetic data. However, most of the oversampling techniques at data level may generate data samples very much similar to existing samples by considering only the nearest neighbor samples. To overcome these problems, we propose a novel approach to solve the class imbalance problem at the data level. The main motivation behind this method is to balance the training data by removing noise lying in the data in the form of outliers after Synthetic Minority Oversampling Techniques (SMOTE). SMOTE generates synthetic data using k-nearest neighbor algorithm (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), (Batista, Prati, & Monard, 2004), (Mahalanobis, 1936). This technique selects the data instances that are the nearest neighbors using Euclidean distance. After synthetic sample generation, noise and outliers usually present in the data instances. However, selecting only those data instances that are nearer may pose the potential challenge of generating noise and sparse data instances.

Here, the main motivation of this work is to make balance between both classes (until the classes are represented in a more balanced way). This work aims is to reduce false negative as much as possible. This work eliminates the samples of majority class that are redundant and balances the class distribution using under-sampling for majority class and oversampling methods to generate the synthetic data for minority class. Here, we present an under-sampling method using EMD to get rid of redundant samples in majority class. Also, SMOTE with Mahalanobis distance (Mahalanobis, 1936) which is known to be useful for identifying outliers for minority samples. In summary, this section investigates reasons behind working towards to solve class imbalance problem. Now, next section will discuss our proposed approaches/ methods in detail.

3. Proposed Methodology

This section discusses the proposed methodology to handle the problem of class imbalance in classification with respective to Undersampling and Oversampling techniques.

3.1. Under-Sampling Technique Based on Earth Mover's Distance

This work uses Earth Mover's Distance (EMD) based under-sampling technique to compute the similarity existing in the majority samples data sets and eliminate them as redundant. The EMD method is a popular distance method often used in computer vision (Rubner, Tomasi, & Guibas, 1998). It finds the dissimilarity between two multi-dimensional distributions of data. The EMD describes the cost that must be paid to convert from one distribution into the other. The EMD measures the least amount of effort needed to fill the holes with earth. It is a linear optimization technique applied to the transportation problem. For this problem, the EMD finds the least expensive flow required to move from one distribution to another according to some given constraints. The solution to EMD is based on well-known transportation problem, first introduced by Monge (Zadrozny, & Elkan, 2001). Suppose there are several suppliers, each with a given amount of goods and required to supply for several consumers, each with a given limited capacity. The transportation problem is to find a least-expensive flow of goods to be moved from the suppliers to the consumers that satisfy the consumer's demand. The majority samples of the imbalanced data-set can be represented as a transportation problem by defining one instance as the supplier and the other as the consumer. Intuitively, the solution is then to find the minimum amount of "work" needed to transform one instance into the other. The cost for a supplier-consumer pair to equal the distance between an element in the supplier and an element in the consumer. Intuitively, the solution is to find the minimum amount of work needs to transform one element into the other. This can be formalized as the following linear programming problem. Let R = {(r1, wr1), ...,(rm, wrm)} be the supplier with m clusters, where ri is the cluster representative and wrm is the weight of the cluster; S= {(s1, ws1), . . . ,(sn, wsn)} the consumer with n clusters; and D = [dij] the distance matrix where dij is the distance between cluster ri and si . We want to find a flow F that minimizes the overall cost between ri and si . The flow F fij between ri and si is computed by Eq.1.

$$\text{FLOW } (\mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij}\, f_{ij}$$

and subject to the following constraints:

$$\left(\begin{array}{c} f_{ij} \geq 0 \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \\ \sum_{j=1}^{m} f_{ij} \leq w_{ri}, \quad 1 \leq i \leq m \\ \sum_{i=1}^{n} f_{ij} \leq w_{si}, \quad 1 \leq i \leq n \\ \sum_{j=1}^{m} \sum_{i=1}^{n} f_{ij} = min\left(\sum_{j=1}^{m} w_{ri}, \sum_{j=1}^{m} w_{si}\right) \end{array} \right)$$

The first constraint allows moving of elements from R to S only and not vice-versa. The next two constraints limit the amount of mass which can be sent from the elements in R not exceed the weight values and to the amount which can be received by elements in S (again limited by the weights). The last constraint forces to move the maximum amount of mass possible. Once the transportation problem is solved and we have to compute the optimal flow F, the EMD is defined as the work normalized by the total flow:

$$\mathrm{EMD}(R, S) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} \cdot f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

The EMD is a robust method and naturally extends the notion of a distance between single elements to that of a distance between distribution of elements. In this work, we use the EMD distance to measure the similarity between the two representations of majority samples. We denote the training data set T, and the majority class instances set Q = {x1, .., xn}, where each xi consists of m attributes. We use EMD to calculate the similarity between the majority samples by representing each instance, Qi as a matrix. The EMD between two matrices such as Qi and Qj with M columns are calculated as a sum of EMD between each column in the source matrix and the corresponding column in the target matrix. If EMD [Qi , Qj] = 0, then the two instances are completely identical, and if EMD [Qi , Qj] = 1, then the two instances are completely different. Based on the EMD if the distance between the two instance tends to be 0, we eliminated one instance by considering it as redundant.

3.1.1. Proposed framework with respect to Undersampling

Figure 1 shows the procedure for EMD-based under-sampling approach. The objectives adopted in the proposed framework are: a) redundancy b) noise c) outliers. The main motivation is to eliminate higher redundancy and noise while under-sampling is performed. The proposed framework focuses on the skewed distribution of data with binary class labels. Consider binary class imbalanced data-set D with skewed class distribution. The majority and minority class contain M observations with N features or data points respectively. The framework presented in two phases: (i) Data Pre-processing phase (ii) Classification phase. Initially, the training data-set is input to the data pre-processing phase. This phase performs EMD-based under-sampling of the majority class by eliminating the instances which are redundancy, noise, and outliers. The pre-processing phase is described as follows. The imbalanced data-set is divided into training set and test set in the first phase. Secondly, split the training samples into two subsets each of majority and minority class. The third step is to employ the EMD-based undersampling method.

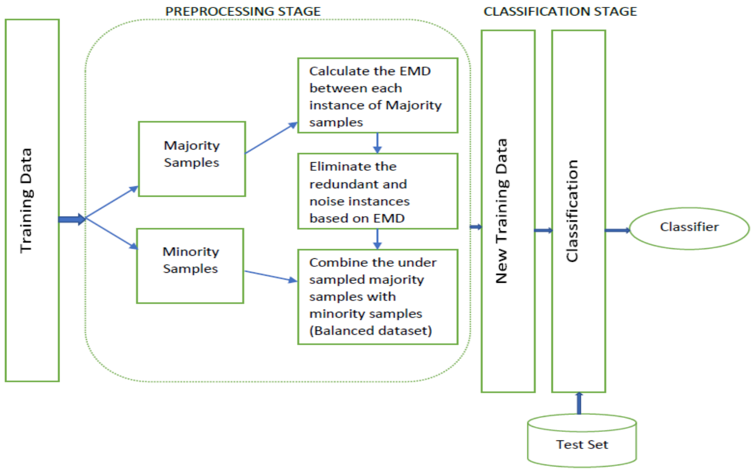
PREPROCESSING STAGE
CLASSIFICATION STAGE
Training Data
Majority Samples
Minority Samples
Calculate the EMD between each instance of Majority samples
Eliminate the redundant and noise instances based on EMD
Combine the under sampled majority samples with minority samples (Balanced dataset)
New Training Data
Classification
Classifier
Test Set

Figure 1: Proposed EMD - Based Undersampling Approach

The EMD-based undersampling technique identifies the redundant samples in the majority class and eliminates them. It leads to a smaller majority subset without redundant samples. Now, we combine the resultant majority samples subset with that of minority class subset. In classification phase, the revised training data are classified by using the different classification algorithms and then evaluated against the test set. The different classification algorithms used in this study are Decision tree (c4.5), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Naive Bayes (NB) and Multilayer Perceptron (MLP).

3.2. Over-Sampling Technique Based on SMOTE-MD

Synthetic Minority Oversampling Method (SMOTE) generates synthetic data using the k-nearest neighbor algorithm (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and (Mahalanobis, 1936). This technique selects the data instances that are the nearest neighbors using Euclidean distance. After synthetic sample generation, some problems usually present in the data instances. However, selecting only those data instances that are nearer to the existing samples may pose the potential challenge of generating noise and sparse data instances. Further, the data samples generated after resampling were in some samples may tend to fall outside the boundary of minority class as outliers which may lead to bias in classification. In our proposed method Mahalanobis distance is used to remove outliers appeared in the data after generating the synthetic samples. Mahalanobis distance measure is considered as unit-less measure and provides a relative measure of an instance distance and helps in detecting outliers.

Considering two data instances $x = (x_1, x_2, x_3, \ldots., x_n)'$ and $y = (y_1, y_2, y_3, \ldots., y_n)'$, the Mahalanobis distance between them is defined as $d_M(x, y) = \sqrt{(x - y)^T S - 1(x - y)}$, where S − 1 is the covariance matrix. We use this measure to help rank and sort the data samples according to their distance in a decreasing order. By sorting the data, we are able to distinguish data samples that are far or close from the central data instance. It works well for multivariate datasets and also overcomes the inherent scale and correlation problems, associated with Euclidean distance. The removal of outlier samples might provide a better performance on classifiers.

3.2.1. Proposed framework with respect to Oversampling

The intuition behind our approach is to remove the outliers existing in the data samples after generating the synthetic samples for minority classes. Our proposed approach is compared with the common Synthetic Minority Oversampling Techniques (SMOTE) (by Chawla et al. (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). The SMOTE technique oversamples the minority classes by generating synthetic data by introducing data samples along the line segments that join any of the k nearest neighbors minority class sample.

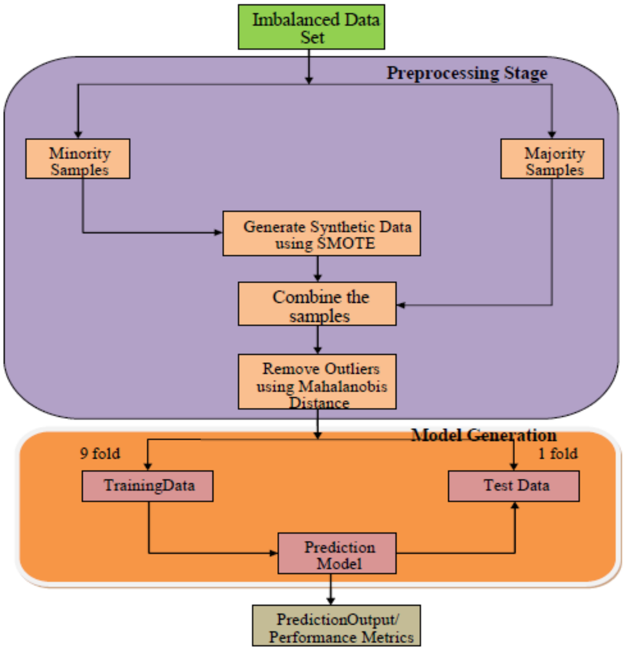

Figure 2: Proposed Framework for Oversampling (Rekha, & Krishna Reddy 2018)

Our proposed method comprises two stages, i.e., Stage one is pre-processing stage and the second stage is for model generation. To generate the synthetic data for minority samples, in the first stage we divided the data samples into minority and majority data samples based on their class label. Then, for minority samples, we generated synthetic data using SMOTE. The synthetic data samples are generated to balance the class samples. Then, we combine the data samples of both minority and majority class which represent balance data sets. Now, we find the outliers in the data and measure the diversity in the data sets using Mahalanobis distance (Mahalanobis, 1936). This measure is adopted because it works well to eliminate the diversity existing in the data. It is adopted because of its multivariate effect size. Using this measure, we generate the ranks and sort the data instances in decreasing order to their distance. We eliminate the data samples that are far or close from the center and consider the remaining data samples for the next stage. In the second stage, we performed model generation using the data samples produce from the first stage. We trained the data samples on various classification algorithms as mentioned in Table 2. We employed 10-fold cross-validation techniques, were in 2/3 of the data samples are picked randomly as training data and the remaining 1/3 samples as the testing data. The framework of the proposed method is showed in Figure 2.

Hence, this section discusses about our ideas (and framework) with respect to undersampling and oversampling techniques (as our proposed method). Now, next section will discuss about simulation results (with used data description) with respect to undersampling and oversampling techniques.

4. Experimental Simulation and Results

4.1. Data set description (used in this work)

We choose 15 imbalanced data sets for the experiment, obtained from Keel Repository (Alcalá-Fdez, Fernández, Luengo, Derrac, García, Sánchez, & Herrera, 2011). Table 1 lists the name of the data-sets with their Imbalance Ratio (IR), a total number of instances and the number of features.

Table 1. Datasets Used in the Experiment

Dataset	# of Attributes	IR	Total
glass1	9	1.82	214
glass6	9	6.38	214
Haberman	3	2.78	306
iris0	4	2	150
new_thyroid 1	5	5.14	215
new_thyroid 2	5	5.14	215
Pima	8	1.87	768
vehicle0	18	3.25	846
Wisconsin	9	1.86	683
vehicle1	18	2.52	846
vehicle2	18	2.52	846
vehicle3	18	2.52	846
yeast1	8	2.46	1484
yeast3	8	8.11	1484
page-blocks0	10	8.77	5472

These are all binary class classification data-sets. For training and testing the classifier, all the data-sets were divided into 80% training and 20% testing sets and adopted the ten-fold cross-validation approach. The overall classification accuracy is not a good metric for imbalanced data sets, because a traditional classifier may predict every case as the majority class, lead to higher accuracy in a skewed distribution. In our experiment, we used Precision, Recall, F-Measure, G-Mean and ROC (AUC the area under the Receiver Operating Characteristic (ROC) curve as the evaluation metric.

4.2. Classification Algorithms (used in this work)

The performance evaluation of proposed framework for both undersampling and oversampling technique is compared with five state-of-the art approaches such as Decision Tree (c4.5), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Multilayer Perceptron (MLP) and the Naive Bayes (NB). The experiments were all conducted using R (Team, 2013), an open source statistical tool in windows 7 environment. Table 2 presents the list of parameters used in c4.5, k-NN, SVM, NB, and MLP classification models.

Table 2. Classification Algorithms Used and the list of parameters

Algorithm Name	Classifier	Parameter Name	Parameter Value considered
K-Nearest-Neighbor algorithm	k-NN	K	5
		Rule	Nearest
		Distance	Euclidean
Support Vector Machine	SVM	Method	Least squares (LS)
		Kernel function	Gaussian Radial Basis function
		Scaling factor	0.1

Nave Bayes algorithm	Naïve Bayes	Prior	Uniform, Emperical
		Distribution	Kernel
Multi-layer perceptron	MLP	Hiddenlayers	attribs + classes
		Learningrate	0.3
Decision tree algorithm	C4.5	Confidencefactor	0.25
		Reduced-Error Pruning.	3

1.1 Evaluation Metrics (used in this work)

In this section, we present the brief description of metrics used. Accuracy could provide a reasonable measure of classifier performance on balanced datasets. Table 3 illustrates the confusion matrix (comprises four entries) for binary class problems. It comprises a number of TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). In our experiment, positive instance refers to minority class and negative instance refer to majority class. The confusion matrix provides information about the actual and predicted values after classification. However, the classifier performance is evaluated based on the confusion matrix.

Table 3. Confusion Matrix

	Positive Prediction	**Negative Prediction**
Positive class	TP	FN
Negative class	FP	TN

Note that True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) have their respective meaning which has been discussed in (Rekha, & Krishna Reddy 2018). However, for class imbalance data sets the evaluation of the classification results should take into account the performance of both the minority and majority classes. If a classifier is overwhelmed by the majority class, then it may classify all instances as majority classes. If majority class instances comprise 90% of the evaluation set then the classifiers may obtain 90% accuracy. Therefore, several metrics are derived from the confusion matrix to handle the imbalanced data sets. Also, this work used Precision, Recall and F-score metrics for measuring performance of (our) proposed approaches/ to make comparative performance analysis. Hence, readers are requested to go through our work (Rekha, Krishna Reddy, & Tyagi, 2018) for knowing more about above (precision, recall and F-score) metrics. Apart that, this work also used ROC curve and Area under Curve metrics in our/ this work, which are include as:

- ROC curve: A ROC curve for a given classifier (Błaszczyński, & Stefanowski, 2015) shows the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). TPR and FPR are the two operating characteristics being compared. TPR is the proposition of positive tuples that are correctly labeled by the classifier. FPR is the proposition of negative tuples that are misclassified as positive.
- Area under ROC curve: Receiver Operating Characteristic (ROC) curves are useful for assessing the accuracy of predictions (Hart, 1968). It is a two-dimensional graph in which x-axis represents the FPR or probability of false alarm and y-axis represent the TRP or probability of detection. It plots curve between PD (Probability of Detection) and false alarm rate (PF).

A ROC curve can be defined as:

a. Point (0, 0) denotes that it would never issue a positive classification nor it never triggers a false alarm.
b. Point (0, 1) denotes the ideal position.
c. Any point between (0,0) and (1,1) contain no information.

G-mean: It is the geometric mean of recall and precision, i.e., G-Mean = $\sqrt{recall * precision}$
The overall performance is measured by Precision, Recall, F-Measure, G-Mean and ROC (AUC) in our experiment.

1.2 Simulation Results

Our proposed scheme is tested with 15 datasets and five conventional classifiers mentioned in Section 4.2. The validated of the proposed framework and the baseline approaches are done by 10-fold cross-validation. The performance is report using accuracy, precision, recall, AUC and f-measure. Table 4 and Table 5 shows the Classification accuracy of traditional classifiers using EMD-undersampling approach and SMOTE-MD oversampling approach. Further, the performance of proposed methods in terms of Precision, Recall, F-Measure, G-Mean and AUC has shown in Table 6. Based on the experiment, we observed that our proposed algorithm shows significant improvement on C4.5, k-Nearest Neighbor (K-NN) and Naive Bayes (NB) algorithms (in terms of Precision, Recall, F-Measure, G-Mean and AUC).

Table 4. Classification Accuracy of Traditional Classifiers using EMD-Undersampling Approach

Dataset Name	**C4.5**	**SVM**	**NB**	**MLP**	**K-NN**
glass1	100%	99.09%	97.73%	97.73%	100%
glass6	94.20%	93.17%	90.10%	88.74%	94.88%
Haberman	95.76%	97.88%	91.87%	96.47%	96.11%
iris0	94.23%	91.35%	88.78%	89.10%	95.83%
new_thyroid 1	92.49%	85.92%	74.65%	62.91%	91.55%

new_thyroid 2	94.63%	90.24%	93.66%	92.20%	95.61%
Pima	89.67%	85.92%	63.85%	62.91%	84.04%
vehicle0	96.94%	94.90%	97.96%	96.94%	96.94%
Wisconsin	77.01%	75.86%	72.80%	79.69%	78.93%
vehicle1	98.95%	100%	100%	100%	100%
vehicle2	98.10%	98.10%	93.84%	98.10%	99.05%
vehicle3	96.63%	97.60%	92.79%	96.63%	99.04%
yeast1	99.15%	97.00%	94.49%	91.25%	96.87%
yeast3	81.92%	77.06%	76.79%	76.65%	80.03%
page-blocks0	95.86%	93.73%	95.15%	66.39%	98.70%

Table 5. Classification Accuracy of Traditional Classifiers using SMOTE-MD Oversampling Approach

Dataset Name	C4.5	SVM	NB	MLP	K-NN
glass1	100%	99.09%	97.73%	97.73%	100%
glass6	98.20%	92.17%	91.10%	88.74%	94.88%
haberman	95.76%	97.88%	91.87%	96.47%	97.21%
iris0	94.23%	91.35%	88.78%	90.12%	95.83%
new_thyroid 1	92.49%	85.92%	74.65%	62.91%	91.55%
new_thyroid 2	94.63%	90.24%	93.66%	92.20%	95.61%
pima	89.67%	85.92%	63.85%	62.91%	84.04%
vehicle0	96.94%	94.90%	97.96%	96.94%	96.94%
Wisconsin	77.01%	75.86%	72.80%	79.69%	78.93%
vehicle1	99.95%	100%	100%	100%	100%
vehicle2	98.10%	98.10%	93.84%	99.10%	99.05%
vehicle3	96.63%	97.60%	92.79%	96.63%	99.04%
yeast1	99.15%	97.00%	95.49%	91.25%	96.87%
yeast3	81.92%	77.06%	78.79%	78.65%	80.03%
page-blocks0	95.86%	93.73%	95.15%	86.39%	98.70%

Table 6. Performance of Traditional Classifiers using EMD-undersampling and SMOTE-MD Oversampling Approach

Algorithm	Name of study	Proposed Approach	
		EMD	SMOTE+MD
C4.5	Precision	0.889556	0.891444
	Recall	0.888889	0.890667
	f-measure	0.888778	0.890444
	G-mean	0.889222	0.891
	AUC	0.898111	0.900778
kNN	Precision	0.895	0.897667
	Recall	0.894222	0.897333
	f-measure	0.894111	0.897222
	G-mean	0.894556	0.897556
	AUC	0.932778	0.937778
NB	Precision	0.863222	0.874778
	Recall	0.847667	0.858556
	f-measure	0.842667	0.853222

	G-mean	0.855111	0.866444
	AUC	0.878333	0.887
MLP	Precision	0.889333	0.892333
	Recall	0.889	0.892222
	f-measure	0.889	0.892111
	G-mean	0.889	0.892333
	AUC	0.897222	0.908111
SVM	Precision	0.877111	0.902222
	Recall	0.863889	0.887667
	f-measure	0.854111	0.882444
	G-mean	0.870222	0.894778
	AUC	0.863889	0.887667

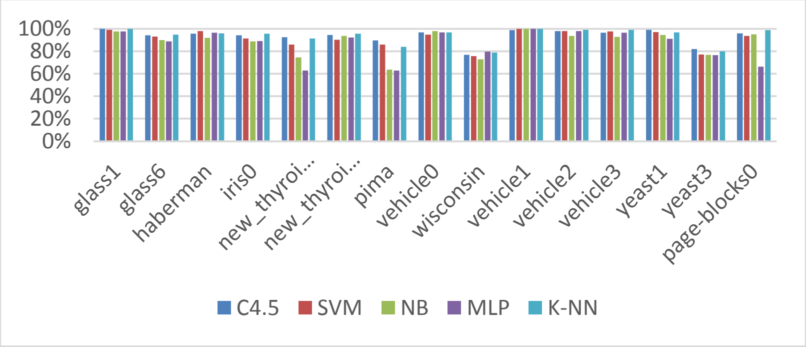

Figure 3: **Classification accuracy of traditional Classifiers using EMD-undersampling approach**

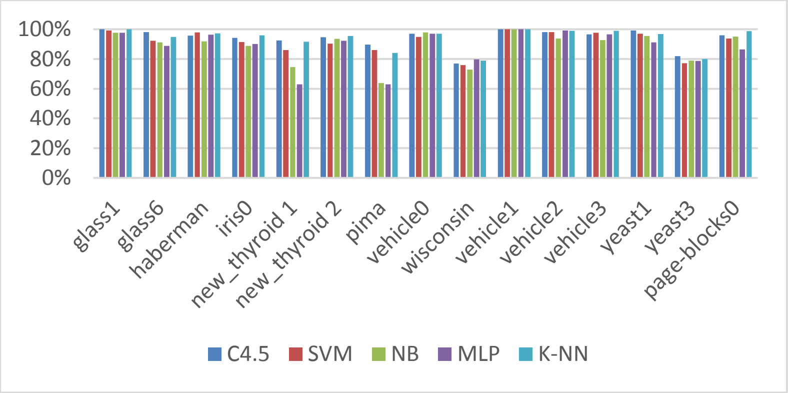

Figure 4: **Classification accuracy of traditional Classifiers using SMOTE-MD Oversampling approach**

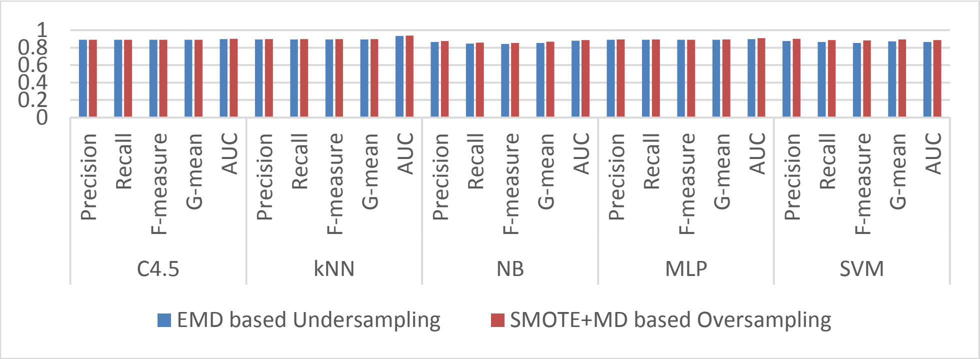

Figure 5: Performance of EMD based Undersampling and SMOTE-MD based Oversampling approach in terms of Precision, Recall, F-measure, G-mean and AUC.

Hence, this section discusses comparison/ performance analysis of EMD and SMOTE-MD algorithms separately in figure 3-4 and then together in figure 5. We find that both oversampling and under sampling perform better but according to required datasets. In summary, undersampling performs better than oversampling technique for similar data sets (to balance data sets) according to computation costs, but for different data sets this assumption fails and oversampling work better than undersampling techniques. Now, next section will concludes this work in brief.

5. Conclusion

During processing/ removing class imbalance problem or balancing datasets, this work used undersampling and oversampling techniques on 15 datasets. After performing several comparison with metrics f- measure, or recall, we find that using such techniques discard useful data which may essential for the learning process (for future). Also we found that, Oversampling technique takes longer training time and inefficiency (in terms of memory, due to the increased number of training instances) than undersampling technique and it suffers from high computational costs (for pre-processing the data). Hence, we reached to a conclusion that undersampling perform better than oversampling technique for similar data sets (to balance a data sets). In summary, this work presents a performance analysis of undersampling and oversampling techniques on 15 data sets (received or collected from UCI repository). Our Experimental results (performed on numerical datasets) with figure 3-5, shows that our method can achieve a significant decrease in the training time, while maintaining the same or achieving even higher g-means values by using less number of training instances using undersampling method over oversampling method/ technique. Hence for future work, we kindly invite all data mining researchers/ researchers interested to this (class imbalance) problem, to find an optimal and efficient solution for solving class imbalance problem.

Conflicts of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgements

This research is funded by the Koneru Lakshmaiah Education Foundation, Lingaya's Vidyapeeth and Anumit Academy's Research and Innovation Network (AARIN), India. The authors would like to thank Koneru Lakshmaiah Education Foundation, Lingaya's Vidyapeeth and AARIN, India, an education foundation body and a research network for supporting the project through its financial assistance.

References

Elrahman, S. M. A., & Abraham, A. (2013). A review of class imbalance problem. *Journal of Network and Innovative Computing*, *1*(2013), 332-340.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, *250*, 113-141.

Ramyachitra, D., & Manikandan, P. (2014). Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, *5*(4).

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463-484.

Beyan, C., & Fisher, R. (2015). Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition*, *48*(5), 1653-1672.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Hu, S., Liang, Y., Ma, L., & He, Y. (2009, October). MSMOTE: improving classification performance when training data is imbalanced. In *2009 second international workshop on computer science and engineering* (Vol. 2, pp. 13-17). IEEE.

Japkowicz, N. (2000, June). The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*.

Chen, S., Guo, G., & Chen, L. (2010, April). A new over-sampling method based on cluster ensembles. In *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops* (pp. 599-604). IEEE.

Koto, F. (2014, October). SMOTE-Out, SMOTE-Cosine, and Selected-SMOTE: An enhancement strategy to handle imbalance in data level. In *2014 International Conference on Advanced Computer Science and Information System* (pp. 280-284). IEEE.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003, September). SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery* (pp. 107-119). Springer, Berlin, Heidelberg.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *40*(1), 185-197.

Wang, S., & Yao, X. (2009, March). Diversity analysis on imbalanced data sets by using ensemble models. In *2009 IEEE Symposium on Computational Intelligence and Data Mining* (pp. 324-331). IEEE.

Barandela, R., Valdovinos, R. M., & Sánchez, J. S. (2003). New applications of ensembles of classifiers. *Pattern Analysis & Applications*, *6*(3), 245-256.

Błaszczyński, J., & Stefanowski, J. (2015). Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing*, *150*, 529-542.

Hart, P. (1968). The condensed nearest neighbor rule (Corresp.). *IEEE transactions on information theory*, *14*(3), 515-516.

Tomek, I. (1976). Two modifications of CNN. *IEEE Trans. Systems, Man and Cybernetics*, *6*, 769-772.

Yen, S. J., & Lee, Y. S. (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, *36*(3), 5718-5727.

García, S., & Herrera, F. (2009). Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation*, *17*(3), 275-306.

Wong, G. Y., Leung, F. H., & Ling, S. H. (2014, July). An under-sampling method based on fuzzy logic for large imbalanced dataset. In *2014 ieee International Conference on Fuzzy Systems (fuzz-ieee)* (pp. 1248-1252). IEEE.

Ng, W. W., Hu, J., Yeung, D. S., Yin, S., & Roli, F. (2015). Diversified sensitivity-based undersampling for imbalance classification problems. *IEEE transactions on cybernetics*, *45*(11), 2402-2412.

Fu, Y., Zhang, H., Bai, Y., & Sun, W. (2016, August). An Under-sampling Method: Based on Principal Component Analysis and Comprehensive Evaluation Model. In *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 414-415). IEEE.

Ha, J., & Lee, J. S. (2016, January). A new under-sampling method using genetic algorithm for imbalanced data classification. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication* (p. 95). ACM.

Devi, D., & Purkayastha, B. (2017). Redundancy-driven modified Tomek-link based undersampling: a solution to class imbalance. *Pattern Recognition Letters*, *93*, 3-12.

Rubner, Y., Tomasi, C., & Guibas, L. J. (1998, January). A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)* (pp. 59-66). IEEE.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, *17*.

Team, R. C. (2013). R: A language and environment for statistical computing.

Zadrozny, B., & Elkan, C. (2001, August). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 204-213). ACM.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), 20-29.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.

Rekha, G, Krishna Reddy, V., & Tyagi, A. K. (2018). A Novel Approach to solve class imbalance problem using Noise Filter method, in: *Proceedings of the 18th International Conference on Intelligent Systems Design and Applications (ISDA), VIT Vellore*.

Rekha, G, Krishna Reddy K, (2018). A novel approach for handling outliers in imbalance data, International Journal of Engineering & Technology, (pp. 1-5).