International Journal of Hybrid Intelligent Systems 12 (2015) 219–227
DOI 10.3233/HIS-160217
IOS Press

Class imbalance problem using a hybrid ensemble approach

Shaza M. Abd Elrahman[a] and Ajith Abraham[b,*]
[a]*Faculty of Computer Science & Information Technology, Sudan University of Science Technology, Khartoum, Sudan*
[b]*Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, WA, USA*

Abstract. This paper proposes a comparative study that investigates the effects of using resampling (undersampling and oversampling) methods with homogenous ensemble methods Bagging and AdaBoost in imbalanced data sets. We presented a hybrid ensemble approach that combined multi resampling by integrating both undersampling and oversampling to get benefits and reduces drawbacks caused by each of them. The proposed approach has improved the performance even those most sensitive to imbalanced class data sets.

Keywords: Data mining, machine learning, class imbalance, ensemble learning

1. Introduction

Imbalanced class classification problem occur when the instances of class outnumber the instances of other classes. Imbalance data sets degrades the performance of data mining and machine learning techniques as the overall accuracy and decision making be biased to the majority class which lead to misclassifying the minority class samples or furthermore treated them as noise. However, in many applications the classes have lower numbers of instances are the more interesting and important ones. Many real world applications suffer from these phenomena such as medical diagnosis, insurance fraud(credit card, phone calls, insurance), network intrusion detection, pollution detection, fault monitoring, biomedical, bioinformatics and remote sensing (land mine, under water mine). As an example insurance fraud considered one of highly imbalance class problem. Although there is massive data however, most of them are legitimate and a little are fraudulent. Moreover, the cost of misclassifying the minority class is very high in comparison with the cost of misclassifying the majority class. Consider fraud versus non-fraud, the error of misclassification of positive class (fraud) as negative (non-fraud) is very big and may cause huge losses. The researchers for solving the imbalance problem have proposed various approaches. However, there is no general approach proper for all imbalance data sets and there is no unification framework. In this paper we investigate the two-class imbalance class's problem through using a hybrid ensemble approach.

Several methods proposed for solution the imbalance class problems include re-sampling and feature selection at the data level and other ones at the algorithm level such as cost sensitive, one class (recognition based) learning and ensemble methods. Sampling methods is a preprocessing of data, which handle the imbalance problem by constructing balanced training data set and adjusting the prior distribution for minority and majority class. Sampling methods include under sampling and over sampling methods. Under sampling balance the data by removing samples from majority class. Oversampling balanced the data by create copies of the existing samples or adding more samples

*Corresponding author: Ajith Abraham, Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, WA, USA. E-mail: ajith.abraham@ieee.org.

to the minority class. Oversampling can be done using a non-heuristic approach by randomly duplicates samples of minority class or adding new samples by using a heuristic approach. However, random over sampling may cause over fitting and may introduce additional computational tasks. The cost learning techniques take the misclassification cost in its account by assigning higher cost of misclassification to the positive class (minority class) i.e. C(+,-)> C(-,+) and generate the model with lowest cost. However, the misclassification errors costs are often unknown and furthermore, cost sensitive learning may lead to over fitting. In recognition based method or (one-class learning) the classifier learned on the just target class samples. This approach improves the performance of the classifier on unseen data by recognized only those belong to that class. In this paper we review those ensemble approaches proposed as solutions for class imbalance problem. Ensemble is a combination of multiple classifiers so as to improve the generalization ability and increase the prediction accuracy. The most popular combining techniques are boosting and bagging. In boosting, each classifier is dependent on the previous one, and focuses on the previous one's errors. Samples that are misclassified in previous classifiers are chosen more often or weighted more heavily. Whereas, in bagging, each model in the ensemble votes with an equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly drawn subset of the training set [1].

The major objective of this paper is to investigate the imbalanced class problem through using an ensemble approach. Additional objectives are listed as follows:

- To investigate classifiers those are less sensitive to the class imbalance problem.
- To evaluate the classifiers performance under several circumstances.
- To improve the performance through using an ensemble approach.

2. Related works

This section explains the related works proposed as solutions for class imbalance problem based on ensemble approach. Kang and Cho [2] proposed an ensemble of under sampled SVM (EUS SVMs). They integrated the good generalization ability of SVM by boosting ensemble scheme. Their proposed method overcame the drawback of under sampling method and reduced the time complexity of oversampling method. Zhang and Wang [3] presented an ensemble model that combining cost sensitive SVM and query by committee (QBC) with AdaBoost learning. The majority class divided into several subsets regarding to imbalance ratio. Then QBC which is an active learning method is used to generate nominee training samples the effective ones be chosen. AdaBoost is used to train the sub classifiers. Khoshgoftaar et al. [4] studied empirically the use of different data sampling with Boosting including random undersampling, random oversampling, SMOTE, Borderline SMOTE and wilson's editing. The best performance usually obtained by undersampling. SMOTE and borderline SMOTE given better results than the random oversampling and Wilson's editing. They concluded that Boosting improve performance over sampling methods.

A hybrid kernel ensemble that integrated two types of kernel machine: one class SVM and binary SVM (BSVM) proposed by Chan and Fang in [5]. Also, sampling methods used with (BSVM). Khoshgoftaar et al. [6] and Govindaraj and Lavanya [7] proposed a hybrid approach using random undersampling with AdaBoost (RUSBoost). They obtained the desired distribution by randomly remove samples from majority class. RUSBoost is simpler and faster technique comparing to SMOTEBoost and other technique. Both oversampling and undersampling used with AdaBoost [7]. Yuan and Ma [8] improved the performance by using SMOTE with AdaBoost and an objective function using optimization technique such as genetic algorithm.

Ren et al. [9] illustrated an ensemble model that integrates sampling with AdaBoost using Naïve Bayes (NB) and decision tree C4.5 as a base classifier. Random sampling used with NB to denote the data distribution and Undersampling used with C4.5/C4.5+ AdaBoost.

To tackle the deficiency of undersampling Liu et al. [10] proposed two ensemble models called Easy ensemble and balanced cascade ensemble. In easy ensemble the combined classifiers are trained on different subsets separately. In balanced cascade ensemble the combined classifiers trained sequentially using a guide in the sampling process for each classifier by removing samples those are classified correctly. Recently, Tianyu [11] used easy ensemble based feature selection. To improve the performance, PSO is applied to get the optimal feature subsets. Khoshgoftaar et al. [12] proposed a filter-based feature ranking techniques. They applied an iterative feature selection strategy and combined it with sampling and boost-

ing techniques. This method repeatedly employed data sampling followed by feature selection. The ranked features sets is taken from each iteration and applied to boosting learners. An investigation on the performance of random sampling and advanced under sampling (CUBE) and two modeling techniques (gradient boosting and weighted random forests) was introduced by Burez and Poel [13]. They concluded that under sampling improved the prediction accuracy comparably with sophisticated under sampling which had no any effect on the performance. Also, they found that Boosting is a robust classifier but not surpassed the other techniques and weighted random forest performed better than random forest.

Gue and Viktor [14] proposed an ensemble based learning approach (DataBoost-IM) that combined boosting with data generation. The hard examples were identified then they were used to generate synthetic examples for both classes to be focus by the next classifier component in the boosting procedure. However, synthetic examples prevented boosting from over fitting on hard examples. Another ensemble in a hierarchical frame was proposed by Zhang and Luo [15]. They proposed a parallel classification method to improve classifying speed; two classifiers (simple one and complicated one) were trained serially but worked in parallel. The results showed that their proposed approach effectively improved performance and speed.

An approach based on repeated sub-sampling was proposed by Khalilia et al. [16]. Authors compared the performance of SVM, bagging, boosting and Random Forest (RF). They emphasized the effectiveness of repeated sub-sampling in dealing with highly imbalance data sets. However, RF outperformed other methods plus its ability to estimate the importance of each variable in classification process.

Recently, a hybrid ensemble model that integrated sampling, clustering and bagging proposed in [17]. Firstly, the borderline majority samples are removed using Tome links undersampling technique. Then the remaining majority class divided into a number of subsets (clusters). These subsets are combined with the minority class using bagging learning technique. For diversity they used random forests and decision tree as base classifiers for the ensemble. In [18] a hybrid ensemble model is proposed that combined feature selection with cost sensitive learning. Random feature subspaces are used for the diversity of ensemble. Cost matrix is used to construct the base classifiers. To promote the performance, an evolutionary algorithm is used for classifier selection and assignment of committee member weights.

3. Methodology and experimental layout

3.1. Experiment design

In our experiments we used seven classifiers with three different scenarios and phases. In **Phase One**, we have tested the performance of seven selected classifiers (Naïve Bayes (NB), Back Propagation Neural Network (BP), Support Vector Machine (SVM), Radial Basis Function Neural Network (RBF), C4.5, Random Tree (RT), and Random Forest (RF)) and compare their results when applied using homogenous ensemble approaches such as Bagging and AdaBoost learning methods. The main objective of this phase is to compare the performance of different classifiers to reveal those sensitive to class imbalanced class problem. Also from this phase we display the impact of using ensemble approaches and their improvements on the performance of classifiers when dealing with imbalanced classes data using original data (without applying any resampling method).

In **Phase Two**, we have balanced data by resampling it using random undersampling by selecting random subsets from the majority classes that equal the size of minority classes. Then we tested the performance of seven classifiers using the new resampled data and compare their results when applying ensemble approaches such as Bagging and AdaBoost. The objective of this phase is to examine the effects of using undersampling on the performance of different classifiers when using them solitary or within homogenous ensemble approaches when dealing with imbalanced classes.

In **Phase Three**, we resampled data using oversampling by increasing the minority class samples using SMOTE (Synthetic Minority Oversampling Technique) proposed by Chawla et al. [19] by generating synthetic examples rather than replacement with replication for the existing minority class samples. SMOTE works by selecting some or all the nearest neighbors for each minority sample and then take the difference between the feature vector (minority sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1 and add it to the feature vector under Consideration to produce the synthetic samples and add them to the minority class. The new resampled data is used for training classifiers. We compare their result when applied them using ensemble approaches such as Bagging and AdaBoost. The objective of this phase is to examine the effects of using oversampling on the perfor-

Table 1

Data set	#of instances	# of Attributes	# of instances in majority class	# of instances in minority class	Imbalance ratio	Ref
Insurance fraud	15420	32	14497 (94%)	923 (6%)	15.7	[20]
Pima	768	9	500 (65%)	286 (35%)	1.9	[21]
Hepatit	155	20	123 (79%)	32 (21%)	3.8	[21]
German	1000	21	700 (70%)	300 (30%)	2.3	[21]
Haberman	306	4	225 (74%)	81 (26%)	2.8	[21]

Table 2
Performance of classifiers on different datasets in term of accuracy

Classifier	Fraud detection	German	Pima	Hepatit	Haberman
NB	0.93	0.75	0.76	0.83	0.75
SVM	0.94	0.70	0.65	0.79	0.74
BP	0.94	0.73	0.75	0.82	0.73
RBF	0.94	0.74	0.75	0.83	0.74
C4.5	0.94	0.71	0.74	0.81	0.72
RF	0.97	0.74	0.73	0.84	0.69
RT	0.96	0.66	0.71	0.77	0.64

mance of different classifiers when using them solitary or within homogenous ensemble using ensemble approaches when dealing with imbalanced classes.

3.2. Evaluation measures

In our experiments, we used the most evaluation metrics related to imbalance classes which TP rate (sensitivity) (1), TN rate (specificity) (2), precision (3), f-measure (4) and AUC (5) as evaluation performance measures. TP rate and TN rate are used to monitor the classification performance on each individual class: positive (minority) class and negative (majority) class respectively. While precision is used in problems which interested on highly performance on only one class, F-measure is used when the performance on both classes – majority and minority classes-needed to be high. These metrics are derived from the following confusion matrix:

Actual class	Predicted Class +ve	Predicted Class -ve
+ve	True Positive (TP)	False Negative (FN)
-ve	False Positive (FP)	True Negative (TN)

$$\text{Sensitivity (true positive rate)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (1)$$

$$\text{Specifity (true negative rate)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (2)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

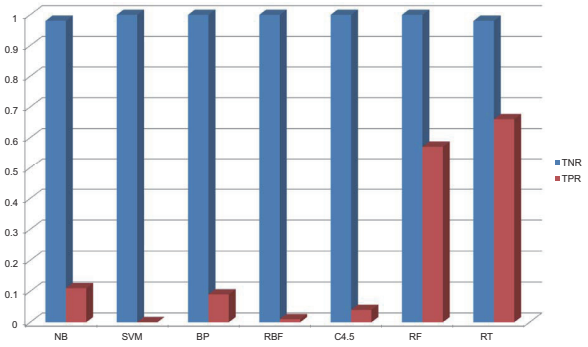

Fig. 1. The detection rates for positive and negative classes in Insurance fraud data set. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/HIS-150217)

$$\text{F} = \frac{2.\text{precision.Sensitivity}}{\text{precision} + \text{Sensitivity}} \quad (4)$$

AUC (5) is the area under the ROC (Receiver Operating Characteristic) curve, which is computed by:

$$\text{AUC} = \frac{\text{TPrate} + \text{TNrate}}{2} \quad (5)$$

3.3. Experiments analysis and discussion

In our experiments we used five data sets with different imbalance ratios as summarized in Table 1. As evident from Table 2 and Fig. 1, all classifiers have overall accuracy up to (92%). But if we compare the performance using detection rate for each class: TP rate and TN rate as depicted in Table 3, we find out that the detection rates for majority class (True negative rates) are always up to (98%) regardless of the classifier used. Contrary to these results, the highest detection rates for the minority class (TP rates) are 65%, 56%, which are obtained by random tree and random forests respectively but all other classifiers have given TP rates less than 1%. The same thing occurs with other four datasets, which are depicted in Tables 4–7. The obtained results for all datasets emphasize that the overall accuracy are biased towards majority class. Obviously, we can also deduce that all the used classifiers are very sensitive for imbalanced classes but the most influenced one is SVM.

Table 3
Performance of different classifiers on Insurance Fraud data sets using different performance measures

Classifier		Original data distribution					Balancing data using undersampling					Balancing data using oversampling				
		TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC
NB		0.98	0.11	0.27	0.16	0.81	0.59	0.94	0.13	0.23	0.81	0.72	0.67	0.14	0.23	0.77
SVM		1	0	0	0	0.50	0.61	0.90	0.13	0.23	0.76	0.86	0.51	0.19	0.28	0.68
BP		1	0.09	0.86	0.17	0.76	0.61	0.90	0.13	0.23	0.83	0.95	0.40	0.33	0.36	0.81
RBF		1	0.01	0.00	0.01	0.81	0.63	0.85	0.13	0.23	0.79	0.79	0.57	0.16	0.25	0.72
C4.5		1	0.04	0.83	0.07	0.65	0.59	0.96	0.13	0.24	0.79	0.95	0.42	0.37	0.40	0.84
RF		1	0.57	0.97	0.72	0.93	0.66	0.91	0.15	0.26	0.88	0.99	0.66	0.79	0.72	0.93
RT		0.98	0.66	0.70	0.68	0.84	0.63	0.90	0.14	0.24	0.81	0.96	0.73	0.56	0.63	0.86
Using bagging	NB	0.98	0.09	0.25	0.13	0.81	0.57	0.93	0.13	0.23	0.80	0.72	0.67	0.14	0.23	0.77
	SVM	1	0.01	1	0.01	0.52	0.60	0.90	0.13	0.23	0.80	0.86	0.51	0.19	0.29	0.74
	BP	1	0.16	0.97	0.28	0.90	0.79	0.91	0.16	0.28	0.87	0.95	0.46	0.40	0.43	0.87
	RBF	0.856	0.51	0.19	0.28	0.74	0.66	0.81	0.14	0.23	0.80	0.79	0.58	0.16	0.25	0.73
	C4.5	1	0.04	0.83	0.07	0.78	0.64	0.93	0.15	0.25	0.86	0.99	0.66	0.79	0.72	0.93
	RF	1	0.34	1	0.51	0.93	0.65	0.94	0.15	0.26	0.89	0.99	0.62	0.78	0.69	0.94
	RT	1	0.57	0.97	0.72	0.93	0.66	0.91	0.15	0.26	0.88	0.99	0.66	0.79	0.72	0.93
Using adaBoost	NB	0.98	0.11	0.25	0.15	0.79	0.65	0.84	0.14	0.24	0.80	0.72	0.67	0.14	0.23	0.70
	SVM	0.99	0.39	0.67	0.49	0.88	0.67	0.82	0.14	0.24	0.80	0.93	0.48	0.30	0.37	0.86
	BP	1	0.09	0.86	0.17	0.55	0.65	0.90	0.15	0.25	0.82	0.95	0.40	0.33	0.36	0.67
	RBF	0.99	0.06	0.29	0.09	0.81	0.70	0.77	0.14	0.24	0.80	0.83	0.52	0.17	0.26	0.77
	C4.5	0.99	0.67	0.88	0.76	0.90	0.70	0.91	0.17	0.29	0.86	0.99	0.67	0.74	0.70	0.91
	RF	1	0.64	0.92	0.75	0.89	0.69	0.90	0.16	0.27	0.88	0.99	0.66	0.79	0.72	0.88
	RT	0.98	0.67	0.74	0.71	0.84	0.6	0.90	0.13	0.23	0.78	0.98	0.68	0.67	0.68[1]	0.92

[1]TNR: True Negative rates (detection rate of negative (majority) class), TPR: True Positive rates (detection rate of positive (minority) class), Prec: Precision, F-M: F-measure and ROC: Roc Area.

Table 4
Performance of different classifiers on German data sets using different performance measures

Classifier		Original data distribution					Balancing data using undersampling					Balancing data using oversampling				
		TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC
NB		0.86	0.50	0.8	0.83	0.79	0.88	0.71	0.86	0.77	0.87	0.79	0.80	0.79	0.79	0.86
SVM		0	1	0	0	0.50	0.16	0.87	0.51	0.64	0.51	0.91	0.35	0.80	0.49	0.63
BP		0.82	0.52	0.55	0.53	0.74	0.79	0.74	0.78	0.76	0.85	0.79	0.79	0.79	0.79	0.86
RBF		0.87	0.45	0.60	0.51	0.76	0.81	0.75	0.80	0.77	0.86	0.79	0.76	0.78	0.77	0.84
C4.5		0.84	0.39	0.52	0.44	0.64	0.81	0.73	0.80	0.76	0.79	0.78	0.77	0.78	0.77	0.78
RF		0.88	0.41	0.59	0.49	0.73	0.78	0.76	0.78	0.77	0.87	0.82	0.80	0.81	0.81	0.89
RT		0.75	0.45	0.43	0.44	0.60	0.71	0.71	0.71	0.71	0.72	0.71	0.77	0.73	0.75	0.75
Using bagging	NB	0.87	0.51	0.13	0.56	0.79	0.88	0.72	0.85	0.78	0.87	0.79	0.79	0.79	0.79	0.86
	SVM	1	0	0	0	0.50	0.62	0.38	0.50	0.43	0.50	0.92	0.33	0.80	0.47	0.68
	BP	0.87	0.53	0.64	0.58	0.77	0.82	0.76	0.81	0.78	0.88	0.81	0.81	0.81	0.81	0.89
	RBF	0.89	0.43	0.63	0.51	0.78	0.82	0.77	0.81	0.79	0.88	0.78	0.79	0.78	0.79	0.86
	C4.5	0.87	0.43	0.58	0.49	0.75	0.87	0.71	0.85	0.77	0.88	0.80	0.79	0.80	0.80	0.88
	RF	0.92	0.36	0.67	0.47	0.79	0.84	0.76	0.87	0.79	0.89	0.83	0.86	0.83	0.83	0.91
	RT	0.88	0.41	0.59	0.49	0.73	0.78	0.76	0.78	0.77	0.87	0.82	0.80	0.81	0.81	0.89
Using AdaBoost	NB	0.87	0.49	0.62	0.55	0.75	0.82	0.74	0.81	0.77	0.86	0.78	0.81	0.79	0.80	0.85
	SVM	1	0.01	0.40	0.01	0.50	0.50	0.49	0.50	0.49	0.50	0.92	0.34	0.80	0.47	0.63
	BP	0.81	0.53	0.55	0.54	0.68	0.79	0.74	0.78	0.76	0.79	0.79	0.78	0.79	0.79	0.81
	RBF	0.85	0.47	0.58	0.52	0.74	0.78	0.76	0.77	0.77	0.86	0.76	0.78	0.76	0.77	0.84
	C4.5	0.79	0.46	0.48	0.47	0.70	0.75	0.75	0.75	0.75	0.84	0.81	0.78	0.80	0.79	0.88
	RF	0.89	0.42	0.62	0.50	0.74	0.83	0.74	0.81	0.78	0.86	0.84	0.81	0.83	0.82	0.89
	RT	0.79	0.45	0.48	0.46	0.63	0.71	0.70	0.71	0.70	0.72	0.74	0.74	0.74	0.74	0.76

The same thing occurs with other four datasets, which are depicted clearly in Figs 2–5. The higher detection rates are for the negative class and the lower detection rates are for the positive class which emphasize that the overall accuracy are biased towards negative class. Obviously, we can also deduce that all the used classifiers are very sensitive for imbalanced classes but the most influenced one is SVM which biased totally to the negative class and produced TP rates equal to zero.

The lower TP rates also produced lower precision and lower F-measure but Roc not affected as it increased by high values of TN rates.

Table 5
Performance of different classifiers on Pima data sets using different performance measures

Classifier		Original data distribution					Balancing data using undersampling					Balancing data using oversampling				
		TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC
NB		0.84	0.61	0.68	0.64	0.82	0.62	0.65	0.66	0.65	0.63	0.78	0.70	0.77	0.73	0.83
SVM		1	0	0	0	0.50	0	1	0.54	0.70	0.50	0.98	0.26	0.94	0.41	0.62
BP		0.83	0.61	0.66	0.63	0.79	0.66	0.72	0.71	0.71	0.76	0.68	0.84	0.73	0.78	0.83
RBF		0.87	0.54	0.69	0.61	0.78	0.67	0.73	0.72	0.72	0.77	0.74	0.74	0.74	0.74	0.81
C4.5		0.81	0.60	0.63	0.61	0.75	0.60	0.78	0.69	0.74	0.77	0.68	0.80	0.72	0.76	0.78
RF		0.85	0.52	0.64	0.57	0.79	0.68	0.71	0.72	0.71	0.77	0.79	0.78	0.79	0.79	0.86
RT		0.768	0.60	0.58	0.59	0.68	0.65	0.65	0.68	0.67	0.65	0.72	0.74	0.73	0.74	0.73
Using bagging	NB	0.85	0.61	0.69	0.65	0.82	0.71	0.70	0.73	0.72	0.79	0.79	0.71	0.77	0.74	0.83
	SVM	1	0	0	0	0.50	0.0	1	0.54	0.70	0.51	0.46	0.68	0.56	0.62	0.63
	BP	1	0	0	0	0.73	1	0	0	0	0.65	1	0	0	0	0.80
	RBF	0.87	0.55	0.70	0.62	0.81	0.62	0.79	0.71	0.74	0.79	0.73	0.77	0.74	0.76	0.83
	C4.5	0.82	0.60	0.65	0.62	0.80	0.67	0.76	0.73	0.74	0.78	0.75	0.82	0.77	0.79	0.86
	RF	0.84	0.58	0.66	0.62	0.81	0.64	0.77	0.71	0.74	0.79	0.76	0.84	0.78	0.81	0.89
	RT	0.85	0.52	0.64	0.57	0.79	0.68	0.71	0.72	0.71	0.77	0.79	0.78	0.79	0.79	0.86
Using	NB	0.85	0.60	0.68	0.64	0.80	0.71	0.73	0.75	0.78	0.78	0.74	0.77	0.75	0.76	0.83
	SVM	1	0	0	0	0.52	0.20	0.81	0.54	0.65	0.51	0.70	0.39	0.57	0.46	0.56
	BP	0.82	0.60	0.65	0.62	0.79	0.65	0.73	0.71	0.72	0.76	0.69	0.83	0.73	0.78	0.81
	RBF	0.85	0.57	0.67	0.62	0.80	0.66	0.73	0.71	0.72	0.76	0.70	0.77	0.73	0.75	0.81
	C4.5	0.79	0.61	0.60	0.61	0.78	0.64	0.74	0.70	0.72	0.72	0.73	0.78	0.75	0.76	0.84
	RF	0.84	0.55	0.64	0.59	0.78	0.68	0.68	0.71	0.70	0.74	0.79	0.78	0.79	0.79	0.87
	RT	0.76	0.57	0.56	0.56	0.66	0.62	0.65	0.66	0.65	0.63	0.72	0.73	0.72	0.73	0.72

Table 6
Performance of different classifiers on *Hepatitis* data sets using different performance measures

Classifier		Original data distribution					Balancing data using undersampling					Balancing data using oversampling				
		TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC
NB		0.89	0.63	0.61	0.62	0.85	0.91	0.88	0.90	0.89	0.96	0.87	0.84	0.87	0.86	0.91
SVM		1	0	0	0	0.50	0.34	0.50	0.43	0.46	0.42	0.95	0.51	0.92	0.65	0.73
BP		0.87	0.63	0.90	0.88	0.84	0.81	0.91	0.83	0.87	0.96	0.81	0.95	0.84	0.89	0.93
RBF		0.88	0.66	0.58	0.62	0.81	0.84	0.94	0.86	0.90	0.84	0.82	0.90	0.84	0.87	0.93
C4.5		0.90	0.44	0.54	0.48	0.70	0.88	0.84	0.87	0.86	0.80	0.83	0.87	0.84	0.85	0.86
RF		0.89	0.63	0.61	0.62	0.85	0.81	0.94	0.83	0.88	0.94	0.83	0.95	0.85	0.90	0.96
RT		0.85	0.50	0.46	0.48	0.67	0.78	0.81	0.79	0.80	0.80	0.85	0.83	0.86	0.84	0.84
Using	NB	0.89	0.66	0.62	0.64	0.89	0.88	0.94	0.88	0.91	0.96	0.86	0.86	0.86	0.86	0.91
bagging	SVM	1	0	0	0	0.50	0.23	0.78	0.50	0.61	0.47	0.85	0.53	0.79	0.64	0.72
	BP	0.033	1	1	0.06	0.59	0.23	1	0.56	0.72	0.89	0.06	0.99	0.52	0.69	0.61
	RBF	0.94	0.69	0.73	0.71	0.88	0.91	0.88	0.90	0.89	0.91	0.86	0.90	0.87	0.89	0.92
	C4.5	0.93	0.56	0.67	0.61	0.83	0.88	0.81	0.87	0.84	0.88	0.81	0.92	0.84	0.88	0.94
	RF	0.94	0.50	0.67	0.57	0.87	0.88	0.88	0.88	0.88	0.95	0.87	0.92	0.88	0.90	0.97
	RT	0.894	0.63	0.61	0.62	0.85	0.81	0.94	0.83	0.88	0.94	0.83	0.95	0.85	0.90	0.96
Using	NB	0.94	0.50	0.68	0.56	0.77	0.84	0.84	0.84	0.84	0.93	0.89	0.88	0.89	0.88	0.92
AdaBoost	SVM	0.99	0	0	0	0.53	0.47	0.56	0.51	0.54	0.49	0.96	0.42	0.92	0.58	0.69
	BP	0.90	0.53	0.59	0.56	0.81	0.81	0.91	0.83	0.87	0.86	0.86	0.95	0.87	0.91	0.91
	RBF	0.94	0.63	0.74	0.68	0.83	0.84	0.88	0.85	0.86	0.88	0.85	0.92	0.87	0.89	0.94
	C4.5	0.94	0.53	0.68	0.60	0.82	0.78	0.81	0.79	0.80	0.84	0.89	0.93	0.90	0.91	0.96
	RF	0.90	0.47	0.56	0.51	0.82	0.75	0.91	0.78	0.84	0.94	0.85	0.94	0.86	0.90	0.96
	RT	0.81	0.44	0.37	0.40	0.62	0.94	0.75	0.92	0.83	0.85	0.84	0.87	0.85	0.86	0.85

In the next phase, we have balanced data and applied undersampling by selecting randomly a subset from the majority (negative) class that equal to the size of minority (positive) class. We notice that there are clear improvements in the true positive rate however; the TN rates have become less. And this has degraded the other performance such as precision and f-measure. So the results have reflected that the using of undersampling resulted in loss of information by removing significant samples from the negative class. This case is clearly significant in insurance fraud but in the other low imbalanced ratio there is no clear degradation in TN rates. In the third phase, we have oversampled data by applying SMOTE. SMOTE used to generate new synthetic samples and added them to minority class. Using SMOTE has improved the performance of clas-

Table 7
Performance of different classifiers on *Haberman* fraud data sets using different performance measures

Classifier		Original data distribution					Balancing data using undersampling					Balancing data using oversampling				
		TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC	TNR	TPR	Prec	F-M	ROC
NB		0.94	0.21	0.58	0.31	0.65	0.93	0.75	0.91	0.82	0.88	0.88	0.36	0.76	0.49	0.67
SVM		0.99	0.03	0.50	0.05	0.51	0.60	0.82	0.67	0.73	0.71	0.72	0.76	0.73	0.74	0.74
BP		0.88	0.30	0.48	0.37	0.66	0.92	0.72	0.89	0.80	0.87	0.74	0.58	0.69	0.63	0.72
RBF		0.95	0.17	0.54	0.26	0.67	0.84	0.75	0.82	0.79	0.88	0.71	0.64	0.68	0.66	0.71
C4.5		0.87	0.30	0.45	0.36	0.61	0.90	0.67	0.87	0.76	0.82	0.70	0.71	0.70	0.70	0.74
RF		0.84	0.26	0.37	0.30	0.64	0.84	0.77	0.83	0.80	0.86	0.75	0.70	0.74	0.72	0.81
RT		0.77	0.30	0.32	0.31	0.53	0.74	0.78	0.75	0.76	0.76	0.73	0.72	0.23	0.72	0.73
Using	NB	0.94	0.20	0.55	0.29	0.66	0.92	0.75	0.90	0.82	0.89	0.89	0.36	0.76	0.50	0.67
Bagging	SVM	0.98	0.06	0.50	0.11	0.49	0.59	0.82	0.66	0.73	0.73	0.70	0.74	0.71	0.71	0.79
	BP	1	0	0	0	0.54	0.57	0.51	0.54	0.52	0.60	0.13	0.92	0.51	0.66	0.60
	RBF	0.94	0.19	0.54	0.28	0.68	0.82	0.77	0.81	0.79	0.88	0.74	0.63	0.71	0.67	0.71
	C4.5	0.88	0.22	0.40	0.29	0.64	0.92	0.72	0.89	0.80	0.85	0.70	0.78	0.72	0.75	0.80
	RF	0.84	0.25	0.35	0.29	0.66	0.83	0.75	0.81	0.78	0.86	0.72	0.81	0.74	0.77	0.82
	RT	0.84	0.26	0.37	0.30	0.64	0.84	0.77	0.83	0.80	0.86	0.75	0.70	0.74	0.72	0.81
Using	NB	0.91	0.28	0.52	0.37	0.67	0.93	0.75	0.91	0.82	0.84	0.76	0.55	0.69	0.61	0.66
AdaBoost	SVM	0.87	0.15	0.29	0.20	0.60	0.52	0.83	0.63	0.77	0.75	0.70	0.78	0.72	0.75	0.80
	BP	0.88	0.28	0.47	0.35	0.59	0.85	0.73	0.83	0.79	0.86	0.78	0.60	0.73	0.66	0.73
	RBF	0.94	0.21	0.55	0.30	0.67	0.76	0.74	0.75	0.75	0.83	0.72	0.64	0.69	0.66	0.72
	C4.5	0.84	0.42	0.48	0.45	0.63	0.92	0.72	0.89	0.80	0.85	0.71	0.74	0.71	0.73	0.77
	RF	0.82	0.33	0.40	0.36	0.63	0.72	0.75	0.73	0.74	0.85	0.75	0.71	0.73	0.72	0.79
	RT	0.76	0.26	0.28	0.27	0.59	0.76	0.77	0.76	0.76	0.85	0.75	0.73	0.74	0.73	0.80

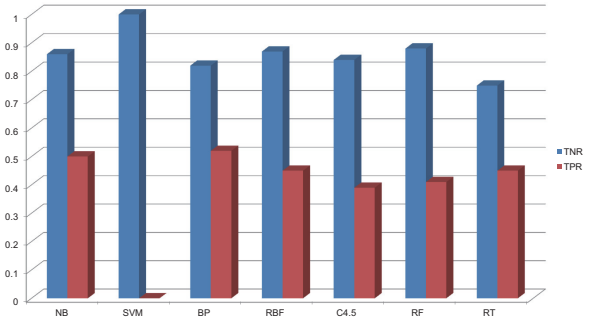

Fig. 2. The detection rates for positive and negative classes in German data set. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/HIS-150217)

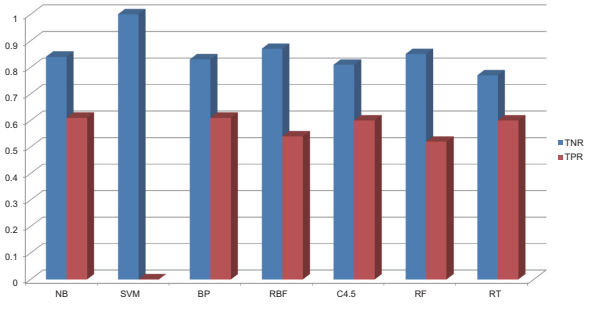

Fig. 3. The detection rates for positive and negative classes in Pima data set. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/HIS-150217)

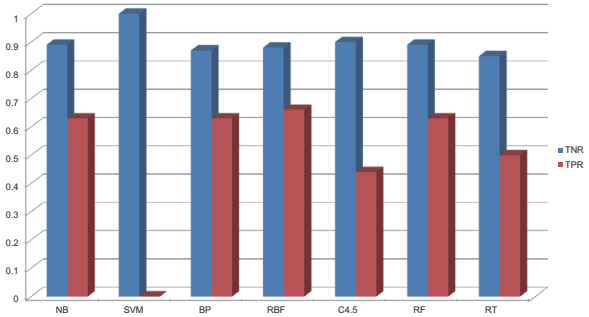

Fig. 4. The detection rates for positive and negative classes in Hepatitis data set. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/HIS-150217)

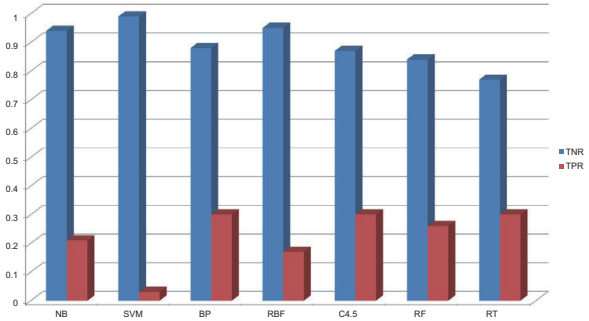

Fig. 5. The detection rates for positive and negative classes in Haperman data set. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/HIS-150217)

Table 8
Performance of the proposed method on different sets using different performance measures

Data set	Classifier	TNR	TPR	Prec	F-M	ROC
Insurance	NB	0.7	0.687	0.134	0.224	0.764
fraud	SVM	0.985	0.866	0.799	0.831	0.994
	BP	0.946	0.677	0.458	0.546	0.951
	RBF	0.918	0.889	0.909	0.899	0.96
	C4.5	1	1	0.998	0.999	1
	RF	1	1	0.998	0.999	1
	RT	1	1	0.998	0.999	1
German	NB	0.793	0.774	0.793	0.793	0.836
	SVM	0.891	0.895	0.882	0.889	0.909
	BP	0.891	0.895	0.882	0.889	0.93
	RBF	0.918	0.889	0.909	0.899	0.96
	C4.5	0.933	0.896	0.924	0.910	0.958
	RF	0.924	0.881	0.914	0.897	0.96
	RT	0.863	0.858	0.852	0.855	0.868
Pima	NB	0.733	0.729	0.733	0.731	0.801
	SVM	0.395	0.644	0.517	0.574	0.551
	BP	0.738	0.746	0.741	0.743	0.811
	RBF	0.768	0.803	0.776	0.790	0.852
	C4.5	0.785	0.801	0.789	0.795	0.866
	RF	0.788	0.764	0.783	0.773	0.85
	RT	0.873	0.697	0.681	0.689	0.685
Hepatitis	NB	0.919	0.875	0.913	0.894	0.919
	SVM	0.751	0.725	0.742	0.733	0.802
	BP	0.889	0.948	0.892	0.919	0.941
	RBF	0.899	0.917	0.898	0.907	0.954
	C4.5	0.869	0.927	0.873	0.899	0.945
	RF	0.909	0.948	0.910	0.929	0.958
	RT	0.909	0.948	0.910	0.929	0.958
Haberman	NB	0.927	0.753	0.910	0.824	0.881
	SVM	0.598	0.815	0.667	0.733	0.706
	BP	0.915	0.716	0.892	0.795	0.871
	RBF	0.841	0.753	0.824	0.787	0.883
	C4.5	0.902	0.667	0.871	0.755	0.818
	RF	0.841	0.765	0.827	0.795	0.857
	RT	0.744	0.778	0.750	0.764	0.764

sifiers in low imbalanced data sets. However, in insurance fraud data set we realize that there is a clear degradation on the detection rates for the negative class and there are proportional improvements on the true positive rates. Hence to deal with imbalanced data in this paper we propose a hybrid ensemble using multi resampling method that integrated both undersampling and oversampling methods to get benefits and reduce drawbacks caused by each of them. Our proposed solution described in the following algorithm:

Input: training data set (DT) and testing dataset (DS) and classifier C.

Split training data into majority (M) and minority (N) class

Calculate IR (imbalance ratio) = size M/size N

For i=1 to ceil(IR+2)

*Ni = SMOTE(N,100/i);/*oversampling the minority class using SMOTE*/*

*Mi = RUS(M,size(Ni));/*random Undersampling the majority class*/*

*DTi = Ni U Mi;/*the training set for the component i*/*

Train the classifier C using DTi;

Evaluate the model using DS;

Compute TPR, TNR, precision, F-measure, ROC.

Output: average for all measures

The basic idea for the proposed method is applying multiple The basic idea for the proposed method is applying multiple resampling methods at various rates to construct several balanced datasets. Instead of designing multiple classifiers with the same dataset, we can manipulate the training set by resampling the original data using undersampling and oversampling.

Also, in our proposed method we integrate ensemble of multiple classifiers which is one of the popular techniques being used recently to increase and boost the performance of weak learners.

Firstly we have used SMOTE to oversample the minority class by adding new synthetic samples. SMOTE finds the k nearest neighbors for each minority sample according to the percentage of increase. Then, it selects randomly a point that lie on the line between each pair of nearest neighbors to generate the new added minority samle. In our method the percentage of increase is controlled by the number of repetition for the loop to generate different set in each repetition and overcome overfitting for the new model which is a defection for oversampling.

In the next step we have applied undersampling by selecting randomly a subset from the majority class by size equal to the new oversampled minority class that obtained in the previous step. The objective of this step is to overcome the defection of undersampling which causes loss of information by forming multiple subsets that contain different samples from majority class.

Then, the new trained data set obtained by combining new balanced sets for minority and majority subsets. Then these multiple balanced datasets used to train the base classifiers of ensemble. The number of the base classifiers that formed for ensemble depends on the imbalance ratio between majority and minority class. After that, the trained ensemble model is evaluated on the testing data. The predicted class for any testing sample is calculated using average function for the output of all base classifiers with threshold value 0.5. Finally, the output is performance measures for the ensemble. The results our proposed method are depicted in Table 8.

By comparing the performance of our method with the basic classifiers, Bagging, AdaBoost or SMOTE

with/out Bagging or AdaBoost, our proposed method achieved superior performance results. From the other side, our proposed method mostly gives TPRs closely to those performed by using undersampling with/out Bagging/AdaBoost. However, our proposed method give higher TPRs without decreases the detection rates for the negative class or caused bias to the one class.

In terms of precision, our proposed method achieved competitive results on most data sets. We note that using Bagging with SVM and BP in insurance fraud data set and using BP with bagging in hepatitis data set resulted in a higher precision than our method but this occur due to the total biased to the positive class.

In terms of f-measure, our proposed method performed a significant improvements and superior results in comparison to the other tested methods.

In terms of ROC, our proposed method yields a good results compared to the other tested methods.

4. Conclusions

In this paper, we studied empirically the effects of using resampling (undersampling and oversampling) methods with homogenous ensemble methods Bagging and AdaBoost in imbalanced data sets. From our experiments we can conclude that using Bagging or AdaBoost have no significant effect in improving the classifiers performance when using the original distribution for the imbalanced data sets. We proposed a hybrid ensemble approach that integrated both undersampling and oversampling to get benefits and reduce drawbacks caused by each of them. The proposed approach has improved the performance of classifiers even those most sensitive to imbalanced class probelm. And, there is a significant improvement in the performance measures, especially in high class imbalanced data set.

References

[1] A. Fernando, E. Barrenechea, H. Business, F. Herrera and M. Galar, A Review on ensembles for the class Imbalance Problem, *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews* **42**, 2012.

[2] P. Kang and S. Cho, EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems, in International Conference on Neural Information Processing, 2006.

[3] Y. Zhang and D. Wang, A Cost-Sensitive Ensemble Method for Class-Imbalanced Datasets, Abstract and Applied Analysis, 2013.

[4] T.M. Khoshgoftaar, J.V. Hulse, A. Napolitano and C. Seiffert, Building Useful Models from Imbalanced with Sampling and Boosting, in Proceedings of The Twenty-First International FLAIRS Coference, 2008, pp. 306–311.

[5] K.L. Chan, W. Fang and P. Li, Hybrid Kernel Machine Ensemble for Imbalanced Data Sets, in 18th International Conference on Pattern Recognition, 2006, pp. 1108–1111.

[6] T.M. Khoshgoftaar, J.V. Hulse, A. Napolitano and C. Seiffert, RUSBoost: A Hybrid Approach to Alleviating Class Imbalance, *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans* **40**(**1**), 185–197.

[7] M. Govindaraj and S. Lavanya, A Combined Boosting And Sampling Approach For Imbalanced Data Classification, *International Journal of Advanced Research in Data Mining and Cloud Computing* **1**(1) (July 2013), 44–50.

[8] B. Yuan and X. Ma, Sampling + Reweighting: Boosting the Performance of AdaBoost on Imbalanced Datasets, IEEE World Congress on Computational Intelligence, pp. 2680–2685, June 2012.

[9] Y. Ren, P. Jia and H. Xiong, A Novel Classification Approach for C2C E-Commerce Fraud Detection, *International Journal of Digital Content Technology and its Applications* **7**(1) (January 2013), 504–511.

[10] J. Wu, Z. Zhou and X. Liu, Exploratory Undersampling for Class-Imbalance Learning, *IEEE Transactions On Systems, Man And Cybernetics – Part B 1* **39**(2) (2008), 500–539.

[11] L. Tianyu, Imbalance learning for fault diagnosis gearbox in wind turbine, *Journal of Chemical and Pharmaceutical Research* **7**(3) (2015), 1287–1292.

[12] T.M. Khoshgoftaar, R. Wald and K. Gao, The Use of Under-And Oversampling with in Ensemble Feature Selection and Classification for Software Quality, *International Journal of Reliability, Quality and Safety Engineering* **21**(1) (2014).

[13] J. Burez and D.V. Poel, Handling class imbalance in customer churn prediction, *Experts System with Applications* **36** (2009), 4626–4636.

[14] H. Gue and H. Viktor, Learning from Imbalanced Data Sets with Boosting and Generation: The DataBoost-IM Approach, *SIGKDD Explorations* **6**, 30–39.

[15] Y. Zhang and B. Luo, Parallel Classification Ensemble with Hierarchical Machine Learning for Imbalanced Classes, in The seventh International conference on Machine Learning and Cybernetics, Kunming, 2008.

[16] S. Chakraborty, M.L. Popescu and M. Khalilia, Predicting disease risks from highly imbalanced data using random forest, *BMC Medical Informatics and Decision Making* **11**(51) (2011).

[17] Q. Xu, L. Zhou and H. Wang, Seminal Quality Prediction Using Clustering-Based Decision Forests, *Algorithms* **7** (2014), 405–417.

[18] M. Wo'zniak, G. Schaefer and B. Krawczyk, Cost Sensitive Decision Tree Ensembles for Effective imbalanced Classification, Applied Soft Computing, pp. 554–562.

[19] K.W. Bower, L.O. Hall, W.P. Kegelemer and N.V. Chawla, SMOTE: Synthetic Minority Over-Sampling Technique, *Artificial Intelligence Research* **16** (2002), 321–357.

[20] M. Vasu and V. Ravi, A hyprid under-sampling approach for mining unbalanced datasets: Applications to banking and insurance, *Int J. Data Mining Modelling and Management* **3** (2011), 75–105.

[21] (2013) University of California at Irvine (UCI) repository. [Online]. https://archive.ics.uci.edu/ml/machine-learning-databases/.