Prediction Model Framework for Imbalanced Datasets

Maria Rossana C. de Leon
Southern Luzon State University
Lucban, Quezon Philippines
e-mail: rossana.4481@gmail.com

Eugene Rex L. Jalao
University of the Philippines Diliman
Quezon City Philippines
e-mail: eljalao@upd.edu.ph

***Abstract*—Generally, prediction requires significant and good quality input data that will give accurate prediction. However, real-data are often noisy, inconsistent, and imbalanced. If the classes are imbalanced, the class accuracy is unlikeable because the prediction tends to favor those in majority class since it has relatively significant class size. To resolve the imbalance problem, a resampling algorithm is proposed which improves the prediction accuracy of each class. The algorithm was tested in 4 different datasets each using different prediction and classification methodologies, such as Regression Analysis, Decision Tree, Rule Induction, and Artificial Neural Networks. Results show that the framework works in either methodologies and prediction accuracy generally improves after resampling. The framework was also compared to the existing sampling methodologies and results show that it is comparable with the ROS/RUS, but the resampling rate is minimized with the proposed framework.**

Keywords:Prediction model framework; Class imbalance problem; Data mining.

I. INTRODUCTION

Prediction, pattern recognition, and classification problems are not new. By definition, predictive analytics is the utilization of statistics, data mining, and game theory to analyze current and historical facts in order to make predictions about future events [1]. It enables decision makers to develop mathematical models to help them better understand the relationship among variables.

One of the major issues in coming up with accurate predictions lies in the quality of input data, which are usually *incomplete* (lacking attribute values or certain attributes of interest, or containing only aggregate data), *noisy* (containing errors, or *outlier* values that deviate from the expected), *inconsistent* (e.g., containing discrepancies in the department codes used to categorize items), and *imbalanced* (occurs when one class is underrepresented in the data set). Accordingly, low quality data will lead to low quality prediction and classification results [2]. This research mainly focuses on addressing imbalanced datasets.

Generally, a two-class data set is said to be imbalanced (or skewed) when one of the classes, called the *minority class*, is heavily under-represented in comparison to the other class, called the *majority class*. Dataset imbalance on the order of 100 to 1 is prevalent in fraud detection and imbalance of up to 100,000 to 1 has been reported in other applications [3]. In such a situation, most of the classifiers are biased towards the major classes and hence show very poor classification rates on minor classes. It is also possible that a classifier predicts everything as major class and ignores the minor class [4]. Class distribution, i.e., the proportion of instances belonging to each class in a data set, plays a key role in classification. Data sets with skewed class distribution usually tend to suffer from class overlapping, small sample size or small disjuncts, which difficult classifier learning [5]. Furthermore, the evaluation criterion can lead to ignore minority class examples (treating them as noise) and hence, the induced classifier might lose its classification ability in this scenario. In many applications, misclassifying a rare event can result in more serious problem than common event. For example, in medical diagnosis in case of cancerous cell detection, misclassifying non-cancerous cells leads to some additional clinical testing but misclassifying cancerous cells leads to very serious health risks. However, in classification problems with imbalanced data, the minority class examples are more likely to be misclassified than the majority class examples, due to their design principles; most of the machines learning algorithms optimize the overall classification accuracy which results in misclassification minority classes.

Various studies have already been conducted that answers class imbalance problem. Yet, the techniques are either simple or complex. Simple techniques use either oversampling or undersampling or combination of both, but the techniques usually assume that a fully balanced dataset can be attained. With this assumption, the chance of over-fitting, particularly for oversampling, is highly possible. In the case of undersampling, too much useful data that are excluded in the training set can make the prediction or classification inaccurate. Multi-dimensional (*n*-dimension) data sets can be resolved using more sophisticated techniques which can be hard to interpret and would require a significant amount of processing cost.

Against this background, there is a need to develop a prediction model framework that can pre-process and resolve the imbalance problem by utilizing a proposed iterative oversampling and undersampling methodology for *n*-dimensional datasets.

We begin by presenting related works of others in Section 2. In Section 3, we describe the proposed framework and the algorithm to resolve the class imbalance

 ISBN: 978-1-61208-358-2

problem. Then, the results of the experimental runs and the analyses are discussed in Section 4. Finally, conclusion and areas for future studies are described in Section 5.

II. RELATED WORKS

Large numbers of approaches have previously been proposed to deal with the class-imbalance problem [6]. The approaches are categorized into two groups: the *internal approaches acting on the algorithm* that create new algorithms or modify existing ones to take the class-imbalance problem into consideration, and *external approaches acting on the data* that use unmodified existing algorithms, but resample the data presented to these algorithms so as diminish the effect caused by their class imbalance [1]. Internal approaches modify the learning algorithm to deal with the imbalance problem. They can adapt the decision threshold to create a bias toward the minority class or introduce costs in the learning process to compensate the minority class. Cost sensitive learning is probably the most well-known method of dealing with the class imbalance [3]. External approaches act on the data instead of the learning method. They have the advantage of being independent from the classifier used.

Sampling is the most popular means for overcoming the class imbalance problem [3]. Sampling is used as a means of altering the distribution of the minority class so that it is not under represented when training the learner. There are three basic approaches to overcome the class imbalance problem. These are over sampling of the minority class, under sampling of the majority class or the use of a hybrid approach based on both.

A. Random Over-sampling (ROS)

ROS [3] can be described as the random sampling of the minority class with replacement. This randomly samples with replacement the minority class and adds them to the minority class sample set until the size of the minority class is the same size as the majority class. With replacement means that after each resample, the samples are placed back in the 'pot' (the minority sample set) and can be resampled again. Over-sampling can result in a number of problems, these include over-fitting of a model especially in the cases of noisy data. Also, over-sampling does not result in more information being included in the training set, which can cause the production of overly complex models.

B. Random Under-sampling (RUS)

When under sampling [11] is used to overcome the problem of class imbalance, the number of majority class examples is reduced until the number of majority samples equals the number of minority samples. When using this solution to the class imbalance problem, certain problems may arise from removing such a larger number of the majority class, due to a fact that a number of potentially useful samples from the majority class may be discarded. Under-sampling does offer a number of benefits to over-sampling. The main one being that it results in a smaller training set as compared to oversampling, thus resulting in shorter training times.

C. Combination of ROS and RUS (ROS/RUS)

ROS/RUS is a combination of random over-sampling and random under-sampling. When this approach is used, the majority class would be under-sampled and the minority class would be over-sampled.

D. Synthetic Minority Over-sampling Technique (SMOTE)

The use of SMOTE algorithm [9] to artificially synthesize items belonging to the minority class has also been postulated as a means of overcoming the class imbalance problem. When SMOTING a dataset, the class having the smaller number of examples is over-sampled through the synthesis of artificial instances, as opposed to over-sampling existing samples with replacement. The class having the smaller number of examples is over-sampled by the use of a *kNN* to add artificial instances along the line segments connecting some or the entire population of nearest neighbors. The number of nearest neighbors to add is dependent on the level of over-sampling required.

E. Cost-sensitive Learning

Cost-sensitive learning framework incorporates both data level transformations (by adding costs to instances) and algorithm level modifications (by modifying the learning process to accept costs) [1]. It biases the classifier toward the minority class the assumption higher misclassification costs for this class and seeking to minimize the total cost errors of both classes. The major drawback of these approaches is the need to define misclassification costs, which are not usually available in the data sets [5].

Several studies that used the abovementioned approaches have been done to answer the class imbalance problem. Some of which are discussed below:

Kubat and Matwin [4] selectively under-sampled the majority class while keeping the original population of the minority class. The minority examples were divided into four categories: some noise overlapping the positive class decision region, borderline samples, redundant samples and safe samples. The borderline examples were detected using the Tomek links concept.

Japkowicz [8] discussed the effect of imbalance in a dataset. She evaluated three strategies: under-sampling, resampling and a recognition-based induction scheme. She experimented on artificial 1D data in order to easily measure and construct concept complexity. Two resampling methods were considered. Random resampling consisted of resampling the smaller class at random until it consisted of as many samples as the majority class and focused resampling consisted of resampling only those minority examples that occurred on the boundary between the minority and majority classes. Random under-sampling was considered, which involved under-sampling the majority class samples at random until their numbers matched the

 ISBN: 978-1-61208-358-2

number of minority class samples; focused under-sampling involved under-sampling the majority class samples lying further away. She noted that both the sampling approaches were effective, and she also observed that using the sophisticated sampling techniques *did* not give any clear advantage in the domain considered.

Another study used the cost-sensitive learning, as cited by Chawla [9]. He compares the "meta-cost" approach to each of majority under-sampling and minority over-sampling. He finds that meta-cost improves over either, and that under-sampling is preferable to minority over-sampling. Error-based classifiers are made cost-sensitive. The probability of each class for each example is estimated, and the examples are relabeled optimally with respect to the misclassification costs. The relabeling of the examples expands the decision space as it creates new samples from which the classifier may learn [3].

Estrabooks [6] used the external approach to resolve the class imbalance problem. Resampling was conducted using the following strategies: over-sampling consisted of copying existing training examples at random and adding them to the training set until a full balance was reached. Under-sampling consisted of removing existing examples at random until a full balance was reached. The results suggested the neither over-sampling nor the under-sampling strategy is always the best one to use, and finding a way to combine them could perhaps be useful, especially if the bias resulting from each strategy is of a different nature [4]. Furthermore, the study suggested that resampling to full balance is generally not the optimal resampling rate, at least when the test set is balanced. The optimal resampling rate varies from domain to domain and resampling strategy to resampling strategy. In general, over-sampling changes its effect gradually and in a stable manner with different rates, while under-sampling does so radically and in an unstable manner.

Brennan [3] made a survey on the methods for overcoming the class imbalance problem in fraud detection. RapidMiner, R and Weka were used to study the various methods for overcoming the class imbalance problem. All the sampling methodologies and the cost-sensitive learning method were applied in three different datasets such as car insurance fraud dataset, consumer fraud insurance dataset, and thyroid disease dataset. For all datasets used, the data methods proved to be superior to the algorithmic methods. The data methods surveyed were found to be simple to implement and at least some of them were highly effective. They also proved easier to implement and did not lead to sizable increases in training time or resources needed. SMOTE, ROS, and ROS/RUS proved to be the best performing methods for treating the imbalance in the data. However, ROS may be criticized as a method as it can lead to over-fitting a model as it over trains a model to recognize a small number of minority classes. SMOTE addresses this by creating artificial replicants and thereby creating a less specific feature space for the minority group. However, if the SMOTE algorithm has proved ineffectual at replicating the characteristics of the minority class, it will result in a situation where the minority class sample qualities are too similar to those of the majority class. This result in the problem of "class mixture" and the resulting model will misclassify classes of the minority class as members of the majority class.

This study focuses on the data methodologies (external approach) because of the advantages that it offers and literature argued that this approach is better than the algorithmic methodologies (internal approach). However, the existing data methodologies only assume a random sampling technique that *fully* balances the number of examples in the majority and minority classes.

Furthermore, literature suggests that external approach may be divided into two types of categories. First, there are approaches that focus on studying what the best *data* for inclusion in the training set [10], and second, there are approaches that focus on studying what the best *proportion* of positive and negative examples to include in training set [11]. The work of Estrabooks [6] focused on the second category by creating a framework that deals with the proportion question. This study attempts to answer both questions. The suggested framework provides the best *rate* at which with the combined over-sampling and under-sampling methodology will provide good prediction accuracy even without fully balancing the number of examples in the minority and majority classes. At the same time, it will provide a methodology in determining the data for inclusion in the training set for the over-sampling and the data for exclusion in the training set for the under-sampling methodology.

III. PROPOSED ALGORITHM

This research proposes a combination of oversampling and undersampling methodologies to determine the appropriate number of samples in each class that will give higher class accuracy.

Suppose we have a dataset represented by matrix $\boldsymbol{A}$ with a set of row r and column c. The rows represent examples and the columns are the attributes of the examples with d dimensions. Thus, the matrix element r_{rc} is the value of example with ID r in the attribute with ID c. Consider such a matrix $\boldsymbol{A}$, with n rows and m columns, defined by its set of rows, $R = \{r_1, \ldots, r_n\}$, and its set of columns, $C = \{c_1, \ldots, c_m\}$. Thus, matrix $\boldsymbol{A}$ can be denoted by (R, C). This study provides a framework that can predict variable y, which is directly or indirectly affected by the attributes defined by $C = \{ c_1, \ldots, c_m\}$. If after discretizing the predictor variable y, matrix $\boldsymbol{A}$ can be partitioned into k classes with n_1 elements in class 1… n_k elements in class k, where $n_1, n_2, \ldots, n_k$ are not uniformly distributed. We can define a majority class if $n_k = n_{\text{maximum}}$, otherwise n_k is called minority class. Discussed below is the proposed resampling algorithm to resolve the class imbalance problem.

Algorithm
If matrix ***A***, the original dataset, is divided into *k* classes: $\{C_1, C_2, \ldots, C_k; n_1, n_2, \ldots, n_k\}$
S% = resampling rate
m = number of minority class samples
M = number of majority class samples
n_t = number of examples in class *k* at *i*th iteration.
n_{t-1} = number of examples in class *k* at previous iteration

1 Initialize *S*% = 0.
2 Choose class C_k, where $C_k \subseteq \boldsymbol{A}$
3 If C_k is minority class, then oversample the C_k class.
4 For the array of original minority class samples of size *m*, assign a two-digit number, initialized to 01.
5 Generate random numbers, between 0 to 1, equal to the required oversample size.
6 Use the first two digits to create the synthetic sample of size n_k from the generated random numbers.
7 If C_k is majority class, then undersample the C_k class.
8 Compute $i = \frac{n_{t-1}}{(n_{t-1}-n_t)}$.
9 For the array of original majority class samples of size *M*, eliminate every *i*th instance from the data set.
10
11 Predict *y*
12 Compute over-all accuracy and class accuracy
13 While class accuracy improves
14 Increment *S*%
15 Go to 3
16
17 Stop

Resampling has been conducted using the following strategies: oversampling consisted of copying existing training examples at random and adding them to the training set; while undersampling consisted of removing existing examples at random until the desired number of samples is reached. For example, if datasets initially has 1000 samples in majority class and 50 samples in minority class, that is at *S* = 0%. Using a 10% resampling rate will contain 1000 – [0.10*(1000-50)] = 905 majority class examples and $50 + (0.10 * 1000) = 150$ samples in the minority class. Lines 4 – 6 of the abovementioned algorithm show how oversampling is being done. Oversampling starts by assigning a two-digit number to each of the original minority examples, since the number of minority class is usually less than 100, for most situations. However, the user can redefine it depending on the number of minority examples. By generating random numbers, which is equal to the desired number of examples defined by *S%*, copy the examples at random and add them to the training set. Conversely, lines 8 – 9 show the undersampling procedure. Computing the *i* determines the dataset that will be excluded from the original majority class. The datasets will therefore be resampled by simultaneously reducing the number of majority examples and increasing the number of minority examples. Thus, the algorithm will make the dataset approximately uniformly distributed without abruptly changing the class size, hence, overfitting can be avoided, as well as removing the important data in the majority class can also be prevented. The algorithm terminates if the class accuracy starts to plateau, meaning, the accuracy stops improving. Iterating it further will not improve the accuracy. It therefore implies that the iteration stops at the same time, that is, to minimize the computational time of the algorithm. By applying a combination of undersampling and oversampling, the initial bias of the learner towards the majority class is avoided.

This study assumes that the framework can be applied to any classification/prediction methodologies. Some methodologies that have been investigated are regression, decision tree induction, rule induction, and artificial neural network.

The performance of the prediction/classification is evaluated by a confusion matrix [3], as illustrated in Figure 1.

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Figure 1. Confusion Matrix

The confusion matrix is a useful tool for analyzing how well the classifier can recognize examples of different classes. The columns are the Predicted class and the rows are the Actual class. In the confusion matrix, TN is the number of negative examples correctly classified (True Negatives), FP is the number of negative examples incorrectly classified as positive (False Positives), FN is the number of positive examples incorrectly classified as negative (False Negatives) and TP is the number of positive examples correctly classified (True Positives). Predictive accuracy is the performance measure generally associated prediction or classification algorithms and is defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

In the context of balanced datasets and equal error costs, it is reasonable to use error rate as a performance metric. Error rate is 1 – *Accuracy*.

IV. RESULTS AND DISCUSSION

A. Test Case Data

The proposed algorithm has been tested on four test data sets with various features. The first two data sets are original datasets while the last two data sets are from Knowledge Extraction based on Evolutionary Learning

(KEEL) Data Repository [14]. Table I lists down the four data sets and their corresponding properties, while Figure 2 shows the distribution histogram of the classes of each dataset.

TABLE I – COMPARISON OF TEST CASE DATA

Data Set	Number of Instances	Number of Predictor Variables	Number of Classes	Proportion of Class Imbalance
Crop Yield	1,003	21	3	10:1:1
Credit Risk	348	19	2	3:1
Ecoli	336	7	2	16:1
Yeast	1,489	8	10	90:50:10:1

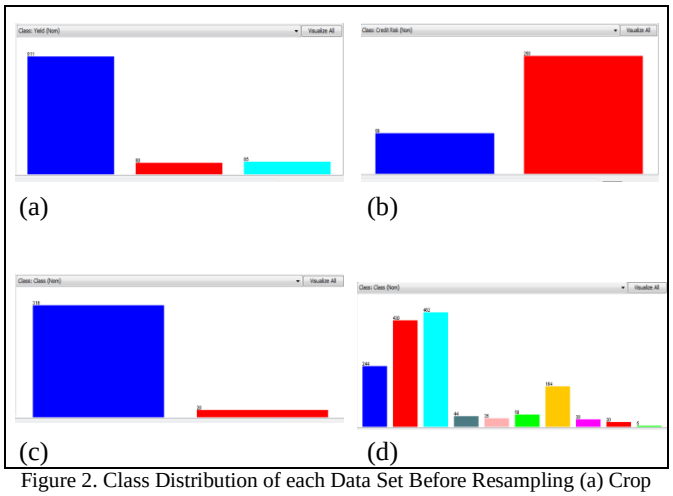

Figure 2. Class Distribution of each Data Set Before Resampling (a) Crop Yield, (b) Credit Risk, (c) Ecoli, and (d) Yeast

B. Prediction

As mentioned in the previous section, various prediction/classification methodologies have been applied, such as regression analysis, decision tree, rule induction technique, and artificial neural network. Data mining algorithms are applied using Waikato Environment for Knowledge Analysis (WEKA) version 3.6.10 [14] software for decision tree, rule induction, and neural network. It includes a wide variety of learning algorithms and preprocessing tools. Minitab 16 has been used to implement the regression analysis.

The performance of the applied data mining algorithms is estimated by the 10-fold cross validation. Data are randomly partitioned into 10 blocks, one block is held out for the test purpose and the model is built on the remaining nine blocks. This method is then repeated for other blocks. After repeating the calibration and validation processes with ten different combinations, the results (prediction accuracy and MAE) obtained with these ten different validation datasets are summarized by calculating the mean value and 95% confidence interval. It should be noted, however, that the calibration and validation dataset are independents throughout the procedure.

C. Resampling Rate

Resampling rate is a parameter that will be selected by the user (user-specified). Choosing the appropriate resampling rate and its increment size define the computational time in achieving the balanced dataset. Selecting a low increment size might mean a slow convergence while incrementing it at high values might show a very fast convergence, which in effect might not give the best resampling size. Figure 2 shows the summary of the analysis.

It is gleaned from Fig. 3 that incrementing it at 1% shows a very slow convergence, which is after 29 runs. However, having a step increment of 20% shows a very fast convergence, which only took 3 runs. This is not good because it will not give us the appropriate resampling size.

A resampling size is said to be appropriate if it is not exceedingly under-sampled nor oversampled. Generally, increment size of the resampling rate depends on how large or small is the gap of the majority class and the minority class. The smaller the gap between the majority and minority class means small increment size for resampling rate can be used. However, if the disparity is high, higher size increment is suggested.

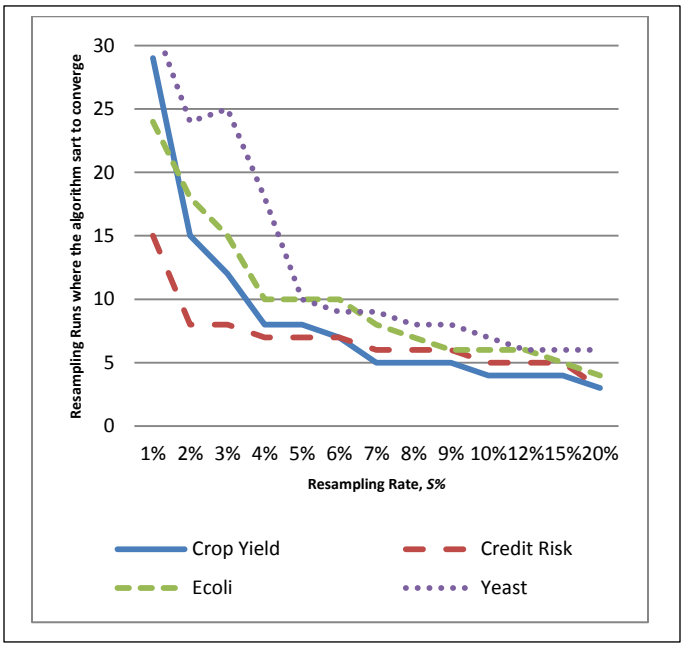

Figure 3. Convergence at Different Resampling Rate

D. Resampling Size

After identifying the increment size of the resampling rate, we can now determine the appropriate resampling size. As mentioned above, a good resampling size will give the user an approximately balanced class size that is not overly undersampled nor oversampled, thus, overfitting can be precluded. Table II shows the different results of the resampling runs.

 ISBN: 978-1-61208-358-2

TABLE II – RESAMPLING RUNS

Resampling Size	Data Sets			
	Crop Yield	*Credit Risk*	*Ecoli*	*Yeast*
Class A	568	232	269	370
Class B	328	115	67	344
Class C	323	-	-	288
Class D	-	-	-	224
Class E	-	-	-	132
Class F	-	-	-	128
Class G	-	-	-	116
Class H	-	-	-	120
Class I	-	-	-	108
Class J	-	-	-	96
Ressampling Rate	5%	2%	5%	5%
Number of Resampling Runs to Converge	8	4	7	6

It is shown in Table II that the resampling rate of 5% implies that each run increases the undersampling or oversampling rate by 5%. Most of the data sets in this research use a resampling rate of 5%. The number of resampling runs shows the number of runs by which the accuracy graph converges. After that run, the next runs show very little improvement in each class accuracy, that is the graph starts to plateau. Hence, it is considered as the appropriate resampling size for each data set. Fig. 4 shows the improved class distribution after resampling.

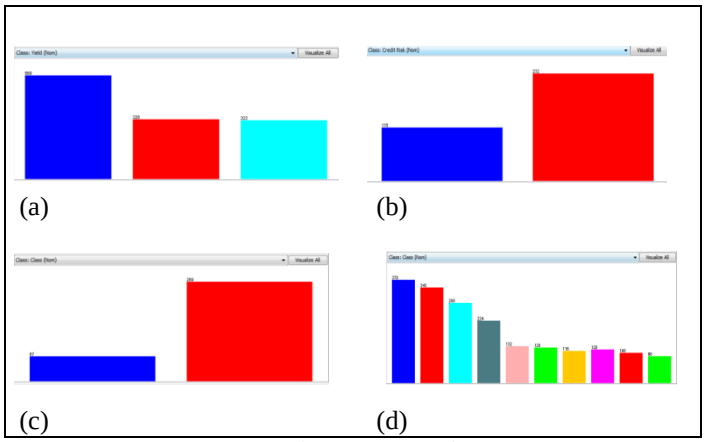

Figure 4. Class Distribution of each Data Set After Resampling (a) Crop Yield, (b) Credit Risk, (c) Ecoli, and (d) Yeast

The class accuracy generally improves for each data set after resampling, using any of the prediction/classification methodologies. Figure 5a to Figure 5d show the comparison of the overall accuracy and class accuracy before and after the resampling using the regression analysis, decision tree induction, rule induction, and artificial neural network.

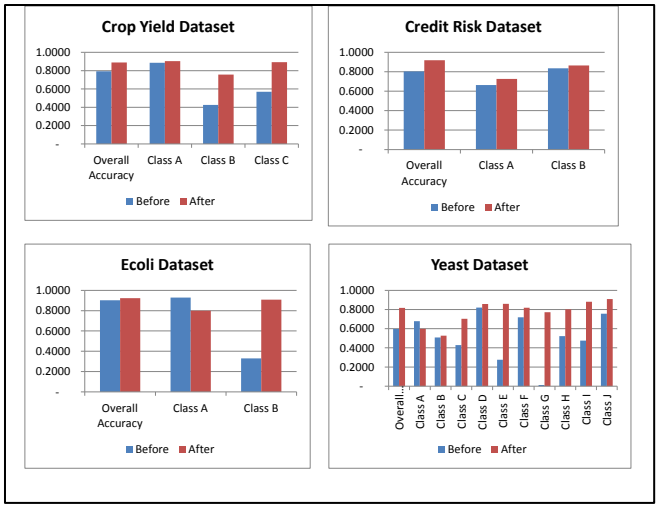

Figure 5a. Comparison of Over-all and Class Accuracy Before and After Resampling Using Regression Analysis

Regression analysis is implemented using Minitab 16 software. Since the predictor variable is discretized, general regression is used. A regression model is developed initially using the original dataset (before resampling) for each dataset. For the Crop Yield dataset, although the overall accuracy is almost 80%, investigating the class accuracy shows that Class B is the problematic class since only 42% was correctly classified examples. Class C is also problematic since it only 57% was correctly classified. The low prediction accuracy for these classes is caused by the fact that Classes B and C are both minority class. Credit Risk dataset has little problem on class A accuracy since the accuracy is almost 66% as compared to class B accuracy which is 84%. Notice that Class A is the minority class in this case, hence low accuracy is explained by this fact. For the Ecoli dataset, Class B has the lowest accuracy, which is only 33% as compared to Class A accuracy of 93%. It is because Class B is the minority class. For Yeast dataset, five classes have class accuracy of below 50%, which are Classes C, E, G, H, and I, Class G being the most problematic with class accuracy of only 1.19%.

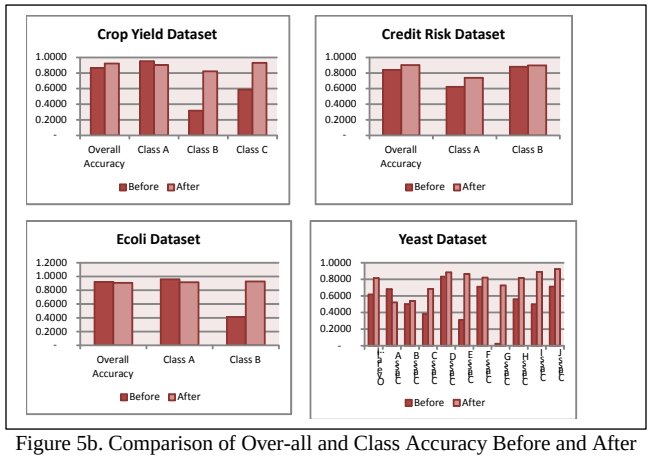

Figure 5b. Comparison of Over-all and Class Accuracy Before and After Resampling Using Decision Tree Induction Technique

The datasets are also analyzed using decision tree technique. The situation before resampling is almost the same as that of the previous technique, except that the prediction accuracy is relatively higher in using this technique. However, the minority classes are still the classes with low class accuracy. To resolve the imbalance, the proposed framework is then applied, this time using decision tree technique. Decision tree technique is applied using C4.5 algorithm, which is a built-in algorithm in WEKA. As can be seen in Figure 5b, class accuracy is again improved significantly.

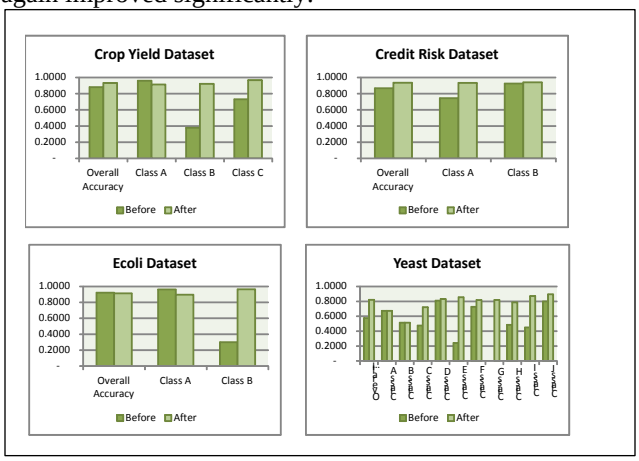

Figure 5c. Comparison of Over-all and Class Accuracy Before and After Resampling Using Rule Induction Technique

Rule Induction technique is also employed to the 4 datasets to determine if the framework can also be applied. JRip algorithm is used under this technique, which is also implemented using WEKA. Similarly, the situation is also the same with respect to class accuracy. Minority classes have the low accuracy. Resampling the class size again improves the class accuracy as shown in Figure 5c.

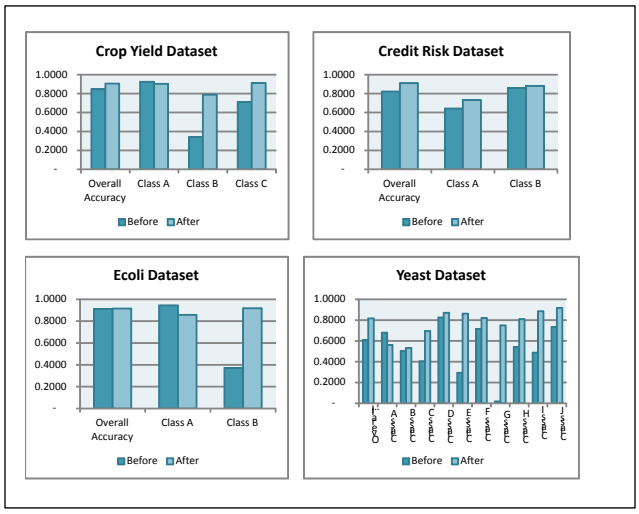

Figure 5d. Comparison of Over-all and Class Accuracy Before and After Resampling Using Artificial Neural Network

Another commonly used technique in classification is the Artificial Neural Network (ANN) [2]. Backpropagation algorithm is used under this technique. Same scenario can be observed even this technique is used, the minority classes are the classes that have the low accuracy. The framework is again applied, this time using the ANN. As can be seen in Figure 5d, the class accuracy also improved by resampling the class size.

E. Validation

The performance of the proposed framework is compared to the existing sampling methodologies used in resolving the class imbalance problem. As discussed in the literature, ROS, RUS, ROS/RUS, and the SMOTE are the most commonly used under resampling techniques. ROS has been implemented by randomly creating a number of replicants of the minority class without replacement until it is equal to the number of samples of the majority class. RUS has been implemented by simply choosing a random sample of the majority class which matches the number of minority class examples. The ROS/RUS involves the combination of the two prior methods until the number of minority samples equals the number of majority samples in the ratio 50/50. SMOTE has been implemented through WEKA, where it consists of reiterating the framework until the prediction accuracy stops improving.

The proposed framework is also compared to one of the most commonly used algorithmic technique, which is the cost-sensitive learning (CSL). Cost-sensitive learning has been implemented using Cost-Sensitive Rough Sets (COSER) in WEKA.

The methodologies are compared based on the oversampling rate, undersampling rate, elapsed time, and prediction accuracy. Oversampling rate is the percentage by which the minority class is oversampled to be equal to that of the majority class, while the undersampling rate is the percentage by which the majority class is undersampled to be equal to the minority class. Certainly, we want to minimize both the undersampling rate and the oversampling rate in the shortest possible time, so that both the overfitting and loss of data issues are avoided. Elapsed time refers to the total time by which the methodology used is able to reach the predicted accuracy. Times are measured in minutes. The prediction accuracy is the performance measure used in comparing the methodologies. The results are summarized in Table III.

Examination of the table shows that ROS has the best prediction accuracy. Similarly, ROS has the shortest elapsed time among all the dataset tested. The success of the ROS method can be explained by the fact that all the members of the majority class are being utilized and the random replication of the minority class until it is balanced with the majority class. This redistribution of the class size provides the prediction/classification learner sufficient samples to be able to train a model to recognize the minority class, and not treating them as noise. However, the near perfect performance of the ROS method must be tempered by the fact that oversampling can result in overfitting [3].

 ISBN: 978-1-61208-358-2

TABLE IV – COMPARISON WITH THE EXISTING METHODOLOGIES

		Ratio of Major class and Minor class(Cost Ratio)	Largest Down sampling	Largest Upsampling Rate	Elapsed Time (minutes)	Predicted Accuracy
Crop Yield	ROS	1.00	-	1013.75%	2.54	0.9716
	RUS	1.00	90.14%	-	4.23	0.7561
	Hybrid	1.00	49.94%	199.75%	4.89	0.9262
	SMOTE	1.19	-	750.00%	16.71	0.8315
	CSL	0.40	-	-	40.25	0.8815
	Proposed	1.76	29.96%	247.26%	24.35	0.9310
Credit Risk	ROS	1.00	-	289.88%	0.72	0.9893
	RUS	1.00	65.50%	-	1.46	0.6157
	Hybrid	1.00	50.00%	144.94%	1.18	0.9012
	SMOTE	2.35	-	23.60%	0.53	0.7647
	CSL	0.125	-	-	26.87	0.6667
	Proposed	2.02	10.08%	224.35%	2.74	0.9345
Ecoli	ROS	1.00	-	1580.00%	1.63	0.9914
	RUS	1.00	93.67%	-	1.84	0.2354
	Hybrid	1.00	50.00%	790.00%	2.26	0.8831
	SMOTE	7.90	-	100.00%	0.90	0.8324
	CSL	0.05	-	-	18.29	0.6502
	Proposed	4.01	14.87%	335.00%	3.82	0.9137
Yeast	ROS	1.00	-	9240.00%	20.78	0.9865
	RUS	1.00	89.18%	-	31.52	0.5327
	Hybrid	1.00	50.00%	462.00%	25.66	0.8329
	SMOTE	4.96	-	1820.00%	42.69	0.5634
	CSL	0.005	-	-	153.74	0.3160
	Proposed	3.85	19.91%	192.00%	45.21	0.8208

RUS delivered the poorest prediction accuracy among the dataset tested. The effect of reducing the number of majority samples results in the learner not having enough majority samples to train an effective model. This is due to the learner not being exposed to enough of the majority samples in the training set and ignoring most of the population of the majority class examples, as they are simply discarded, while training the model [3].

The ROS/RUS tends to have good performance in terms of prediction accuracy and elapsed time. It is because, to a less extent, only some of the majority examples are being removed, causing the synthesis of a less than perfect model. This technique is also better than ROS since it will avoid the chance of overfitting since the minority class will not be overly sampled.

SMOTE is also ineffectual at creating artificial replicants of the minority class. This is due to the artificial replicants it created based on the minority class are too similar to the majority class [2].

The cost-sensitive learning has not proved to be capable of improving the prediction accuracy across all the datasets used. They tend to be poor at differentiating the majority from the minority class and misclassifying the majority class as minority class members (false positives). This led to poor decision of the model. It has also been difficult to implement than the resampling methods. As for each learner, the cost ratio of the majority to the minority class misclassification cost, had to be derived empirically. It explains the long elapsed time for cost-sensitive learning methodology. Furthermore, time complexity tends to increase as the number of classes and the proportion of class imbalance increase.

The proposed framework is almost comparable to that of the hybrid technique. It can predict with a higher degree of accuracy and at the same time avoids the possibility of overfitting, as in the case of ROS. It is a better technique since it aims to determine the best class size that will give a higher prediction accuracy even without overly oversampling nor overly undersampling the classes. However, one limitation of the framework is that it takes longer time in realizing good prediction accuracy. It is because the algorithm takes more iterations than the simple ROS, RUS, and ROS/RUS since it does not require the classes to be fully balanced. Hence, it tries to determine the appropriate size for each class so that overfitting can be avoided. Furthermore, it tries to determine the appropriate sample to be included in the resampled class. In general, the proposed prediction framework for imbalanced dataset shows satisfactory performance as compared to the existing methodologies.

V. CONCLUSION AND AREAS FOR FURTHER STUDY

Literature proved that sampling methodologies are better than the algorithmic methodologies in resolving the class imbalance problem. Among the sampling methodologies being used are Random Oversampling (ROS), Random Undersampling (RUS) Hybrid technique (ROS/RUS), and the SMOTE. Nevertheless, the existing approaches for sampling have disadvantages. The questions on how much should we oversample and undersample, and which data must be included/excluded in the dataset still exist. Hence, the proposed prediction framework is developed to provide answers to these issues. This study generally aims to develop a prediction model framework that can pre-process data and resolve the imbalance problem by utilizing a proposed iterative oversampling and underampling methodology for *n*-dimensional data sets.

The framework consists of pre-processing component, which consists of data discretization and data resampling. The resampling algorithm is an iterative one which attempts to determine the best data to include/exclude in the training set and to determine the appropriate resampling rate. The appropriate resampling rate is a user-defined parameter which can be determined by computing the prediction accuracy for each resampling size. It is said appropriate if the increase in prediction accuracy started to stabilize at that point. Based on the analysed data, resampling rate generally depends on the gap of the majority and minority class. The smaller the gap between the majority and minority class means small increment size for resampling rate can be used. However, if the disparity is high, higher size increment is suggested.

Four datasets have been used, which are the crop yield dataset, credit risk dataset, ecoli dataset, and yeast dataset,

 ISBN: 978-1-61208-358-2

to test the performance of the framework. The prediction component of the framework attempts to investigate if the framework can be applied in any prediction/classification methodologies. Regression analysis, decision tree (DT), rule induction, and artificial neural networks (ANN) have been used as prediction/classification methodologies. The study reveals that the framework can be applied to any of the methodologies mentioned, though it works well in rule induction technique because it provides the highest overall and class accuracy.

The framework is also compared to the existing approaches in resolving class imbalance. The analysis reveals that the proposed framework is comparable to the hybrid technique, but the main difference is that the framework minimizes the oversampling and undersampling rate, but still gives good prediction accuracy.

For the future, there are different ways in which this study could be expanded. First, the procedure in determining the appropriate resampling size can be established. A mathematical model that can give the optimum resampling size can be done. Second, other performance measure, aside from prediction accuracy, can be investigated. Finally, prediction framework based algorithmic methodologies (internal approach) can also be studied.

VI. ACKNOWLEDGMENT

This research was funded by the Engineering Research and Development for Technology (ERDT) grant, under the Department of Science and Technology.

VII. REFERENCES

[1] Ling, C., and Li, C."Data Mining for Direct Marketing Problems and Solutions" The Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98) New York, NY. AAAI Press, 1998

[2] Gorni, A. A., "The Application of Neural Networks in the Modeling of Plate Rolling Processes" 2008.

[3] Brennan, P. "A Comprehensive Survey of Methods for Overcoming the Class Imbalance Problem in Fraud Detection" 2012.

[4] Kubat, M., and Matwin, S. "Addressing the Curse of Imbalanced Training Sets: One Sided Selection" In Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179–186 Nashville, Tennesse. Morgan Kaufmann, 1997.

[5] Hu, X. "Using Rough Sets Theory and Database Operations to Construct a Good Ensemble of Classifiers for Data Mining Applications," in Proc. IEEE Int. Conf. Data Mining, 2001, pp. 233–240.

[6] Estabrooks, A., Jo, T., and Japkowicz, N., "A Multiple Resampling Method for Learning from Imbalanced Data Sets," Comput. Intell., vol. 20, no. 1, 2004, pp. 18–36.

[7] Patterson, L., "The Nine Most Common Data Mining Techniques Used in Predictive Analytics. 2010

[8] Japkowicz, N., and Stephen, S. "The Class Imbalance Problem: A Systematic Study." Intelligent Data Analysis 6:429–450, 2002.

[9] Chawla N. V., Bowyer K. W., Hall, L. O., and Kegelmeyer, W. P. "SMOTE: Synthetic Minority Over-sampling Technique" Journal of Artificial Intelligence Research 16:321– 357, 2002.

[10] Kaspar, T.C., et al., "Relationship Between Six Years of Corn Yields and Terrain Attributes," *Precision Agriculture*, 4(1), 2003, 87-101.

[11] Kumar, Ch. N., et.al., "An Updated Literature Review on the Problem of Class Imbalanced Learning in Clustering," in IJETR, Volume 2, Issue 2, February 2014.

[12] Barandela, R., Sánchez, J.S., García, V., and Rangel, E. "Strategies for Learning in Class Imbalance Problems" Pattern Recognition 36:849–851, 2003.

[13] Fernandez, A., Garcıa, S., del Jesus, M. J., and Herrera, F., "A Study of the Behaviour of Linguistic Fuzzy-Rule-Based Classification Systems in the Framework of Imbalanced Data-sets," Fuzzy Sets Syst., vol. 159, no. 18, 2008, pp. 2378–2398.

[14] KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3, 2011, 255-287.

[15] Accessed August 15, 2013. http://weka.sourceforge.net/packageMetaData/