

Cluster Based Under-Sampling for Unbalanced Cardiovascular Data

M. Mostafizur Rahman¹ and D. N. Davis²

Abstract—Most medical datasets are not balanced in their class labels. Indeed in some cases it has been noticed that the given class labels do not accurately represent characteristics of the data record. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. This is because they aim to optimize the overall accuracy without considering the relative distribution of each class. In this paper we propose a cluster based under sampling technique that solves the class imbalance problem for our cardiovascular data. It shows significant better performance than existing methods.

Keywords— class imbalance, under-sampling, over-sampling, clustering, SMOTE.

I. INTRODUCTION

A well balanced training dataset is very important for creating a good training set for the application of classifiers. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. They aim to optimize the overall accuracy without considering the relative distribution of each class [1]. Typically real world data are usually imbalanced and it is one of the main causes for the decrease of generalization in machine learning algorithms [2]. Conventional learning algorithms do not take into account class imbalance; giving the same attention to a data record irrespective of whether it is from the majority class or the minority class. When the imbalance is massive, it is hard to build a good classifier using conventional learning algorithms [3]. Conventional classification algorithms like neural networks, decision tree, Native Bayes and K-nearest neighbour assume that all classes have a similar number of records in the training data and the cost derived from all the classes is equal [1-4]. Actually, the cost in miss-predicting minority classes is higher than that of the majority class for many class imbalance datasets; this is particularly so in medical datasets where high risk patients tend to be the minority class. Indeed in many cases the class labels do not accurately reflect the nature of a patient. Some patients die for some reason other than the target cause and some patients are alive by chance. Therefore, there is a need of a good sampling technique for such datasets where the target classes are not balanced and the given labels are not always appropriate.

Sampling strategies have been used to overcome the class imbalance problem by either eliminating some data from the majority class (under-sampling) or adding some artificially generated or duplicated data to the minority class (over-sampling) [4].

Over-sampling techniques [5] increase the number of minority class members in the training set. The advantage of over-sampling is that no information from the original training set is lost since all members from the minority and majority classes are kept. However, the disadvantage is that

the size of the training set is significantly increased [5]. Random over-sampling is the simplest approach to over-sampling, where members from the minority class are chosen at random; these randomly chosen members are then duplicated and added to the new training set [6]. Chawla [5] proposed an over-sampling approach called SMOTE in which the minority class is over-sampled by creating “synthetic” examples rather than over-sampling with duplicated real data entries. Depending upon the amount of over-sampling required, neighbours from the k nearest neighbours of a record are randomly chosen. SMOTE blindly generates synthetic minority class samples without considering majority class samples; this may cause overgeneralization [7].

In summary, over-sampling may cause longer training time and over-fitting. The alternative to over-sampling is under-sampling. If we do not consider the time taken to resample, under-sampling betters over-sampling in terms of time and memory complexity [1]. Drummond and Holte [8] showed that random under-sampling yields better minority prediction than random over-sampling. Under-sampling is a technique to reduce the number of samples in the majority class, where the size of the majority class sample is reduced from the original datasets to balance the class distribution. One simple method of under-sampling (random under-sampling) is to select a random subset of majority class samples and then combine them with minority class sample as a training set [7]. Many researchers have proposed more advanced ways of under-sampling the majority class data. Chyi [9] developed an under-sampling approach based on distance uses four distinct modes to select the representative samples from the majority class: the nearest; the farthest; the average nearest; and the average farthest distances between Minority and Majority classes.. For every minority class sample in the dataset, the first method (“nearest”) calculates the distances between all majority class samples and the minority class samples, and selects k majority class samples which have the smallest distances to the minority class sample. If there are n minority class samples in the dataset, the “nearest” method would finally select ($k \times n$) majority class samples ($k > 1$). However with this method, some samples within the selected majority class samples might be duplicated. The “farthest” method selects the majority class samples which have the farthest distances to each minority class sample. The third method (“average nearest”) calculates, for every majority class sample in the dataset, the average distances between one majority class sample and all minority class samples. This method selects the majority class samples which have the smallest average distances. The last method “average farthest” is similar to the “average nearest” method. It selects the majority class samples which have the farthest average distances from all the minority class samples. The Chyi under-sampling approaches [9] spend a lot of time selecting the majority class samples in the large dataset, and they are not efficient in real applications [7].

In the rest of this paper we present a cluster based under-sampling technique to balance cardiovascular data.

Manuscript received March 22, 2013; revised April 06, 2013. M. Rahman is a PhD student in the Department of Computer Science, University of Hull, UK, (e-mail: M.M.Rahman@2009.hull.ac.uk). Darryl N. Davis is the Director of Research, Department of Computer Science, University of Hull, UK (e-mail: D.N.Davis@hull.ac.uk).

II. CLUSTER BASED UNDER-SAMPLING

Down-sizing the majority class results in a loss of information that may result in overly general rules [10]. In order to overcome this drawback of the under-sampling approach Yen and Lee (2009) proposed cluster-based under-sampling. Their approach is to first cluster all the training samples into K clusters (they have run the experiment with different K values to observe the outcome) then choose appropriate training samples from the derived clusters. The main idea is that there are different clusters in a dataset, and each cluster seems to have distinct characteristics. If a cluster has more majority class samples and less minority class samples, it will behave like a majority class sample. On the other hand, if a cluster has more minority class samples and less majority class samples, it does not hold the characteristics of the majority class samples and behaves more like the minority class samples. Therefore, their approach selects a suitable number of majority class samples from each cluster by considering the ratio of the number of majority class samples to the number of minority class samples in the derived cluster [7].

They first cluster the full data to K clusters. A suitable number (M) of majority class samples from each cluster are then selected by considering the ratio of the number of majority class samples to the number of minority class samples in the cluster. The number M is determined by equation 1, and they randomly choose the M numbers of majority class samples from each cluster. In the i^{th} cluster ($1 \leq i \leq K$) the $Size_{MA}^i$ will be:

$$Size_{MA}^i = (m \times Size_{MI}^i) \times \frac{Size_{MA}^i}{\sum_{i=1}^K Size_{MA}^i} \times \frac{Size_{MI}^i}{Size_{MI}^i} \quad (1)$$

This approach may be suitable for datasets where class labels are confidently defined and truly reflect the property of the labeled class. But as we mention earlier that in some cases, especially for medical datasets, there is no guarantee that the class labels are truly reflecting the actual characteristics of that record.

III. PROPOSED CLUSTER BASED UNDER-SAMPLING METHOD

Our approach to under-sampling is different to the approach of Yen and Lee (2009). As shown in the figure 2, we first separated the data in to two sets, one subset has all the majority class samples and the other subset has the entire minority class sample. Next we cluster the majority class samples to K clusters ($K > 1$) then make K subsets of majority class samples, where each cluster is considered to be one subset of the majority class. The aim was not to derive a majority and minority class ratio of 1:1; we just wanted to reduce the gap between the numbers of majority class samples to the numbers of minority class samples. All the subsets of the majority class are separately combined with the minority class samples to make K different training data sets (The K value is dependent on the data domain, in our implementation

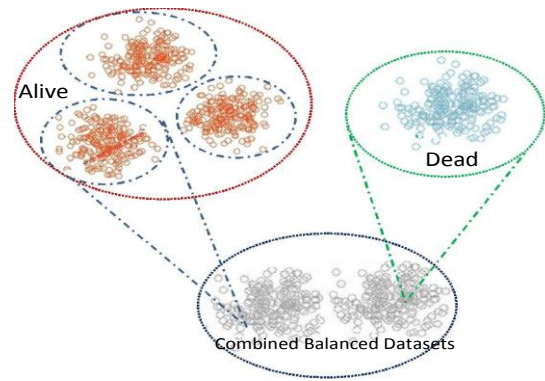


Figure 1. Example cluster based under-sampling of cardio vascular dataset

(the K value was 3). All the combined datasets are classified with decision tree [11] and Fuzzy Unordered Rule Induction Algorithm [12]. We kept the datasets that gave the highest accuracy with the majority of the classifiers for further data mining processes.

For experiments we prepared several datasets using different clustering and classified using decision tree. The experimental outcomes are discussed in the result section.

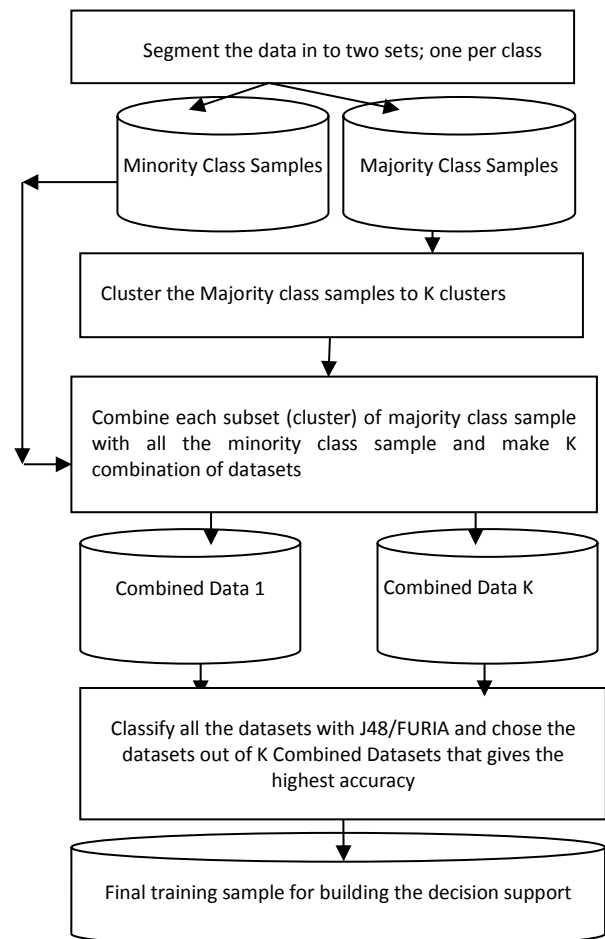


Figure 2. Cluster Based Under-Sampling Process

IV. EXPERIMENTS AND ALGORITHMS

We have used two cardiovascular datasets from Hull and Dundee clinical sites. K-Means [13] clustering is used to cluster the majority samples. For choosing the best subset, we have used decision tree [11] and Fuzzy Unordered Rule Induction Algorithm [12] as classifiers.

A. Cardiovascular Data

The Hull site data includes 98 attributes and 498 cases of cardiovascular patients and the Dundee site data includes 57 attributes, and 341 cases from cardiovascular patients. After combining the data from the two sites, 26 matched attributes are left.

After combining the data and removing the redundant attributes we found that out of 26 attributes with 839 records. Out of all 18 attributes have a missing value frequency from 1% to 30%. Out of 839 records, 612 records have 4% to 56% missing values in their attributes. We removed 7 records those 50% of attributes have missing value.

From these two data sets, we prepared a combined dataset having 26 attributes with 823 records. Out of 823 records, 605 records have missing values and 218 records do not have any missing values. The missing values are imputed by the machine learning technique proposed in our previous work [14]. Among all the records 120 patients are dead and 703 patients are alive, with the majority and minority ratio 6:1. For this experiment according to clinical risk prediction model (CM1) [15], patients with status "Alive" are consider to be "Low Risk" and patients with status "Dead" are consider to be "High Risk". The "High Risk" profiles are assumed to represent characteristics of concern; however, some alive (i.e. "Low Risk") patients will have similar profiles. Conversely, some "High Risk" patients display data characteristics similar to alive (i.e. "Low Risk") patients. We accept, for the time being, this flaw in labelling the data.

B. K-means Clustering

K-means is one of the simplest unsupervised learning algorithms, first proposed by Macqueen in 1967, which has been used by many researchers to solve some well-known clustering problems [16]. The technique classifies a given data set into a certain number of clusters (assume k clusters). The algorithm first randomly initializes the clusters center. The next step is to calculate the distance between an object and the centroid of each cluster. Next each point belonging to a given data set is associated with the nearest centre. The cluster centres are then re-calculated. The process is repeated with the aim of minimizing an objective function known as squared error function given by:

$$J v = \sum_{i=1}^c \sum_{j=1}^{c_i} ||x_i - v_j||^2 \quad (2)$$

Where, $|x_i - v_j|$ is the Euclidean distance between the data point x_i and cluster centre v_j , c_i is the number of data points in i^{th} cluster and c is the number of cluster centers.

C. Overview of FURIA

Fuzzy Unordered Rule Induction Algorithm (FURIA) is a fuzzy rule-based classification method, which is a modification and extension of the state-of-the-art rule learner RIPPER [17]. Fuzzy rules are obtained through replacing intervals by fuzzy intervals with trapezoidal membership functions [18].

$$I^F v \stackrel{\text{df}}{=} \begin{cases} 1 & \phi^{c,L} \leq v \leq \phi^{c,U} \\ \frac{v - \phi^{s,L}}{\phi^{c,L} - \phi^{s,L}} & \phi^{s,L} \leq v \leq \phi^{c,L} \\ \frac{\phi^{s,U} - v}{\phi^{s,U} - \phi^{c,U}} & \phi^{c,U} \leq v \leq \phi^{s,U} \\ 0 & \text{else} \end{cases} \quad (3)$$

Where $\phi^{c,L}$ and $\phi^{c,U}$ are the lower and upper bound of the membership of the fuzzy sets. For an instance $x = (x_1, \dots, x_n)$ the degree of the fuzzy membership can be found using the formula [18]:

$$\mu_{r^F} x = \prod_{i=1, \dots, k} I_i^F(x_i) \quad (4)$$

For fuzzification of a single antecedent, only relevant training data D_T^i is considered and data are partitioned into two subsets and rule purity is used to measure the quality of the fuzzification [18]:

$$D_T^i = x = x_1 \dots x_k \in D_T^i | I_j^F x_j > 0 \text{ for all } j \neq i \subseteq \quad (5)$$

$$P_{ur} = \frac{p_i}{p_i + n_i} \quad (6)$$

Where,

$$p_i \stackrel{\text{def}}{=} \sum_{x \in D_{T+}^i} \mu_{A_i} A$$

$$n_i \stackrel{\text{def}}{=} \sum_{x \in D_{T-}^i} \mu_{A_i} A$$

The fuzzy rules $r_1^j \dots r_k^j$ are learnt for the class λ_j , the support of this class is defined by [18]:

$$s_j x = \sum_{i=1, \dots, k} \mu_{r_i^j} x \cdot CF r_i^j \quad (7)$$

Where, the certainty factor of the rule is defined as

$$CF r_i^j = \frac{2 \frac{|D_T^j|}{|D_T|} + \sum_{x \in D_T^j} \mu_{r_i^j} x}{2 + \sum_{x \in D_T} \mu_{r_i^j} x} \quad (8)$$

The use of the algorithm in different areas of data mining can be found in [12, 18, 19].

C. Decision Tree

The decision tree classifier is one of the most widely used supervised learning methods. A decision tree is expressed as a recursive partition of the instance space. It consists of a directed tree with a "root" node with no incoming edges and all the other nodes have exactly one incoming edge [16]. Decision trees models are commonly used in data mining to examine the data. The induced tree and its associated rules will be used to make predictions [11]. Ross Quinlan introduced a decision tree algorithm known as Iterative Dichotomiser (ID3) in 1979. C4.5, as a successor of ID3, is the most widely-used decision tree algorithm [20]. The major advantage to the use of decision trees is the class-focused visualization of data. This

visualization is useful in that it allows users to readily understand the overall structure of data in terms of which attribute mostly affects the class. Typically the goal is to find the optimal decision tree by minimizing the generalization error [16]. The algorithms introduced by Quinlan [21, 22] have proved to be an effective and popular method for finding a decision tree to express information contained implicitly in a data set. WEKA [23] makes use of an implementation of C4.5 algorithm called J48 which has been used for all of our experiments.

D. Classifier Performance Evaluation

The performance of the classification is evaluated by accuracy (ACC); sensitivity (Sen); specificity (Spec) rates, and the positive predicted value (PPV) and negative predicted value (NPV), based on values residing in a confusion matrix.

Assume that the cardiovascular classifier output set includes two typically risk prediction classes as: “High risk”, and “Low risk”. Each pattern x_i ($i=1, 2..n$) is allocated into one element from the set (P, N) (positive or negative) of the risk prediction classes. Hence, each input pattern might be mapped into one of four possible outcomes: true positive- true high risk (TP)- when the outcome is correctly predicted as High risk; true negative- true low risk (TN)- when the outcome is correctly predicted as Low risk; false negative-false Low risk (FN)- when the outcome is incorrectly predicted as Low risk, when it is High risk (positive); or false positive- false high risk (FP) - when the outcome is incorrectly predicted as High risk, when it is Low risk (negative). The set of (P, N) and the predicted risk set can be built as a confusion matrix.

		Predicted classes	
		High risk	Low risk
Expected/ Actual Classes	High risk	TP	FN
	Low risk	FP	TN

Figure 3. Confusion Matrix

The accuracy of a classifier is calculated by:

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \quad (9)$$

The sensitivity is the rate of number correctly predicted “High risk” over the total number of correctly predicted “High risk” and incorrectly predicted “Low risk”. It is given by:

$$Sen = \frac{TP}{TP+FN} \quad (10)$$

The specificity rate is the rate of correctly predicted “Low risk” over the total number of expected/actual “Low risk”. It is given by:

$$Spec = \frac{TN}{TN+FP} \quad (11)$$

Higher accuracy does not always reflect a good classification outcome. For clinical data analysis it is important to evaluate the classifier based on how well the classifier predicts the “High Risk” patients. In many cases it has been found that the classification outcome is showing good accuracy as it can predict well the low risk patients (majority class) but failed to predict high risk patients (the minority class).

V. RESULTS

We tried different methods in preparing a closely balanced datasets through clustering as outlined above. The method never runs with the aim of having class ratio 1:1. Our aim was to reduce the ratio gap between the majority and minority classes. The results are presented in table 1 and 2.

We made six datasets with different combinations of the clusters from majority and minority class samples and named as D1...D6, as described in table 1. For exploring different alternatives we also tried to reduce further the ratio gap of majority class samples to minority class samples. In order to understand the quality of the training sample we also cluster the minority samples into three clusters and group them by different combinations with the clusters of majority class samples. An example of such a dataset is D2. We took the dataset D2 that has the best classification sensitivity among all the other datasets, we further cluster the majority class samples of D2 and select one cluster out of three clusters and combine with the minority class sample of the D2 and made another sample datasets called “D6”.

We also made two more datasets using the under-sampling by clustering method proposed by Yen and Lee (2009). The first dataset (K3M1Yen) was produced by separating the full data to 3 clusters and collected the majority class samples using equation 1 with the majority and minority ratio 1:1 (M=1). The second dataset (K3M2Yen) was produced by separating the full data to 3 clusters and collected the majority class samples using equation 1 with the majority and minority ratio 2:1 (M=2). The datasets are classified using J48 and FURIA and results are presented in tables 2 and 3.

TABLE 1.
THE DESCRIPTIONS OF THE DATASETS

Data	Ratio	Description
D1:	2 : 1	Data consist of all the minority class samples (“dead”) and one cluster of majority class records out of three clusters made by K-Mean.
D2:	2.4 : 1	Data consist of combination of two clusters of the minority class samples and one cluster of majority class samples. Clusters are made with simple k-mean for both of the classes (K=3).
D3:	3 : 1	Data consist of combination of all the minority class samples with randomly (random cut 1) selected samples from majority class sample.
D4:	3 : 1	Data consist of combination of all the minority class samples with randomly (random cut2) selected samples from majority class sample.
D5:	6 : 1	Original data with full samples.
D6:	1.8 : 1	Majority samples of the data set D2 are clustered in to 3 cluster and each clusters are combined with the minority samples.
K3M1Yen	1 : 1	Majority and minority ratio 1:1 (M=1) using Yen and Lee (2009)
K3M2Yen	2 : 1	Majority and minority ratio 2:1 (M=2) using Yen and Lee (2009)

TABLE 2.
CLASSIFICATION OUTCOME OF FURIA

Data Sets	ACC%	SEN%	SPEC%	PPV%	NPV%
D1	85.89	64.17	98.12	95.06	82.94
D2	92.11	79.78	97.21	92.21	92.07
D3	74.68	11.67	96.29	51.85	76.07
D4	70.82	15.83	89.52	33.93	75.78
D5	66.71	30.00	72.97	15.93	85.93
D6	96.39	91.01	99.38	98.78	95.21
K3M1Yen	61.48	67.50	55.65	59.56	63.89
K3M2Yen	60.39	22.50	79.66	36.00	66.90

TABLE 3.
CLASSIFICATION OUTCOME OF DECISION TREE

Data Sets	ACC%	SEN%	SPEC%	PPV%	NPV%
D1	84.08	67.50	93.43	85.26	83.61
D2	92.05	83.15	95.77	89.16	93.15
D3	67.66	35.83	78.57	36.44	78.13
D4	66.60	33.33	77.90	33.90	77.46
D5	79.59	20.00	89.76	25.00	86.80
D6	97.59	93.26	100	100	96.39
K3M1Yen	51.64	52.50	50.81	50.81	52.50
K3M2Yen	59.55	39.17	69.92	39.83	69.33

From the table 2 and 3 we can see that the original unbalance dataset D5 has accuracy of 66.71% with FURIA classification and 79.59 % with decision tree classification. But for both of the classifiers the sensitivity value is very poor (30% and 20%). The accuracy is high because the classifier was able to classify the majority class (*Alive*) sample well (72.97% and 89.76%) but failed to classify the minority. Dataset D1 where data are balanced by clustering the majority class samples and combining all the minority samples shows better classification outcome than the original unbalance data. With the FURIA and decision tree classification of the D1 dataset, we found the sensitivity value 0.642 with the decision tree and 0.675 with the FURIA. The classification outcome of the D1 is 2 to 3 times higher than the original datasets. The datasets prepared by the method proposed by Yen and Lee (2009) could show some increase in the sensitivity value but the accuracy was dropped and overall performance was not good. Under sampling by random cut D3 and D4 also disappointed by its accuracy and sensitivity values.

It is observed from the experiments that the majority and minority ratio is not only the issue of building a good prediction model. There is also a need of good training samples those should have true property of the class label assign to them. As we discuss early that most of the time the class labels of clinical dataset do not reflect the true property of the class. The majority and minority ratio of D1, D2 and D6 are very close but the classification outcomes are not similar. Although the majority minority ratio is almost same but there is a big difference of the classification accuracy, sensitivity and specificity of D1 and D6 can be noticed in the table 2 and 3. The dataset "K3M1Yen" prepared by the method proposed by Yen and Lee (2009) has 1:1 ratio but still have less classification outcome than other datasets.

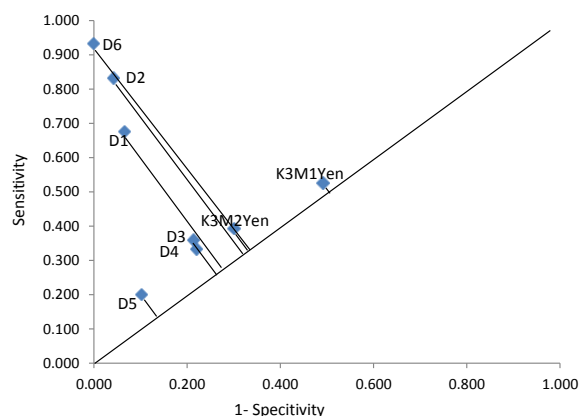


Figure 4. ROC of Decision Tree Classification.

If we analyse the ROC [24] space for all datasets classified with decision tree plotted in figure 4 and FURIA plotted in figure 5, we will find that overall accuracy of all the datasets are above the random line and the datasets D1, D2 and D6 which are prepared by our proposed method are having highest accuracy than all the other datasets.

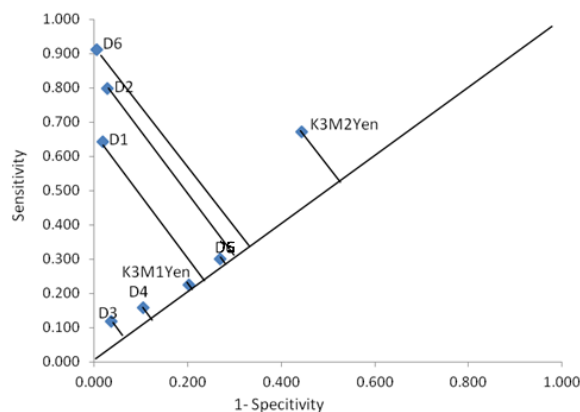


Figure 5. ROC of FURIA Classification

CONCLUSIONS

Class imbalance is a common problem with most medical datasets. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. Sampling strategies have been used to overcome the class imbalance problem by either over-sampling or under-sampling. Many researchers proposed different methods of under-sampling the majority class sample to balance the data. We proposed a cluster based under-sampling method that not only can balance the data but also can choose good quality training set data for building classification models. The outcome labels of most of the clinical datasets are not always appropriate. If we consider the cardiovascular risk based on dead or alive status of previous patients records, some of the patients may have died with some other cause and some are alive by chance. The proposed method is found to be useful for such datasets where the given class labels are not always appropriate and truly reflect the underlying characteristics of the patient record.

In summary, we suggest the techniques used here are of benefit for problematic data and can help to alleviate the class imbalance problems typically found in clinical datasets and data from other domains.

REFERENCE

- [1] Y. Liu *et al.*, "Combining integrated sampling with SVM ensembles for learning from imbalanced datasets," *Information Processing & Management*, vol. 47, no. 4, pp. 617-631, Jul, 2011.
- [2] m.-s. Kim, "An Effective Under-Sampling Method for Class Imbalance Data Problem," in 8th International Symposium on Advance intelligent System (ISIS 2007), 2007.
- [3] Z. Yan-Ping, Z. Li-Na, and W. Yong-Cheng, "Cluster-based majority under-sampling approaches for class imbalance learning," pp. 400-404.
- [4] R. Laza *et al.*, "Evaluating the effect of unbalanced data in biomedical document classification," *Journal of integrative bioinformatics*, vol. 8, no. 3, pp. 177, 2011 Sep, 2011.
- [5] N. V. Chawla *et al.*, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.
- [6] Y. Zhai *et al.*, "An Effective Over-sampling Method for Imbalanced Data Sets Classification," *Chinese Journal of Electronics*, vol. 20, no. 3, pp. 489-494, Jul, 2011.
- [7] S.-J. Yen, and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, pp. 5718-5727, 2009.
- [8] C. Drummond, and R. C. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling," in Workshop on Learning from Imbalanced Data Sets II, 2003.
- [9] Y.-M. Chyi, "Classification analysis techniques for skewed class distribution problems," Department of Information Management, National Sun Yat-Sen University, 2003.
- [10] J. Zhang, and I. Mani, "kNN approach to unbalanced data distributions: A case study involving information extraction," ICML'2003 workshop on learning from imbalanced datasets, 2003.
- [11] R. C. Barros *et al.*, "A Survey of Evolutionary Algorithms for Decision-Tree Induction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 3, pp. 291-312, 2012.
- [12] F. Lotte, A. Lecuyer, and B. Arnaldi, "FuRIA: A Novel Feature Extraction Algorithm for Brain-Computer Interfaces using Inverse Models and Fuzzy Regions of Interest," in 3rd International IEEE/EMBS Conference on Neural Engineering, CNE '07 2007, pp. 175-178.
- [13] J. Han, and M. Kamber, *Data mining : concepts and techniques*, Amsterdam [u.a.]: Elsevier/Morgan Kaufmann, 2011.
- [14] M. Rahman, M., and D. Davis, N., "Fuzzy Unordered Rules Induction Algorithm Used as Missing Value Imputation Methods for K-Mean Clustering on Real Cardiovascular Data."
- [15] D. N. Davis, and T. T. T. Nguyen, "Generating and Verifying Risk Prediction Models Using Data Mining (A Case Study from Cardiovascular Medicine)," in European Society for Cardiovascular Surgery 57th Annual Congress of ESCVS, Barcelona Spain, 2008.
- [16] O. Maimon, and L. Rokach, *Data mining and knowledge discovery handbook*, Berlin: Springer, 2010.
- [17] J. h. Sun, and X. l. Xu, "Large Rotating Machinery Fault Diagnosis and Knowledge Rules Acquiring Based on Improved RIPPER." pp. 549-552.
- [18] J. Hühn, and E. Hüllermeier, "Fuzzy Unordered Rules Induction Algorithm," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293-319, 2009.
- [19] F. Lotte, A. Lecuyer, and B. Arnaldi, "FuRIA: An Inverse Solution Based Feature Extraction Algorithm Using Fuzzy Set Theory for Brain-Computer Interfaces," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3253-3263, 2009.
- [20] I. Yoo *et al.*, "Data mining in healthcare and biomedicine: a survey of the literature," *J Med Syst*, vol. 36, no. 4, pp. 2431-48, Aug, 2012.
- [21] J. R. Quinlan, *Induction of decision trees*, [Broadway, N.S.W., Australia]: New South Wales Institute of Technology, School of Computing Sciences, 1985.
- [22] J. R. Quinlan, *C4.5 : programs for machine learning*, San Mateo: Morgan Kaufmann, 1993.
- [23] R. R. Bouckaert *et al.*, "WEKA-Experiences with a Java Open-Source Project," *Journal of Machine Learning Research*, vol. 11, pp. 2533-2541, Sep, 2010.
- [24] T. C. W. Landgrebe, and R. P. W. Duin, "Efficient Multiclass ROC Approximation by Decomposition via Confusion Matrix Perturbation Analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 810-822, 2008.