Abstract: Most medical datasets are not balanced in their class labels. Furthermore, 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. The class imbalance problem can also affect the feature selection process. In this paper we propose a cluster based under-sampling technique that solves the class imbalance problem for our cardiovascular data. Data prepared using this technique shows significant better performance than existing methods. A feature selection framework for unbalanced data is also proposed in this paper. The research found that ReliefF can be used to select fewer attributes, with no degradation of subsequent classifier performance, for the data balanced by the proposed under-sampling method.

Keywords: class imbalance, clustering, over-sampling, ReliefF, SMOTE, under-sampling

1. 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]. Actually, the cost in mis-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. Furthermore, 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.

Feature selection is the process of selecting a subset of relevant features for use in model construction. Feature selection is also useful as part of the data analysis process. However, the class imbalance problem can affect the feature selection process. Our research found that very little work is done in this area, to find and address the issues of class imbalance in feature selection. In the work of Al-Shahib et al. [4], the author used random undersampling to balance their data for SVM classification. The author also used feature selection on balanced data. They found that SVM performed well on the balanced data, but the paper does not provide any analysis of the effect of the class imbalance problem in feature subset selection.

Sampling strategies have been used to overcome the class imbalance problem by either eliminating some data from the majority class (undersampling) or adding some artificially generated or duplicated data to the minority class (oversampling) [5]. Over-sampling techniques [6] 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 [6]. 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 [7]. Chawla [6] 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.

In summary, over-sampling may cause longer training time and overfitting [8]. 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 [9] 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 [8]. Many researchers have proposed more advanced ways of under-sampling the majority class data [8, 10, 11].
In the rest of this paper we present a semi supervised cluster based under-sampling technique to balance cardiovascular data for classification and feature selection.

2. CLUSTER BASED UNDER-SAMPLING

Down-sizing the majority class results in a loss of information that may result in overly general rules. In order to overcome this drawback of the under-sampling approach Yen and Lee [8] proposed cluster-based under-sampling. Their approach is to first cluster all the training samples into \( K \) clusters 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, 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.

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. The \( M \) number of majority class samples are randomly chosen from each cluster. In the \( i^{th} \) cluster \((1 \leq i \leq K)\) the \( \text{Size}_{MA}^i \) will be

\[
\text{Size}_{MA}^i = (m \times \text{Size}_{Mi}) \times \frac{\sum_{i=1}^{K} \text{Size}_{MA}^i / \text{Size}_{Mi}}{\sum_{i=1}^{K} \text{Size}_{MA}^i / \text{Size}_{Mi}}
\]

This approach may be suitable for datasets where class labels are confidently defined and truly reflect the properties of the labeled class record. But in some cases, especially for medical datasets, there is no
guarantee that the class labels are truly reflecting the actual characteristics of that record [11].

Our approach to under-sampling is different to the approach of Yen and Lee [8]. The aim is not to derive a majority and minority class ratio of 1:1; but just to reduce the gap between the numbers of majority class samples to the numbers of minority class samples. As shown in the figure 1, 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 make $K$ ($K > 1$) subsets, where each cluster is considered to be one subset of the majority class. All the subsets of the majority class are separately combined with the minority class samples to make $K$ different training data sets. (The value for $K$ is dependent on the data domain, in our implementation the final $K$ value used was 3. All the combined datasets are classified with decision tree [12] and Fuzzy Unordered Rule Induction Algorithm [13]. 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 presented in table 1, using k-means clustering and classified using decision tree. The experimental outcomes are discussed in the result section.

3. FEATURE SELECTION FRAMEWORK FOR UNBALANCED DATA

A framework of feature selection for unbalanced clinical data sets is proposed in this research. The framework is based on k-means clustering and instance based feature selection algorithm. First the cluster based under-sampling is used to balance the datasets; later the ReliefF algorithm is used for feature ranking. The steps of the feature selection process are given in Figure 2.

---

**Figure 2. Feature Selection Framework for Unbalanced Dataset**

- Separate the datasets into two sets. One for the majority class sample and another for all the minority class samples.
- Cluster the majority class sample into $K$ clusters. Combine all the clusters separately with the cluster of minority samples.
- Classify all the clusters with a standard classifier and select the dataset having highest classification accuracy.
- Apply ReliefF feature selection algorithm and rank the features.
- Select the feature subsets based on ranking.

---
4. EXPERIMENTS AND ALGORITHMS

We have used two cardiovascular datasets from Hull and Dundee clinical sites. k-means [14] clustering is used for under-sampling the minority class and ReliefF is used for feature selection. For choosing the best subset, we have used j48 decision tree [14] and Fuzzy Unordered Rule Induction Algorithm (FURIA) [15] as classifiers.

4.1 Overview of FURIA

Fuzzy Unordered Rule Induction Algorithm (FURIA) is a fuzzy rule-based classification method. Fuzzy rules are obtained through replacing intervals by fuzzy intervals with trapezoidal membership functions [15]:

$$\mu_{r^F}(x) = \prod_{i=1...k} t_i^F(x_i)$$  \hspace{1cm} (2)

For fuzzification of a single antecedent, only relevant training data \(D^l_T\) is considered and data are partitioned into two subsets and rule purity is used to measure the quality of the fuzzification [15]:

$$D^l_T = \{x = (x_1..x_k) \in D^l_T | t_i^F(x_j) > 0 \text{ for all } j \neq i \} \subseteq D_T$$ \hspace{1cm} (3)

The fuzzy rules \(r_1^{(i)}, r_k^{(j)}\) are learnt for the class \(\lambda_j\), the support of this class is defined by [15]:

$$s_j(x) = \sum_{i=1...k} \mu_{r_i^{(j)}(x)} \cdot CF(r_i^{(j)})$$ \hspace{1cm} (4)

where, the certainty factor of the rule is defined as

$$CF(r_i^{(j)}) = \frac{\frac{D^j_T}{D^l_T} + \sum_{l \neq j} \mu_{r_l^{(j)}(x)}}{2 + \sum_{l \neq j} \mu_{r_l^{(j)}(x)}}$$ \hspace{1cm} (5)

4.2 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. [14]. Decision trees models are commonly used in data mining to examine the data, and to induce the tree and its rules that will be used to make predictions.
Ross Quinlan introduced a decision tree algorithm (known as Iterative Dichotomiser (ID 3) in 1979. C4.5, as a successor of ID3, is the most widely-used decision tree algorithm [16]. 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 (the root node is always the most significant attribute to the class). Typically the goal is to find the optimal decision tree by minimizing the generalization error. The algorithms introduced by Quinlan [17] have proved to be an effective and popular method for finding a decision tree to express information contained implicitly in a data set. WEKA [18] makes use of an implementation of C4.5 algorithm called J48 which has been used for all of our experiments.

4.3 Relief: An Instance Base Approach to Feature selection

Kira and Rendell [19] introduced an algorithm called Relief that uses instance based learning to assign a relevance weight to each feature. Relief is a simple yet efficient procedure to estimate the quality of attributes. The key idea of the Relief is to estimate the quality of attributes according to how well their values distinguish between instances that are near to each other. Given a randomly selected instance $R_i$ from class $L$, Relief searches for $k$ of its nearest neighbours from the same class called nearest hits $H$, and also $k$ nearest neighbours from each of the different classes, called nearest misses $M$. It then updates the quality estimation $W_i$ for the $i^{th}$ attribute based on their values for $R_i$, $H$, and $M$. If instance $R_i$ and those in $H$ have different values on the $i^{th}$ attribute, then the quality estimation $W_i$ is decreased. On the other hand, if instance $R_i$ and those in $M$ have different values on the $i^{th}$ attribute, then $W_i$ is increased.

$$W[A] := W[A] - \frac{\text{diff}(A, R_i, H)}{m} + \frac{\text{diff}(A, R_i, M)}{m}$$  \hspace{1cm} (6)

Where $A$ is the current attribute; $W[A]$ is the weight of the currently considered attribute; $R_i$ is the $i^{th}$ sample; $H$ is the “hit”; $M$ is the “miss”; $\text{diff}()$ is the probability function; and $m$ is number of the neighbours.

The Relief algorithm is limited to classification problems with two classes. The ReliefF algorithm [20] is an extension of Relief algorithm that can deal with multi-class problems. ReliefF is a simple yet efficient procedure to estimate the quality of attributes in problems with strong dependencies between attributes. In practice, ReliefF is usually applied in data preprocessing as a feature subset selection method.

There are many other extensions of the Relief and ReliefF proposed by many researchers. Details about the algorithms and their application can be found in work of Robnik et al [20].
Algorithm 1. ReliefF

\begin{itemize}
  \item **Input:** For each training instance a vector of attribute values and the class value.
  \item **Output:** The vector $W$ of estimations of the qualities of attributes
  \item **Step 1:** Set all weights $W[A] := 0.0$;
  \item **Step 2:** \textbf{for} $i := 1$ to $m$ \textbf{do begin}
    \begin{itemize}
      \item randomly select an instance $R_i$; find $k$ nearest hits $H_j$;
    \end{itemize}
  \item **Step 3:** \textbf{for} each class $C \in \text{class}(R_i)$ \textbf{do}
    \begin{itemize}
      \item from class $C$ find $k$ nearest misses $M_j(C)$;
    \end{itemize}
  \item **Step 4:** \textbf{for} $A := 1$ to $a$ \textbf{do}
    \begin{itemize}
      \item $W[A] := W[A] - \sum_{j=1}^k \frac{diff(A,R_i,H_j)}{m-k} + \sum_{c=\text{class}(R_i)} \frac{p_{C}}{1-\text{p}_{\text{class}(R_i)}} \sum_{j=1}^k \frac{diff(A,R_i,M_j(C))}{m-k}$;
    \end{itemize}
  \item **Step 5:** \textbf{end;}
\end{itemize}

### 4.4 Cardiovascular Data

We have used two datasets from Hull and Dundee clinical sites. 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, 23 matched attributes are left.

Missing values: After combining the data and removing redundant attributes we found that out of 23 attributes 18 attributes have a missing value frequency from 1% to 30% and out of 832 records 613 records have 4% to 56% missing values in their attributes.

From these two data sets, we prepared a combined dataset having 23 attributes with 823 records. Out of 823 records 605 records have missing values and 218 records do not have any missing values. Among all the records 120 patients are dead (High risk) and 703 patients are alive (Low risk). For this experiment according to clinical risk prediction model (CM1) [21], patients with status “Alive” are consider to be “Low Risk” and patients with status “Dead” are consider to be “High Risk”.

### 4.5 Classifier 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.

<table>
<thead>
<tr>
<th>Predicted classes</th>
<th>Expected or Actual classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High risk</td>
</tr>
<tr>
<td>High risk</td>
<td>TP</td>
</tr>
<tr>
<td>Low risk</td>
<td>FP</td>
</tr>
</tbody>
</table>

Figure 3: Confusion Matrix

The accuracy of a classifier is calculated by:

\[
ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}
\]

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:

\[
Sen = \frac{TP}{TP + FN} \tag{8}
\]

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

\[
Spec = \frac{TN}{TN + FP} \tag{9}
\]

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).

5. 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 2 and 3.
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 make another sample datasets called “D6”.

We also made two more datasets using the under-sampling by clustering method proposed by Yen and Lee [8] 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>
<thead>
<tr>
<th>Data</th>
<th>Ratio</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1:</td>
<td>$2 : 1$</td>
<td>Data consist of all the minority class samples (“dead”) and one cluster of majority class records out of three clusters made by K-Mean.</td>
</tr>
<tr>
<td>D2:</td>
<td>$2.4 : 1$</td>
<td>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).</td>
</tr>
<tr>
<td>D3:</td>
<td>$3 : 1$</td>
<td>Data consist of combination of all the minority class samples with randomly (random cut 1) selected samples from majority class sample.</td>
</tr>
<tr>
<td>D4:</td>
<td>$3 : 1$</td>
<td>Data consist of combination of all the minority class samples with randomly (random cut2) selected samples from majority class sample.</td>
</tr>
<tr>
<td>D5:</td>
<td>$6 : 1$</td>
<td>Original data with full samples.</td>
</tr>
<tr>
<td>D6:</td>
<td>$1.8 : 1$</td>
<td>Majority samples of the data set D2 are clustered in to 3 cluster and each clusters are combined with the minority samples.</td>
</tr>
<tr>
<td>K3M1Yen</td>
<td>$1: 1$</td>
<td>Majority and minority ratio 1:1 (M=1) using Yen and Lee (2009)</td>
</tr>
<tr>
<td>K3M2Yen</td>
<td>$2: 1$</td>
<td>Majority and minority ratio 2:1 (M=2) using Yen and Lee (2009)</td>
</tr>
</tbody>
</table>
Table 2. Classification Outcome of FURIA

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

Table 3. Classification Outcome of Decision Tree

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

From the table 2 and 3 we can see that the original unbalanced 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 64.2% with the decision tree and 67.5% 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 [8] 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 with its accuracy and sensitivity values.
It is observed from the experiments that the majority and minority ratio is not only the issue in building a good prediction model. There is also a need of good training simple, which should display the true properties of the class label assigned to them. As we discussed earlier, that for some data records, the class labels of clinical dataset do not accurately reflect the true properties 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, there is a big difference in the classification accuracy, sensitivity and specificity of D1 and D6, as can be noticed in tables 2 and 3. The dataset “K3MIYen” prepared by the method proposed by Yen and Lee [8] has 1:1 ratio but still has poorer classification outcome than other datasets.

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 display the highest accuracy of all the datasets.

An experiment was made based on the proposed feature selection framework. The aim of the experiment was to observe the effect of the class imbalance problem on feature selection with ReliefF. ReliefF feature selection was applied to rank the unbalance data (D5) and data the balanced by the proposed method (D6), see the figure 6.

![Figure 6. Attribute Ranking by ReliefF for the Balanced and Unbalanced Data](image)

The datasets were later classified by the decision tree classifier. From the experiment it is observed that, for the balanced data out of 23 attributes only 9 attribute were needed to keep the highest sensitivity of the data for decision tree classification which is 40% of the total attributes. On the other hand, for the unbalanced data out of 23 attributes, a minimum of 14 attributes were needed to keep the highest sensitivity of the data which is 63% of the total attributes. The finding (see figure 7) shows that ReliefF can perform better with the data balanced by the proposed under-sampling method and can select a fewer number of attributes with no degradation in classifier performance.
CONCLUSION

Most medical datasets are not balanced in their class labels. In some cases it has been noticed that class labels do not represent a true property of the record. 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 preparing unbalanced datasets where the given class labels are not always appropriate and fail to truly reflect the underlying characteristics of the patient record.

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 chose good quality training set data for building classification models.

The research found that class imbalance not only affects the classification process but also has effect on feature selection process. This is because most of the feature selection methods use the class label of the dataset to select the attribute subset.

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.
ACKNOWLEDGMENT

The authors gratefully acknowledge SEED Software in the Department of Computer Science of The University of Hull, UK, for funding this research project.

REFERENCES


