Data Mining and Knowledge Discovery (2021) 35:1654–1687
https://doi.org/10.1007/s10618-021-00766-4

An overlap sensitive neural network for class imbalanced data

Shaukat Ali Shahee[1] · Usha Ananthakumar[1]

Received: 9 October 2019 / Accepted: 6 May 2021 / Published online: 18 May 2021

Abstract

Class imbalance is one of the well-known challenges in machine learning. Class imbalance occurs when one class dominates the other class in terms of the number of observations. Due to this imbalance, conventional classifiers fail to classify the minority class correctly. The challenges become even more severe when class overlap occurs in imbalanced data. Though literature is available to sequentially deal with class imbalance and class overlap, these methods are quite complex and not so efficient. In this paper, we propose an overlap-sensitive artificial neural network that can handle the problem of class overlapping and class imbalance simultaneously, along with noisy and outlier observations. The strength of this method lies in identifying the overlapping observations rather than the region and in not using multiple classifiers unlike the other existing methods. The key idea of the proposed method is in weighing the observations based on its location in the feature space before training the neural network. The performance of the proposed method is evaluated on 12 simulated data sets and 23 real-life data sets and compared with other well known methods.The results clearly indicate the strength and ability of the proposed method for a wide variety of imbalance ratio and levels of overlapping. Also, it is shown that the proposed method is statistically superior to the other methods in terms of different performance measures.

Keywords Classification · Imbalanced data · Class overlapping · Artificial neural network · Outliers · Misclassification costs

Responsible editor: Pierre Baldi.

✉ Usha Ananthakumar
usha@som.iitb.ac.in

Shaukat Ali Shahee
shaukatali.shahee@iitb.ac.in

[1] SJM School of Management, Indian Institute of Technology Bombay, Bombay 400076, India

1 Introduction

Artificial Neural-Network (ANN) is one of the widely used machine learning techniques to address challenging real life situations. The architecture of ANN is based on the connection of nodes/ artificial neurons in layer-wise structure. The neurons in each layer receive signal from the previous layer that gets processed using some non-linear function and transferred to the subsequent layer neurons through the edges. One common approach of ANN is to learn the weights of the network iteratively by minimizing the sum of squared error (SSE) between the output of the network and the actual target. In this way, ANN approximates the function that fits well on the data under consideration. Though the computation of the parameter weights works well for balanced data, the process becomes biased when imbalance is present in the data set. A data set is called imbalanced if the number of observations of one class (majority class) exceeds the number of observations in the other class (minority class). Imbalanced data has wide real world applications including customer churn prediction (Burez and Van den Poel 2009), financial distress prediction (Cleofas-Sánchez et al. 2016), gene regulatory network reconstruction (Ceci et al. 2015) and information retrieval and filtering (Piras and Giacinto 2012). In most of the applications, the minority class is usually of prime interest. Standard classifiers generally result in higher misclassifications of the minority class due to its bias towards the majority class resulting in sub-optimal solution (López et al. 2013; Thanathamathee and Lursinsap 2013).

In literature, many solutions have been proposed for improving the classification accuracy of the minority class without severely jeopardizing the classification accuracy of the majority class. These solutions have been categorized as sampling methods (Barua et al. 2012; Chawla et al. 2002; Han et al. 2005; Shahee and Ananthakumar 2018a, b), cost-sensitive learning methods (Sun et al. 2007), one class learning (Chawla et al. 2004), and feature selection (Alibeigi et al. 2012; Yin et al. 2013; Shahee and Ananthakumar 2019).

Sampling method is a preprocessing technique that provides a balanced class distribution so that the classifiers behave in a similar manner as traditional classifiers (Batista et al. 2004; Estabrooks et al. 2004). Cost sensitive learning considers different costs of misclassification of each example using cost matrix for handling class imbalance (Elkan 2001; Ting 2002). It assigns higher cost to misclassification of minority class observations compared to majority class observations. In one-class learning, one-class SVM is trained with only the target class (Tax and Duin 2004). Feature selection in case of imbalanced domain is to select the appropriate features for better classification of minority class. However, none of the above mentioned approaches provides uniformly superior performance when the classes are imbalanced. Sampling based method is considered to be the simplest technique for handling class imbalance because it provides balanced class distributions, without modifying the standard classification algorithm (Barua et al. 2012; Chawla et al. 2002; Han et al. 2005; Shahee and Ananthakumar 2018a, b).

Apart from class imbalance, certain data intrinsic characteristics like overlapping between the classes, lack of density and information in the training data, impact of noisy observations/outliers, presence of small disjuncts also worsen the performance of the classifiers (Alshomrani et al. 2015; Japkowicz and Stephen 2002; Jo and Japkowicz

2004; Prati et al. 2004; López et al. 2013). In many real-world applications, data exhibits class imbalance problem along with some data intrinsic characteristics (Tang et al. 2010).

In this paper, we propose a new method for binary class data that can handle class imbalance and class overlapping simultaneously along with noisy / outlier observations. Introduction of class overlap along with class imbalance adds additional challenges to the classification task. Batista et al. (2005) conducted an experiment on synthetic data and found that performance degradation of the classifier is not solely due to class imbalance but is also related to the degree of overlap between the classes. In literature, various methods have been proposed but most of the methods deal with class imbalance and class overlapping sequentially. The proposed method addresses the problem of class overlap in class imbalance using a very different approach of assigning different weights to the observations before training the ANN.

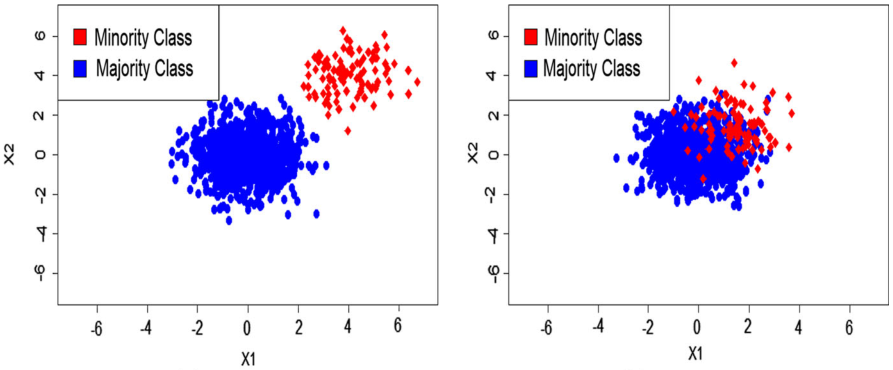
Some of the major contributions of this paper are summarized as follows:

- We propose an overlap sensitive neural network where the loss function of the network varies with respect to the weights of the observations.
- The weights of the observations are computed by taking into account the location of observations in the feature space.
- The presence of noisy/outlier observations and the imbalance in class distributions are also given due consideration while computing the weights.
- To demonstrate the effectiveness of the proposed method, we evaluate its performance on 12 simulated data with different scenarios and 23 real-world data sets and compare with the other methods. The results clearly show that the proposed method outperforms the other methods.

The organisation of the paper is as follows. Related work is presented in Sect. 2 followed by the details of the proposed method in Sect. 3. Evaluation details and parameter settings details are given in Sect. 4. Analysis on simulated data sets and real-life data sets are presented in Sects. 5 and 6 respectively. Section 7 presents discussion on the proposed method and finally conclusion is presented in Sect. 8.

2 Related work

Since our focus is to pursue the challenges in handling data with class overlap in the presence of class imbalance for neural network classifier, we shall review the relevant literature. As shown in Fig. 1, overlapping data enclose an ambiguous region in the feature space where the prior probability of the classes are roughly equal (Das et al. 2013). Generally, overlapping is caused by lack of features to differentiate the classes. When the data set has overlapping regions, traditional classifiers are not able to find a feasible solution for classification (Xiong et al. 2010). In other words, overlapping nature of data sets makes it difficult to identify a class boundary that can perfectly separate the classes (Das et al. 2013). In comparison to solving the problems of imbalanced and overlapping classes independently, finding a solution for classes with both overlap and imbalance is more difficult.



(a) Group A **(b)** Group B

Fig. 1 Left data without class-overlap, Right data with class-overlap

In general, a kernel function is used to solve the class overlap problems (Das et al. 2013; Qu et al. 2011). A kernel function transforms the data from lower dimensional space to higher dimensional space, maximising the chance of identifying a linear separator in higher dimension feature space. However, class overlap can still exist in the higher dimensional feature space (Lee and Kim 2018).

Xiong et al. (2010) introduced an approach in which data located in the overlapping region is treated differently from data in the non overlapping region. Support Vector Data Description (SVDD) is used to find overlapping regions, followed by three overlapping-class modeling schemes, namely discarding, merging and separating. In discarding scheme, models are trained based on data that lie in the non-overlapping region, while data in the overlapping is discarded. In case of merging scheme, data in the overlapping region are given a new class label "overlapping" and two models are trained. The first model is trained by considering overlapping region as a new class and the second model is trained only on data in the overlapping region. On test data sets, if the first model classifies as overlapping class, then the second model is used to determine the original class. In the case of separating scheme, though two models are trained, the first model is trained only on the overlapping region and the second model is trained only on the non-overlapping region. Findings of the paper suggest that the separating scheme is the best among the three schemes. However, in all the three cases, models are trained multiple times leading to decrease in computational efficiency.

Tang et al. (2010) used a probabilistic neural network (PNN) to divide the feature space into overlapping and non-overlapping regions. For deciding the overlapping region, two considerations are taken into account. First, the overlapping region should be large enough to accommodate most of the potentially misclassified observations to ensure that the classification of observations lying in the non-overlapping region is highly accurate. Second, the overlapping region should not be too wide to include too many patterns. For classification of the test set, if an observation falls in the non-overlapping region, then it is classified based on the highest posterior probability. If the test observation falls in the overlapping region, a new method is suggested for classification.

Das et al. (2013) proposed ClusBUS (Clustering-based undersampling technique) for handling class imbalance along with class overlap. Their method identifies different clusters present in the data set without considering the class. After that, it computes the ratio (r) of the number of minority class observations and the size of the cluster. Based on the computed r value and empirically determined threshold τ, it removes all of the majority class observations and retains the minority class observations. In each cluster, vacuum gets created around the minority class observations thus helping classifiers in learning the decision boundary efficiently. However, removal of the majority class could result in loss of information.

Tang and Gao (2007) proposed a multi-model classifier (DR-SVM) which combines SVM and kNN under rough set technique. KNN is used to identify the boundary data and the remaining data in each class is called positive region pattern. Two SVM classifiers are trained in DR-SVM and a pair of separating hyperplanes are obtained. The test set falling in the overlapping region is classified by KNN algorithm.

Lee and Kim (2018) proposed an overlap-sensitive margin (OSM) classifier that separates the data space into soft and hard-overlap regions using the modified fuzzy support vectors machine algorithm. Test set observations falling in soft-overlap region are classified based on decision boundary of the OSM classifier and those observations falling in hard-overlap region are classified using KNN algorithm with $k = 1$. The key point of this method is that each observation has different cost of misclassification. However, this method results in significant decrease in the classification of majority class observations when data is highly imbalanced. Furthermore, this method does not take into account the presence of outliers or noisy examples in the data set.

Lin et al. (2017) proposed a focal loss function for handling class imbalance. The loss function modulates the cross entropy loss function in such a way that it focuses the learning on hard to learn examples and down-weigh the contribution of numerous easy examples. However, the loss function is not balanced between the classes. Cui et al. (2019) modified different loss functions, especially the focal loss function to make it class-balanced loss function. The class-balanced loss is a re-weighting loss function where weight is inversely proportional to the effective number of samples. The data overlap is considered in quantifying the effective number of samples. The proposed class-balanced loss achieves significant performance gains on long-tailed data sets.

From the above literature, we observe that every method has its own limitation and further, most of the studies have considered the problems of class imbalance and overlapping sequentially. In this paper, we propose an overlap sensitive neural network that takes into account class overlap and class imbalance simultaneously. Our algorithm not only increases the performance of the classifiers on the minority class but it also ensures that the performance on the majority class is not compromised. In addition, the proposed method also considers the presence of noisy examples or outliers in the data set.

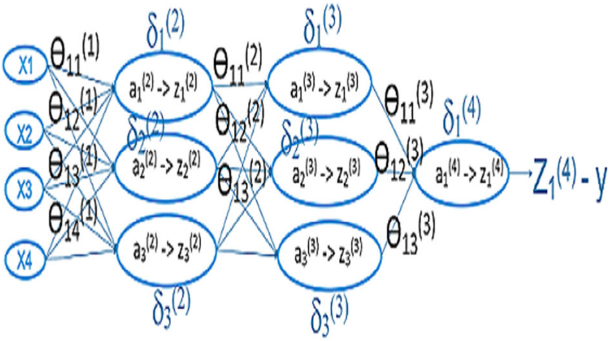


Fig. 2 Feed-forward-neural-network-architecture

3 The proposed method

From the literature, it can be seen that one way of handling class imbalance is to assign higher weight to the minority class compared to the majority class. Motivated from this notion, rather than entire class being assigned a weight, we assign different weights to the observations in our proposed method to handle class imbalance and class overlap simultaneously. The assigned weight to each observation acts as cost of misclassification with respect to that observation. This section discusses our proposed method by describing each of the components in the subsequent subsections.

3.1 Neural network training

In ANN, input is fed via input layer followed by Sigmoid activation function applied at each neuron that finally gets transferred to the output layer. This is also called feedforward network as shown in Fig. 2.

In forward propagation, the output of the jth hidden unit in the first hidden layer is obtained as

$$a_j^{(2)} = \sum_{i=1}^{n} \theta_{ji}^{(1)} x_i + \theta_{j0}^{(1)} \tag{1}$$

where the superscript indicates the layer it belongs to. Here, $\theta_{ji}^{(1)}$ denotes the weight parameter connecting ith input and jth neuron and $\theta_{j0}^{(1)}$ denotes the bias for hidden unit j. Now a_j is transformed using the sigmoid activation function $h(.)$ (McClelland et al. 1988) resulting in

$$z_j = h(a_j) \tag{2}$$

where

$$h(x) = 1/(1 + exp(-x)) \tag{3}$$

Similar process is followed in the next layer with inputs as z_j to finally obtain the output of the network.

For determining the weights θ, ANN in general considers the cross-entropy error function for a classification problem as it leads to faster training and improved generalization (Simard et al. 2003). In the current study, each observation is assigned different importance depending on its location in the feature space and accordingly, the error function considered is

$$E(\theta) = -\sum_{n=1}^{N}\{y_n \log z_n + (1-y_n)\log(1-z_n)\} * w_n \tag{4}$$

where z_n denotes the output of the network for the input x_n, y_n is the actual class label and w_n denotes the weight of the nth observation. Here, the gradient of $E(\theta)$ w.r.t θ is computed using backpropagation iteratively.

As $E(\theta)$ depends on the weight θ_{ji} through the summed input a_j, we apply the chain rule for partial derivative given by

$$\frac{\partial E_n(\theta)}{\partial \theta_{ji}} = \frac{\partial E_n(\theta)}{\partial a_j}\frac{\partial a_j}{\partial \theta_{ji}} w_n \tag{5}$$

Let us denote

$$\delta_j \equiv \frac{\partial E_n(\theta)}{\partial a_j} \tag{6}$$

Hence, Eq. (5) becomes

$$\frac{\partial E_n(\theta)}{\partial \theta_{ji}} = \delta_j \frac{\partial a_j}{\partial \theta_{ji}} w_n \tag{7}$$

where $\frac{\partial a_j}{\partial \theta_{ji}} = x_j$ for layer 1 and z_j for other layers.

This implies that the required derivatives are computed by multiplying the weight w_n of each observation, by the product of the z value at the input end of the weight and the δ value of the following layer. To obtain the derivatives in (7), δ's are computed next.

For the output neuron,

$$\delta_k = z_k - y_k \tag{8}$$

where z_k is the predicted value of the kth observation and y_k is the true class label.

Now, the δ values for hidden layer neurons are obtained by propagating the δ's recursively from the next higher layer using

$$\delta_j = h^{'}(a_j)\sum_k \theta_{kj}\delta_k \tag{9}$$

Substituting these δ's in Eq. (7), the required derivatives are computed.

3.2 Computation of observation weights

The main motivation of assigning weights to the observations is to increase the performance of the classifier on the minority class without losing the performance on the majority class. The computation of weights of observations takes into account class overlapping, class imbalance and noisy/outlier observations. For this, data is divided into three parts, namely safe zone, overlapping and outlier or noisy observations. An observation is called a safe zone observation if it is surrounded by the same class observations. Overlapping observations are those that have few other class observations in its neighbourhood. When an observation is surrounded by all the observations of the other class, it is referred to as an outlier.

3.2.1 Class overlapping

For handling overlapping between the classes, we assign weights to the observations with respect to the level of overlap. Higher weights are assigned to the observations that are less overlapping and lower weights to the observations that are of high overlapping in nature. For incorporating overlapping between the classes, propensity score of each observation is defined by using $K - NN$ algorithm. The value of K is set to 5 like other well known methods in class imbalance domain (Chawla et al. 2002; He et al. 2008).

$$P = NN/5 \tag{10}$$

where P is the propensity score and NN is the number of examples from the same class. $P = 0$ means the observation is located inside the other class, in other words, it is an outlier observation. $P = 1$ refers to safe zone observation, surrounded by observations of the same class. An observation with $0 < P < 1$ is referred to as an overlapping observation. Hence, overlapping between classes is incorporated by computation of the propensity score.

To accommodate different levels of overlapping of outliers in our method, we consider a parameter C in the range [0.0, 0.20]. This range is chosen so that the weight assigned to an outlier cannot exceed that assigned to an overlapping observation. Thus, each observation is assigned a propensity score including outliers being assigned a score C.

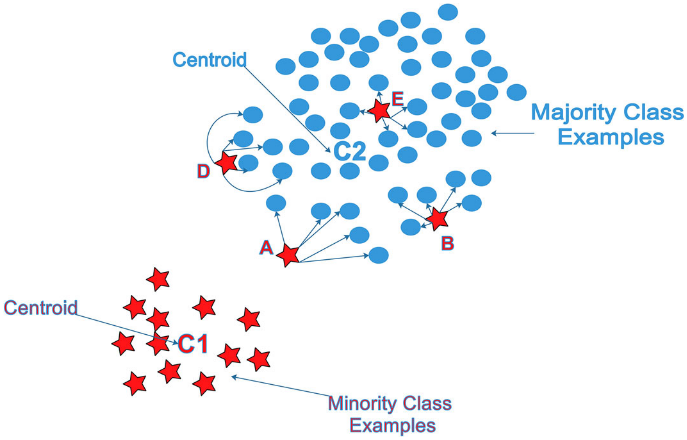


Fig. 3 Different types of outliers: A is least overlapping, followed by B & D and then by E

3.2.2 Handling class imbalance

After dealing with class overlapping by using the propensity score, we now have to deal with class imbalance. One of the techniques for handling class imbalance is to preprocess the data that diminishes the effect of class imbalance by either increasing the minority class observations or by decreasing the majority class observations. In this study, we diminish the effect of class imbalance by making the total weight of the minority class equal to the total weight of the majority class. To do this, we define the Overlapping imbalance ratio (OIR), calculated as the number of overlapping observations in the minority class ($nn1$) divided by the number of overlapping observations in the majority class ($nn0$).

$$OIR = nn1/nn0 \tag{11}$$

Then we multiply the weights of overlapping majority class observations by OIR. Further, to make the total weights of majority class and minority class equal, the minority class examples are multiplied by a factor WGT where WGT is defined as

$$WGT = \text{sum of majority class weights}/\text{Sum of minority class weights} \tag{12}$$

In this way, the sum of the weights of both the classes get balanced and this also ensures that the weights of observations in safe zone are higher than the weights of observations in the overlapping region which in turn are higher than the outliers.

3.2.3 Outlier adjustment

While handling outliers, all such observations were assigned the same score C as indicated in Sect. 3.2.1. Further, weight adjustment while handling class imbalance changes the weights of the outliers uniformly. It is possible that these outliers are not of the same kind as some may be easy to classify than other outliers. This points out the need to weigh the outliers based on their level of overlapping. To illustrate this aspect, Fig. 3 shows four outliers A, B, D and E of the minority class. Here, A is close to the minority class and is relatively easily classifiable compared to B and D and hence the level of overlapping is the least for A, followed by B and D and then by E. Thus, the assigned weights should be in decreasing order of the extent of overlapping. To accommodate this, we compute the distance of the outliers from the centroid of its class using Eq. (13).

$$d = (||X_c - X_i||_2)^{1/2} \tag{13}$$

where d is the distance between the centroid X_c and the outlier X_i. Now the outliers are weighed with respect to the distance and accordingly, we choose a monotonic decreasing function of distance given by (14).

$$w_d = 2/(1 + exp(d)) \tag{14}$$

The weights of the outliers are adjusted by multiplying with this quantity w_d. By this, the outlier that is located far from its centroid gets assigned less weight compared to the outlier that is closer to its centriod. This would result in reduction of weights in outliers and the difference is distributed proportionately among other observations. The proposed method is summarized in *Algorithm 1*.

Algorithm 1 Computation of weight vector of the classes

Input: Training dataset: $S = \{X_i, y_i\}, i = 1, ..., N$; $X_i \in R^n$ and $y_i \in \{0, 1\}$ Positive class (*class1*): $S^+ = \{X_i^+, y_i^+\}, i = 1, .., m^+$; Negative class (*class0*): $S^- = \{X_i^-, y_i^-\}, i = 1, ..., m^-$; $S = S^+ \cup S^-$; $N = m^+ + m^-$

Output: Weight vector of minority class V_1 and weight vector of majority class V_0

```
function WeightVectorComputation($S^+, S^-$)
Compute propensity score of $S^+$ and $S^-$ observations – return $ClassPropPlus$ and $ClassPropNeg$ respectively.
// Initialize $C$ with one of the value $C \in \{0.20, 0.15, 0.10, 0.05, 0.00\}$ to all the outliers of $S^+$
$ClassPropPlusNew$ = []
for $i = 1$ to $length(ClassPropPlus)$ do
    if $ClassPropPlus[i] == 0$ then
        $ClassPropPlusNew[i] = C$
    else
        $ClassPropPlusNew[i] = ClassPropPlus[i]$
    end if
end for
$ClassNeg \leftarrow OIR * ClassPropNeg$
$WGT \leftarrow sum(ClassNag)/sum(ClassPropPlusNew)$
$ClassPosWgt \leftarrow ClassPropPlusNew * WGT$
$V1 \leftarrow OutlierAdjustment(ClassPropPlus, ClassPosWgt)$
$V_0 \leftarrow ClassNag$
return $V_0, V_1$
end function

function OutlierAdjustment($ClassPropPlus, ClassPosWgt$)
ClassPropPlus3 = []
for $i = 1$ to $length(ClassPropPlus)$ do
    if $ClassPropPlus[i] == 0$ then
        $d \leftarrow (||X_c - X_i||_2)^{1/2}$ // $x_c$ is the centroid of class1 and $X_i$ is an outlier of class1.
        $dist \leftarrow 2/(1 + exp(d))$
        $ClassPropPlus3[i] = ClassPosWgt[i] * dist$
    else
        $ClassPropPlus3[i] = ClassPosWgt[i]$
    end if
end for
diff = sum(ClassPosWgt) - sum(ClassPropPlus3)
SumTemp = Sum of the weights of non-outliers of ClassPropPlus3
$ratio = diff/SumTemp$
AdjWgt = NULL
for $i = 1$ to $length(ClassPropPlus)$ do
    if $ClassPropPlus[i] \neq 0$ then
        $AdjWgt[i] = ClassPropPlus3[i] * ratio$
    else
        $AdjWgt[i] = 0$
    end if
end for
$V_1 \leftarrow ClassPropPlus3 + AdjWgt$
return $V_1$
end function
```

3.3 Computation complexity

We analyze the computational complexity of computing the propensity score. Propensity score of an observation is based on the number of observations of the same class in its K-NN. Let the data set have N examples in the n-dimensional feature space. We know that the time complexity of computation of distance between two points in R^n is $\mathcal{O}(n)$. Since $(N-1)$ distances are computed for each example, the corresponding time complexity is $\mathcal{O}(N-1)n$. These distances are sorted using Radix sort. The time complexity of sorting the $(N-1)$ distances using Radix sort is $\mathcal{O}(N-1+b)\log_b D)$ and selecting the first k observations is $\mathcal{O}(1)$. Hence, total time taken for one example is $\mathcal{O}(N-1)n+\mathcal{O}(N-1+b)log_b D$. As we have total N observations, the total time complexity is $\mathcal{O}((N-1)n+(N-1+b)\log_b D)*N$ which is approximately $\mathcal{O}(N^2n)$. The present study uses this propensity score, though it can become expensive for large data sets having large number of features. Further, if categorical variables are present in the dataset, one hot encoding is generally used for its representation. This results in increasing the number of features which further increases the time complexity. Appropriate distance measure needs to be used for calculation of Propensity score to handle non continuous variables which we intend to study later.

4 Evaluation metrics and parameter settings

This section discusses the evaluation metrics used to evaluate the performance of the proposed method. The parameter settings used in the network are also presented.

4.1 Evaluation metrics

The proposed method is evaluated on the basis of evaluation metrics that are commonly used in the literature (He et al. 2008; He and Garcia 2008; Tharwat 2018), being derived from the confusion matrix Table 1. In this confusion matrix, rows denote the number of true class examples and the columns denote the number of examples classified by the classifier.

Some of the well known metrics used for imbalanced data sets are *precision*, *recall*, *F-measure* and *G-mean* (He and Garcia 2008). These metrics are defined as

$$Precision = \frac{TP}{TP+FP} \tag{15}$$

$$Recall = \frac{TP}{TP+FN} \tag{16}$$

$$F-Measure = \frac{(1+\beta^2)Recall * Precision}{\beta^2 * (Recall + Precision)} \tag{17}$$

Here β is a non-negative parameter that controls the influence of precision and recall. With $\beta = 0$, F-Measure is same as recall and when $\beta \to \infty$, it tends to Precision. In

Table 1 Confusion matrix

Classifier Output			
True Class		P	N
	p	TP	FN
	n	FP	TN

this study, we set $\beta = 1$, giving equal importance to precision and recall.

$$G - Mean = \sqrt{\frac{TP}{TP + FN} \frac{TN}{TN + FP}} \tag{18}$$

G-Mean measures the performance by combining the *Recall* of positive and negative classes (Guo and Viktor 2004b).

Another widely used graphical based representation for imbalanced domain is Receiving Operating Characteristic (ROC) curve (Provost et al. 1997). This representation of the performance of the classifier plots *TP rates* on the Y-axis and *FP rates* on the X-axis. The TP rates and FP rates are defined as

$$\text{TP rate} = \frac{TP}{TP + FN} \tag{19}$$

$$\text{FP rate} = \frac{FP}{FP + TN} \tag{20}$$

A quantitative representation of ROC curve is the area under this curve and is called AUC (Bradley 1997; Huang and Ling 2005).

4.2 Parameter settings

In this study, the neural network has one hidden layer, and the number of neurons it contains is (No. of features + classes)/2, as considered in the literature (Guo and Viktor 2004a). The number of input neurons is equal to the number of features of the data set. In addition, batch normalization is used along with *RelU* activation function in the hidden layer and Sigmoid activation in the output layer. Binary cross entropy loss function has been optimized using Stochastic Optimizer *Adam* with learning rate 0.01 and number of epochs being set to 100. This network is built in *PyTorch* sequential model. The degree of overlap parameter $C \in \{0.20, 0.15, 0.10, 0.05, 0.00\}$, as explained in the proposed method section.

As the proposed method assigns weights to the observations before training the network, it is a kind of pre-processing technique and hence we compare our proposed method with certain well known preprocessing based techniques with default parameter for handling class imbalance along with class overlapping, such as, SMOTE (Chawla et al. 2002), ENN (Wilson 1972), SMOTE + ENN (Batista et al. 2004), Tomek links (Tomek 1976) and SMOTE + Tomek (Batista et al. 2004). We evaluate the performance of the proposed method using five-fold stratified cross validation technique and compare with other existing methods on various metric measures.

5 Simulation studies

This section considers the evaluation of the proposed method on simulated data sets and compares its performance with some of the well known preprocessing based methods that are used for handling class imbalance along with class overlapping.

5.1 Simulation setup

In this section, we simulate twelve binary class data sets that can be divided into three groups (A, B and C) of different levels of class overlap, each comprising four data sets. Data sets of *Group A* are simulated using bivariate Gaussian distribution with the majority class having mean vector (0,0) and the minority vector having mean vector (3,3) with common covariance matrix I_{2*2}. *Group B* data sets are simulated in a similar way except that the mean vector of the minority class is shifted towards the majority class from (3,3) to (1,1) to increase the class overlap. To further increase the overlap, data sets in *Group C* are generated in such a way that the mean vector of the minority class is shifted from (1,1) to (0.8, 0.8). Figure 4 shows the structure of three groups with different levels of overlapping. For each group, we create four data sets with different class imbalance ratios, approximately equal to 1:2, 1:5, 1:10 and 1:20. The data sets thus generated are listed in Table 2 along with the number of observations in each class.

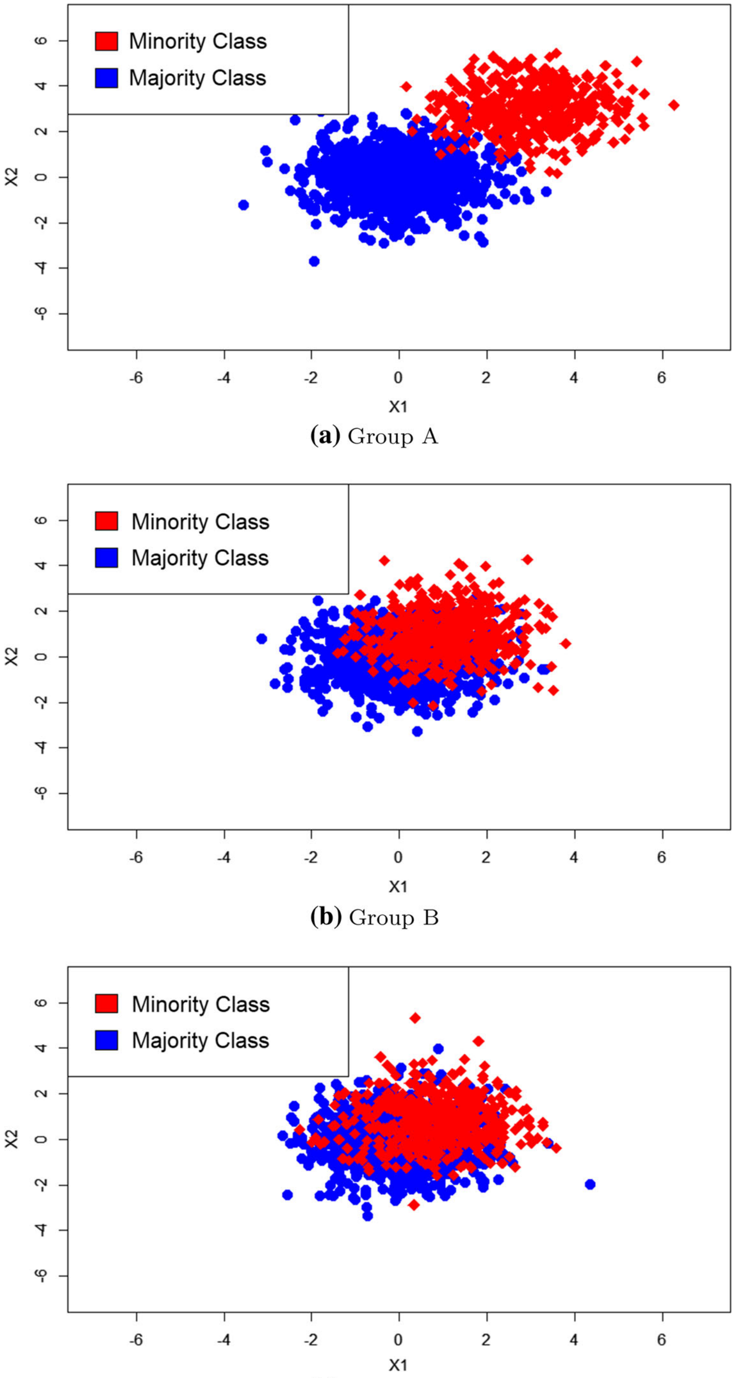
5.2 Results

Tables 3, 4 and 5 present the results of simulation studies for Groups A, B and C respectively. Tables 4 and 5 clearly indicate better performance of the proposed method for Groups B and C in terms of F-measure_1 (F1), G-Mean and AUC for all levels of imbalance ratios except *SimDataC.3*, where *SMOTE_ENN* performs slightly better than the proposed method in G-Mean measure. As the imbalance ratio increases, the performance gap between the proposed method and other methods becomes wider and particularly quite prominent in Group C data *SimDataC.4*, where the imbalance ratio is the highest. However, in case of Group A data sets where overlap is insignificant, SMOTE and SMOTE_TOMEK perform slightly better than the proposed method in terms of AUC and G-Mean, though the proposed method still performs much better in terms of F1.

6 Experiments on real-life data sets

We evaluate the proposed method on 23 real-life data sets and compare its performance with the performance of preprocessing based methods which were used for the simulation studies.

Fifteen out of 23 data sets are chosen from KEEL data sets repository and the maximum class imbalance among these data sets is 58.28 and the maximum number of observations is 1484 (Alcalá-Fdez et al. 2011). For large-scale data set, breast



(a) Group A

(b) Group B

(c) Group C

Fig. 4 Simulated data sets having different levels of overlapping

Table 2 Summary of the simulated data sets

Group Name	Data set	Majority	Minority	Mean (Majority)	Mean (Minority)	Imbalance Ratio (IR)
Group A	SimDataA.1	1000	500	(0, 0)	(3, 3)	2
	SimDataA.2	1250	250	(0, 0)	(3, 3)	5
	SimDataA.3	1364	136	(0, 0)	(3, 3)	10.02
	SimDataA.4	1428	72	(0, 0)	(3, 3)	19.83
Group B	SimDataB.1	1000	500	(0, 0)	(1, 1)	2
	SimDataB.2	1250	250	(0, 0)	(1, 1)	5
	SimDataB.3	1364	136	(0, 0)	(1, 1)	10.02
	SimDataB.4	1428	72	(0, 0)	(1, 1)	19.83
Group C	SimDataC.1	1000	500	(0, 0)	(0.8, 0.8)	2
	SimDataC.2	1250	250	(0, 0)	(0.8, 0.8)	5
	SimDataC.3	1364	136	(0, 0)	(0.8, 0.8)	10.02
	SimDataC.4	1428	72	(0, 0)	(0.8, 0.8)	19.83

Table 3 F-measure, G-mean & AUC values for Group *A* data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
SimDataA.1	ANN	0.971	0.976	0.979	0.987
	SMOTE	0.975	0.984	0.984	0.988
	ENN	0.965	0.975	0.979	0.984
	SMOTE_ENN	0.975	0.982	0.982	0.988
	Tomek_Link	0.971	0.977	0.980	0.987
	SMOTE_Tomek	0.976	0.984	0.984	0.988
	Prop_Method_C1	0.976	0.983	0.983	0.988
	Prop_Method_C2	0.976	0.983	0.983	0.988
	Prop_Method_C3	0.976	0.983	0.983	0.988
	Prop_Method_C4	0.976	0.983	0.983	0.988
	Prop_Method_C5	0.976	0.983	0.983	0.983
SimDataA.2	ANN	0.934	0.942	0.960	0.991
	SMOTE	0.955	0.983	0.983	0.991
	ENN	0.924	0.948	0.965	0.988
	SMOTE_ENN	0.950	0.978	0.979	0.990
	Tomek_Link	0.932	0.944	0.961	0.990
	SMOTE_TOMEK	0.953	0.982	0.982	0.990
	Prop_Method_C1	0.953	0.980	0.981	0.991
	Prop_Method_C2	0.954	0.980	0.981	0.991
	Prop_Method_C3	0.956	0.980	0.981	0.991
	Prop_Method_C4	0.956	0.979	0.980	0.991
	Prop_Method_C5	0.958	0.979	0.980	0.992
SimDataA.3	ANN	0.831	0.849	0.911	0.989
	SMOTE	0.921	0.980	0.980	0.992
	ENN	0.846	0.871	0.927	0.990
	SMOTE_ENN	0.911	0.976	0.976	0.990
	Tomek_Link	0.838	0.858	0.919	0.990
	SMOTE_TOMEK	0.917	0.979	0.979	0.991
	Prop_Method_C1	0.928	0.976	0.979	0.992
	Prop_Method_C2	0.929	0.977	0.979	0.992
	Prop_Method_C3	0.931	0.978	0.980	0.993
	Prop_Method_C4	0.932	0.979	0.981	0.993
	Prop_Method_C5	0.933	0.979	0.981	0.993
SimDataA.4	ANN	0.581	0.592	0.771	0.988
	SMOTE	0.796	0.963	0.963	0.987
	ENN	0.592	0.620	0.788	0.988
	SMOTE_ENN	0.780	0.957	0.958	0.986
	Tomek_Link	0.593	0.611	0.781	0.988
	SMOTE_TOMEK	0.801	0.963	0.964	0.987
	Prop_Method_C1	0.843	0.955	0.960	0.990

Table 3 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
	Prop_Method_C2	0.852	0.953	0.957	0.991
	Prop_Method_C3	0.857	0.950	0.955	0.991
	Prop_Method_C4	0.862	0.949	0.954	0.992
	Prop_Method_C5	0.859	0.943	0.948	0.992

Table 4 F-measure, G-mean & AUC values for Group *B* data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
SimDataB.1	ANN	0.630	0.707	0.729	0.843
	SMOTE	0.682	0.762	0.763	0.807
	ENN	0.677	0.757	0.761	0.805
	SMOTE_ENN	0.682	0.762	0.764	0.802
	Tomek_Link	0.666	0.742	0.752	0.841
	SMOTE_TOMEK	0.682	0.763	0.764	0.809
	Prop_Method_C1	0.682	0.762	0.763	0.812
	Prop_Method_C2	0.683	0.763	0.764	0.813
	Prop_Method_C3	0.683	0.763	0.764	0.813
	Prop_Method_C4	0.684	0.763	0.765	0.814
	Prop_Method_C5	0.684	0.764	0.765	0.815
SimDataB.2	ANN	0.406	0.521	0.637	0.920
	SMOTE	0.482	0.731	0.732	0.827
	ENN	0.489	0.665	0.705	0.894
	SMOTE_ENN	0.473	0.725	0.726	0.817
	Tomek_Link	0.445	0.562	0.659	0.920
	SMOTE_TOMEK	0.485	0.733	0.734	0.829
	Prop_Method_C1	0.499	0.734	0.737	0.847
	Prop_Method_C2	0.502	0.733	0.737	0.850
	Prop_Method_C3	0.504	0.732	0.737	0.854
	Prop_Method_C4	0.506	0.731	0.736	0.857
	Prop_Method_C5	0.507	0.729	0.734	0.861
SimDataB.3	ANN	0.151	0.228	0.550	0.954
	SMOTE	0.349	0.748	0.751	0.834
	ENN	0.329	0.492	0.636	0.946
	SMOTE_ENN	0.351	0.753	0.754	0.835
	Tomek_Link	0.205	0.308	0.569	0.952
	SMOTE_TOMEK	0.347	0.746	0.749	0.832
	Prop_Method_C1	0.390	0.756	0.761	0.876
	Prop_Method_C2	0.394	0.755	0.760	0.880
	Prop_Method_C3	0.401	0.756	0.762	0.884

Table 4 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
	Prop_Method_C4	0.401	0.748	0.756	0.889
	Prop_Method_C5	0.398	0.734	0.745	0.894
SimDataB.4	ANN	0.000	0.000	0.500	0.975
	SMOTE	0.226	0.746	0.748	0.856
	ENN	0.028	0.053	0.508	0.975
	SMOTE_ENN	0.219	0.741	0.743	0.848
	Tomek_Link	0.000	0.000	0.500	0.975
	SMOTE_TOMEK	0.224	0.746	0.747	0.851
	Prop_Method_C1	0.237	0.746	0.752	0.870
	Prop_Method_C2	0.241	0.749	0.755	0.873
	Prop_Method_C3	0.247	0.748	0.755	0.879
	Prop_Method_C4	0.260	0.749	0.757	0.891
	Prop_Method_C5	0.269	0.723	0.739	0.907

Table 5 F-measure, G-mean & AUC values for Group *C* data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
SimDataC.1	ANN	0.537	0.631	0.671	0.819
	SMOTE	0.630	0.717	0.719	0.766
	ENN	0.623	0.700	0.708	0.716
	SMOTE_ENN	0.626	0.710	0.713	0.743
	Tomek_Link	0.580	0.672	0.692	0.811
	SMOTE_TOMEK	0.625	0.713	0.714	0.762
	Prop_Method_C1	0.631	0.718	0.720	0.771
	Prop_Method_C2	0.631	0.719	0.720	0.771
	Prop_Method_C3	0.631	0.719	0.720	0.771
	Prop_Method_C4	0.631	0.718	0.720	0.770
	Prop_Method_C5	0.630	0.718	0.719	0.769
SimDataC.2	ANN	0.255	0.381	0.574	0.914
	SMOTE	0.418	0.679	0.680	0.789
	ENN	0.422	0.601	0.661	0.888
	SMOTE_ENN	0.430	0.689	0.690	0.796
	Tomek_Link	0.319	0.444	0.599	0.914
	SMOTE_TOMEK	0.425	0.684	0.685	0.793
	Prop_Method_C1	0.439	0.694	0.696	0.808
	Prop_Method_C2	0.433	0.688	0.690	0.807
	Prop_Method_C3	0.436	0.689	0.691	0.812
	Prop_Method_C4	0.439	0.689	0.692	0.818
	Prop_Method_C5	0.443	0.689	0.693	0.825

Table 5 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
SimDataC.3	ANN	0.049	0.077	0.514	0.953
	SMOTE	0.299	0.705	0.709	0.793
	ENN	0.215	0.345	0.575	0.947
	SMOTE_ENN	0.303	0.705	0.709	0.808
	Tomek_Link	0.093	0.155	0.528	0.952
	SMOTE_TOMEK	0.298	0.703	0.707	0.794
	Prop_Method_C1	0.319	0.702	0.709	0.838
	Prop_Method_C2	0.320	0.699	0.706	0.843
	Prop_Method_C3	0.320	0.694	0.702	0.847
	Prop_Method_C4	0.324	0.696	0.704	0.850
	Prop_Method_C5	0.332	0.695	0.707	0.861
SimDataC.4	ANN	0.000	0.000	0.500	0.975
	SMOTE	0.175	0.680	0.683	0.810
	ENN	0.000	0.000	0.500	0.975
	SMOTE_ENN	0.178	0.687	0.690	0.809
	Tomek_Link	0.000	0.000	0.500	0.975
	SMOTE_TOMEK	0.174	0.681	0.683	0.809
	Prop_Method_C1	0.191	0.704	0.709	0.825
	Prop_Method_C2	0.193	0.704	0.709	0.828
	Prop_Method_C3	0.196	0.701	0.708	0.835
	Prop_Method_C4	0.196	0.684	0.699	0.848
	Prop_Method_C5	0.192	0.617	0.656	0.889

cancer dataset has been used from the Knowledge Discovery and Data Mining Cup,[1] which contains 102,294 examples with an imbalance ratio of 163.20. Apart from this, three datasets have been considered from *corporate bankruptcy* domain, namely *USA*, *Japan* and *Polish* bankruptcy data (Zhou 2013; Zikeba et al. 2016).

USA data set contains observations from non-financial industry with financial status (Non-bankrupt or Bankrupt) as class label from 1981 to 2009. A bankrupt company is defined as the one whose reason for deletion is marked as "bankruptcy" or "liquidation" in the original Compustat North America dataset. As suggested by Zhou (2013), we use 10 explanatory variables to predict the financial status. These explanatory variables are: net income/total assets (NI/TA), current ratio (CR), retained earnings/total assets (RE/TA), working capital/total assets (WC/TA), EBIT/total assets (EBIT/TA), sales /total assets (S/TA), cash/total assets (C/TA), current assets/total assets (CA/TA), stock holder's equity/total debt (SHE/TD) and cash/current liabilities(C/CL). The number of bankrupt and non-bankrupt firms between year 1981 to year 2009 are 918 and 85,211 respectively, with an imbalance ratio of 92.82. Japan bankruptcy dataset consists of only non-financial firms and indicates whether they were bankrupt or non-bankrupt

[1] http://www.kdd.org/kdd-cup.

Table 6 Summary of the real-life data sets

Data sets	Total observations	Minority	Majority	IR
Pima	768	268	500	1.87
yeast-2_vs_4	514	51	463	9.08
glass0	214	70	144	2.06
ecoli1	336	77	229	3.36
yeast1	1484	429	1055	2.46
winequality-red-4	1599	53	1546	29.17
winequality-red-8_vs_6-7	855	18	837	46.5
winequality-white-3-9_vs_5	1482	25	1457	58.28
wisconsin	683	239	444	1.86
yeast-1_vs_7	459	30	429	14.3
yeast-2_vs_8	482	20	462	23.1
yeast5	1484	1440	44	32.73
yeast6	1484	1449	35	41.4
ecoli4	336	20	316	15.8
glass4	214	201	13	15.47
KDD_Cup_2008_Breast_Cancer[a]	102294	623	101671	163.20
csv result-1year	7027	6756	271	24.93
csv result-2year	10173	9773	400	24.43
csv result-3year	10503	495	10008	20.21
csv result-4year	9792	515	9277	18.01
csv result-5year	5910	410	5500	13.41
Japan	36637	59	36578	619.97
USA	86129	918	85211	92.82

[a] http://www.kdd.org/kdd-cup/view/kdd-cup-2008

during the period 1989 to 2009. As before, we use the same 10 explanatory variables. This data set has 59 bankrupt observations and 36,578 non-bankrupt observations, thus having an imbalance ratio of 619.97.

In case of Polish companies manufacturing sector data, the period of time considered was 2007-2013 for bankruptcy companies and 2000-2012 for operating companies. The data set is divided into 5 parts depending on the forecasting period. The 1st year data contains financial rates from 1st year of the forecasting period and the corresponding class label indicates bankruptcy status after 5 years. The 2nd year data contains financial rates from 2nd year of the forecasting period and the class label indicates bankruptcy status after 4 years. Similarly, 3rd year, 4th year, 5th year data sets have financial rates from 3rd, 4th and 5th years of the forecasting periods and the class labels indicate the bankruptcy status after 3, 2, and 1 year respectively. The characteristics of these data sets are listed in Table 6.

Five-fold stratified cross validation is used to compare the performance of the proposed method with other existing methods, except for *Japan* and *USA* data sets. Following Zhou (2013), for *Japan* and *USA* data sets, models are trained on observations from year 1981 to 2001 and tested on observations between 2002 and 2009. The results of the analysis are presented in Tables 7, 8, 9, 10 and 11 (Best results are highlighted in bold face).

It can be observed that the proposed method does really well on F-measure of minority class, G-Mean and AUC for different values of C. In the case of G-Mean, the proposed method outperforms the other methods on all data sets except *Japan* and *yeast-2_vs_4* data sets, where SMOTE performs better than the other methods. Similarly, while using *AUC* measure, the proposed method outperforms all other methods on all data sets except *Japan* and *yeast-2_vs_4*, where SMOTE and SMOTE_ENN perform better respectively. Figure 5 shows ROC graphs of corporate bankruptcy datasets. It can be observed from these graphs that the proposed method performs better on all data sets except *Japan* data set. We can verify these observations from AUC values of Tables 7, 8, 9, 10 and 11.

To assess whether the proposed method shows significant improvement over the existing methods, we conducted Wilcoxon Signed Rank Test (Richardson 2010) on the F-measure of minority and majority class, G-mean and AUC. The null and alternative hypotheses are as follows:

H_0: The median difference is zero
H_1: The median difference is positive

Wilcoxon signed-rank test ranks the absolute difference between two classifiers. If the null hypothesis is true, the sum of the ranks corresponding to positive differences $(W+)$ and that of negative differences $(W-)$ should be nearly equal. The null hypothesis is rejected in favor of the above alternative hypothesis only if the test statistic $W = W-$ is sufficiently small. For 23 data sets, to reject the null hypothesis at 0.05 significance level, W value must be less than 73 (Richardson 2010). Table 12 presents the details of Wilcoxon signed rank test for AUC values for the proposed method and SMOTE. As we can see from this table, $W+ = 255$, $W- = 21$, and thus W value = 21. $W < 73$ indicates that the proposed method is superior compared to SMOTE in terms of AUC measure. Table 13 presents the summary of $W+$, $W-$ and W values for Wilcoxon signed rank test when comparing the proposed method with the other methods on F-measure of both the classes, G-Mean and AUC. The statistical tests indicate that the proposed method outperforms the other methods in terms of *AUC*, *F-measure* minority and *G-mean* measure.

7 Discussion

In literature, few preprocessing based techniques exist for handling class imbalance and class overlap. Traditionally, SMOTE handles class imbalance by oversampling observations. However, oversampling is carried out without considering its location in the feature space and thus oversampling in the overlapping region degrades the performance of the classifier. Further, oversampling increases the size of the training set, thus increasing the training time substantially. Edited Nearest Neighbours (ENN) and Tomek Links are clean up techniques that remove the overlapping observations. These techniques are combined with SMOTE to handle overlap resulting in tech-

Table 7 F-measure, G-mean & AUC values for real-life data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
Pima	ANN	0.650	0.721	0.735	**0.834**
	SMOTE	0.662	0.737	0.739	0.794
	ENN	0.670	0.742	0.744	0.776
	SMOT_ENN	0.681	0.750	0.754	0.781
	Tomek_link	0.656	0.729	0.738	0.826
	SMOTE_Tomek	0.671	0.744	0.746	0.800
	Prop_Method_C1	0.687	0.758	0.760	0.815
	Prop_Method_C2	0.684	0.756	0.758	0.814
	Prop_Method_C3	**0.690**	**0.761**	**0.763**	0.817
	Prop_Method_C4	0.686	0.758	0.759	0.815
	Prop_Method_C5	0.680	0.753	0.755	0.814
yeast-2_vs_4	ANN	0.749	0.805	0.835	**0.977**
	SMOTE	0.742	0.901	0.904	0.966
	ENN	0.711	0.845	0.865	0.967
	SMOT_ENN	0.727	**0.918**	**0.920**	0.960
	Tomek_link	0.688	0.787	0.819	0.969
	SMOTE_Tomek	0.718	0.885	0.889	0.962
	Prop_Method_C1	0.747	0.892	0.897	0.966
	Prop_Method_C2	0.770	0.898	0.903	0.970
	Prop_Method_C3	0.777	0.900	0.905	0.971
	Prop_Method_C4	**0.781**	0.897	0.903	0.972
	Prop_Method_C5	0.764	0.879	0.886	0.971
glass0	ANN	0.716	0.773	0.795	**0.879**
	SMOTE	0.723	0.798	0.802	0.814
	ENN	0.718	0.794	0.805	0.807
	SMOT_ENN	0.708	0.785	0.792	0.793
	Tomek_link	**0.744**	0.810	0.819	0.867
	SMOTE_Tomek	0.728	0.800	0.805	0.823
	Prop_Method_C1	0.730	0.805	0.807	0.834
	Prop_Method_C2	0.742	0.814	0.816	0.843
	Prop_Method_C3	0.742	**0.814**	**0.816**	0.844
	Prop_Method_C4	0.735	0.808	0.811	0.839
	Prop_Method_C5	0.727	0.802	0.804	0.832
ecoli1	ANN	0.754	0.814	0.850	0.938
	SMOTE	0.772	0.887	0.888	0.916
	ENN	0.761	0.865	0.873	0.918
	SMOT_ENN	0.776	0.886	0.889	0.915
	Tomek_link	0.775	0.839	0.854	0.939
	SMOTE_Tomek	0.779	0.891	0.893	0.918
	Prop_Method_C1	0.808	0.890	0.892	0.938

Table 7 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
	Prop_Method_C2	0.810	0.891	0.893	0.939
	Prop_Method_C3	0.815	0.892	0.895	0.941
	Prop_Method_C4	0.821	0.894	0.897	0.944
	Prop_Method_C5	**0.832**	**0.898**	**0.901**	**0.948**
yeast1	ANN	0.518	0.618	0.670	**0.855**
	SMOTE	0.592	0.715	0.716	0.768
	ENN	0.606	0.725	0.727	0.803
	SMOT_ENN	0.591	0.712	0.716	0.749
	Tomek_link	0.546	0.649	0.685	0.850
	SMOTE_Tomek	0.590	0.713	0.714	0.768
	Prop_Method_C1	**0.617**	**0.733**	**0.735**	0.811
	Prop_Method_C2	0.617	0.733	0.734	0.812
	Prop_Method_C3	0.608	0.723	0.726	0.819
	Prop_Method_C4	0.603	0.719	0.722	0.818
	Prop_Method_C5	0.601	0.717	0.721	0.818

Table 8 F-measure, G-mean & AUC values for real-life data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
winequality-red-4	ANN	0.000	0.001	0.500	**0.983**
	SMOTE	0.173	0.656	0.690	0.908
	ENN	0.000	0.001	0.500	0.983
	SMOT_ENN	0.202	0.751	0.763	0.896
	Tomek_link	0.013	0.025	0.503	0.982
	SMOTE_Tomek	0.166	0.643	0.684	0.908
	Prop_Method_C1	0.249	**0.783**	**0.790**	0.918
	Prop_Method_C2	**0.251**	0.763	0.774	0.923
	Prop_Method_C3	0.216	0.721	0.740	0.917
	Prop_Method_C4	0.238	0.702	0.730	0.933
	Prop_Method_C5	0.313	0.644	0.703	0.962

niques SMOTE + ENN and SMOTE + Tomek. However, these methods suffer loss of information due to removal of some of the overlapping observations.

The proposed method uses an entirely different approach of considering different costs of misclassification for observations while training the ANN. Different costs of misclassification for observations are incorporated by assigning different weights to the observations, depending on its location in the feature space. Accordingly, a dataset is divided into three regions: safe zone, overlapping and noisy or outlier observations. The extent of overlapping is quantified by computing the propensity score. The proposed method initially uses a parameter C to assign all the noisy observations a

Table 8 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
winequality-red-8_vs_6-7	ANN	0.001	0.002	0.500	**0.989**
	SMOTE	0.171	0.518	0.658	0.963
	ENN	0.001	0.002	0.500	0.989
	SMOT_ENN	0.113	0.471	0.626	0.939
	Tomek_link	0.001	0.002	0.500	0.989
	SMOTE_Tomek	0.171	0.518	0.658	0.963
	Prop_Method_C1	0.143	0.677	0.731	0.906
	Prop_Method_C2	0.170	0.735	0.758	0.915
	Prop_Method_C3	0.174	**0.738**	**0.761**	0.916
	Prop_Method_C4	0.155	0.661	0.732	0.918
	Prop_Method_C5	**0.199**	0.542	0.644	0.950
winequality-white-3-9_vs_5	ANN	0.057	0.074	0.518	**0.992**
	SMOTE	0.124	0.615	0.685	0.940
	ENN	0.000	0.000	0.500	0.991
	SMOT_ENN	0.130	0.640	0.693	0.942
	Tomek_link	0.153	0.201	0.550	0.992
	SMOTE_Tomek	0.124	0.615	0.685	0.940
	Prop_Method_C1	0.201	0.663	0.762	0.959
	Prop_Method_C2	0.212	0.660	0.757	0.962
	Prop_Method_C3	0.234	**0.678**	**0.770**	0.965
	Prop_Method_C4	0.265	0.663	0.766	0.971
	Prop_Method_C5	**0.294**	0.493	0.655	0.986
wisconsin	ANN	0.953	0.965	0.965	0.975
	SMOTE	0.954	0.967	0.967	0.975
	ENN	**0.964**	**0.978**	**0.978**	**0.979**
	SMOT_ENN	0.963	0.976	0.976	0.979
	Tomek_link	0.952	0.965	0.965	0.974
	SMOTE_Tomek	0.952	0.966	0.966	0.973
	Prop_Method_C1	0.958	0.971	0.971	0.977
	Prop_Method_C2	0.960	0.973	0.160	0.977
	Prop_Method_C3	0.962	0.975	0.975	0.979
	Prop_Method_C4	0.962	0.975	0.975	0.979
	Prop_Method_C5	0.963	0.975	0.976	0.979
yeast-1_vs_7	ANN	0.042	0.060	0.512	**0.967**
	SMOTE	0.287	0.696	0.716	0.885
	ENN	0.127	0.187	0.514	0.967
	SMOT_ENN	0.250	0.684	0.700	0.852
	Tomek_link	0.061	0.087	0.518	0.967
	SMOTE_Tomek	0.290	0.705	0.724	0.883
	Prop_Method_C1	0.313	0.730	0.746	0.886

Table 8 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
	Prop_Method_C2	0.321	0.724	0.743	0.894
	Prop_Method_C3	0.310	0.717	0.736	0.893
	Prop_Method_C4	0.336	0.724	0.745	0.906
	Prop_Method_C5	**0.410**	**0.741**	**0.764**	0.931

Table 9 F-measure, G-mean & AUC values for real-life data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
yeast-2_vs_8	ANN	0.534	0.588	0.736	0.987
	SMOTE	0.257	0.661	0.709	0.930
	ENN	0.561	0.612	0.754	0.987
	SMOT_ENN	0.220	0.672	0.707	0.908
	Tomek_link	0.585	0.617	0.759	0.989
	SMOTE_Tomek	0.249	0.644	0.696	0.933
	Prop_Method_C1	0.548	0.648	0.760	0.985
	Prop_Method_C2	0.538	0.629	0.747	0.985
	Prop_Method_C3	0.578	0.642	0.760	0.987
	Prop_Method_C4	0.603	0.640	0.760	0.989
	Prop_Method_C5	**0.622**	**0.655**	**0.772**	**0.989**
yeast5	ANN	0.215	0.272	0.605	0.986
	SMOTE	0.670	0.955	0.956	0.985
	ENN	0.302	0.364	0.651	**0.987**
	SMOT_ENN	0.658	0.954	0.955	0.984
	Tomek_link	0.216	0.270	0.598	0.986
	SMOTE_Tomek	**0.670**	0.955	0.956	0.985
	Prop_Method_C1	0.558	0.942	0.944	0.972
	Prop_Method_C2	0.600	**0.958**	**0.959**	0.974
	Prop_Method_C3	0.602	0.957	0.958	0.975
	Prop_Method_C4	0.605	0.957	0.958	0.975
	Prop_Method_C5	0.607	0.954	0.956	0.975
yeast6	ANN	0.000	0.000	0.500	**0.988**
	SMOTE	0.384	0.869	0.876	0.966
	ENN	0.000	0.000	0.500	0.988
	SMOT_ENN	0.360	0.874	0.879	0.961
	Tomek_link	0.000	0.000	0.500	0.988
	SMOTE_Tomek	0.381	0.867	0.875	0.966
	Prop_Method_C1	0.410	0.874	0.883	0.964
	Prop_Method_C2	**0.424**	**0.874**	**0.884**	0.966

Table 9 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
	Prop_Method_C3	0.410	0.861	0.873	0.966
	Prop_Method_C4	0.400	0.846	0.862	0.966
	Prop_Method_C5	0.388	0.818	0.837	0.966
ecoli4	ANN	0.361	0.365	0.674	0.980
	SMOTE	**0.828**	0.894	0.905	**0.990**
	ENN	0.333	0.363	0.665	0.977
	SMOT_ENN	0.747	0.891	0.902	0.983
	Tomek_link	0.299	0.317	0.633	0.977
	SMOTE_Tomek	0.820	0.889	0.900	0.989
	Prop_Method_C1	0.814	0.951	0.957	0.986
	Prop_Method_C2	0.818	**0.951**	**0.958**	0.986
	Prop_Method_C3	0.818	0.951	0.958	0.986
	Prop_Method_C4	0.816	0.851	0.958	0.986
	Prop_Method_C5	0.810	0.903	0.913	0.987
glass4	ANN	0.361	0.736	0.818	0.982
	SMOTE	0.817	0.934	0.943	**0.985**
	ENN	0.655	0.798	0.859	0.976
	SMOT_ENN	0.655	0.932	0.935	0.966
	Tomek_link	0.680	0.739	0.820	0.982
	SMOTE_Tomek	**0.817**	0.934	0.943	0.985
	Prop_Method_C1	0.703	**0.941**	**0.944**	0.971
	Prop_Method_C2	0.694	0.940	0.943	0.971
	Prop_Method_C3	0.695	0.940	0.943	0.971
	Prop_Method_C4	0.718	0.940	0.943	0.973
	Prop_Method_C5	0.715	0.940	0.943	0.973

Table 10 F-measure, G-mean & AUC values for real-life data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
KDD_Cup_2008_Breast_Cancer	ANN	**0.597**	0.682	0.757	**0.998**
	SMOTE	0.461	0.859	0.871	0.995
	ENN	0.584	0.693	0.764	0.998
	SMOT_ENN	0.400	0.864	0.875	0.992
	Tomek_link	0.581	0.680	0.754	0.998
	SMOTE_Tomek	0.461	0.859	0.871	0.993
	Prop_Method_C1	0.423	**0.886**	**0.893**	0.991
	Prop_Method_C2	0.431	0.884	0.891	0.991
	Prop_Method_C3	0.440	0.868	0.878	0.992
	Prop_Method_C4	0.443	0.862	0.872	0.993
	Prop_Method_C5	0.478	0.791	0.812	0.994

Table 10 continued

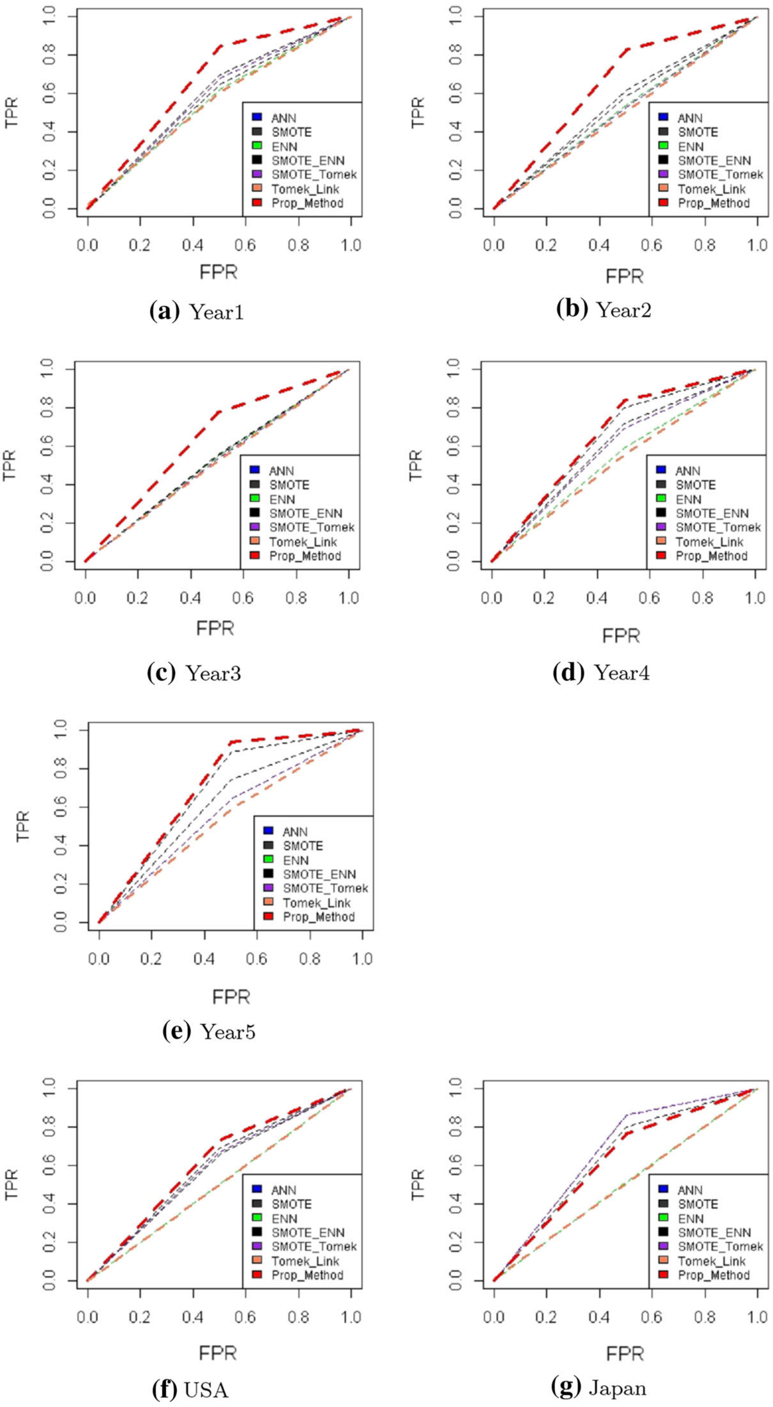
Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
csv_result-1year	ANN	**0.252**	0.339	0.590	**0.983**
	SMOTE	0.099	0.432	0.596	0.343
	ENN	0.261	0.426	0.618	0.976
	SMOT_ENN	0.098	0.449	0.598	0.364
	Tomek_link	0.307	0.406	0.614	0.983
	SMOTE_Tomek	0.106	0.485	0.619	0.406
	Prop_Method_C1	0.130	0.630	0.687	0.617
	Prop_Method_C2	0.138	0.656	0.701	0.661
	Prop_Method_C3	0.153	**0.690**	**0.726**	0.710
	Prop_Method_C4	0.149	0.679	0.717	0.706
	Prop_Method_C5	0.191	0.684	0.712	0.865
csv_result-2year	ANN	0.029	0.082	0.507	**0.980**
	SMOTE	0.093	0.406	0.577	0.320
	ENN	0.082	0.177	0.524	0.980
	SMOT_ENN	0.091	0.401	0.575	0.310
	Tomek_link	0.021	0.071	0.505	0.980
	SMOTE_Tomek	0.087	0.333	0.554	0.233
	Prop_Method_C1	0.115	0.597	0.654	0.581
	Prop_Method_C2	0.124	**0.632**	**0.673**	0.638
	Prop_Method_C3	0.118	0.600	0.655	0.591
	Prop_Method_C4	**0.126**	0.626	0.668	0.648
	Prop_Method_C5	0.118	0.595	0.629	0.655
csv_result-3year	ANN	0.113	0.208	0.536	**0.975**
	SMOTE	0.112	0.406	0.584	0.316
	ENN	0.090	0.204	0.528	0.973
	SMOT_ENN	0.106	0.384	0.570	0.285
	Tomek_link	0.078	0.179	0.523	0.975
	SMOTE_Tomek	0.105	0.333	0.560	0.236
	Prop_Method_C1	0.130	0.568	0.648	0.515
	Prop_Method_C2	0.133	0.578	0.654	0.531
	Prop_Method_C3	0.135	0.588	**0.659**	0.548
	Prop_Method_C4	0.135	0.593	0.656	0.563
	Prop_Method_C5	**0.146**	**0.638**	0.658	0.688
csv_result-4year	ANN	0.092	0.216	0.524	0.973
	SMOTE	0.144	0.546	0.638	0.500
	ENN	0.192	0.360	0.564	0.969
	SMOT_ENN	0.145	0.566	0.645	0.525
	Tomek_link	0.129	0.258	0.536	**0.974**
	SMOTE_Tomek	0.136	0.515	0.623	0.453
	Prop_Method_C1	0.210	0.725	0.737	0.782

Table 10 continued

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
	Prop_Method_C2	0.216	0.726	0.734	0.802
	Prop_Method_C3	0.227	**0.733**	**0.739**	0.823
	Prop_Method_C4	**0.229**	0.732	0.738	0.831
	Prop_Method_C5	0.207	0.688	0.696	0.824
csv_result-5year	ANN	0.256	0.390	0.577	**0.967**
	SMOTE	0.176	0.514	0.625	0.449
	ENN	0.381	0.561	0.652	0.961
	SMOT_ENN	0.209	0.625	0.683	0.597
	Tomek_link	0.270	0.403	0.583	0.967
	SMOTE_Tomek	0.156	0.410	0.579	0.318
	Prop_Method_C1	0.374	**0.783**	**0.787**	0.897
	Prop_Method_C2	0.380	0.782	0.786	0.900
	Prop_Method_C3	0.375	0.755	0.763	0.909
	Prop_Method_C4	**0.403**	0.749	0.761	0.925
	Prop_Method_C5	0.394	0.743	0.755	0.922

Table 11 F-Measure, G-Mean & AUC values for real-life data sets

Data sets	Method	F-measure_1	G-Mean	AUC	F-measure_0
Japan	ANN	0.021	0.033	0.506	**0.999**
	SMOTE	0.019	**0.853**	**0.855**	0.929
	ENN	0.001	0.002	0.500	0.999
	SMOT_ENN	0.019	0.786	0.793	0.940
	Tomek_link	0.001	0.003	0.500	0.999
	SMOTE_Tomek	0.021	0.804	0.810	0.944
	Prop_Method_C1	0.019	0.708	0.734	0.923
	Prop_Method_C2	0.018	0.703	0.730	0.924
	Prop_Method_C3	0.019	0.697	0.726	0.926
	Prop_Method_C4	0.021	0.694	0.725	0.930
	Prop_Method_C5	**0.032**	0.413	0.572	0.955
USA	ANN	0.000	0.000	0.500	**0.995**
	SMOTE	0.005	0.491	0.575	0.461
	ENN	0.000	0.002	0.500	0.995
	SMOT_ENN	0.007	0.631	0.655	0.681
	Tomek_link	0.000	0.000	0.500	0.995
	SMOTE_Tomek	0.005	0.547	0.602	0.542
	Prop_Method_C1	0.006	0.584	0.630	0.621
	Prop_Method_C2	0.006	0.606	0.645	0.663
	Prop_Method_C3	0.007	0.623	0.654	0.707
	Prop_Method_C4	0.008	**0.642**	**0.667**	0.754
	Prop_Method_C5	**0.011**	0.610	0.653	0.879



(a) Year1

(b) Year2

(c) Year3

(d) Year4

(e) Year5

(f) USA

(g) Japan

Fig. 5 Averaged ROC curves for Corporate bankruptcy data sets

Table 12 Illustration of significant test details of AUC between the proposed method and SMOTE

Data set	Proposed method	SMOTE	Difference	Rank
pima	0.763	0.739	0.024	10
yeast-2_vs_4	0.905	0.904	0.001	1.5
glass0	0.816	0.802	0.014	7
ecoli1	0.901	0.888	0.013	6
yeast1	0.735	0.716	0.019	8
winequality-red-4	0.790	0.690	0.1	18
winequality-red-8_vs_6-7	0.761	0.658	0.103	20
winequality-white-3-9_vs_5	0.770	0.685	0.085	15
wisconsin	0.976	0.967	0.009	5
yeast-1_vs_7	0.764	0.716	0.048	11
yeast-2_vs_8	0.772	0.709	0.063	13
yeast5	0.959	0.956	0.003	3
yeast6	0.884	0.876	0.008	4
ecoli4	0.958	0.905	0.053	12
glass4	0.944	0.943	0.001	1.5
KDD_Cup_2008_Breast_Cancer	0.893	0.871	0.022	9
csv_result-1year	0.726	0.596	0.13	22
csv_result-2year	0.673	0.577	0.096	17
csv_result-3year	0.659	0.584	0.075	14
csv_result-4year	0.739	0.638	0.101	19
csv_result-5year	0.787	0.625	0.162	23
Japan	0.734	0.855	-0.121	21
USA	0.667	0.575	0.092	16

$W+ = 255$, $W- = 21$; $W = W- = 21$

uniform weight less than or equal to the propensity score assigned to the overlapping observations. Further, the weights of all the observations in the minority class are proportionally increased to match the sum of majority class observations. Later, the weights for different types of outliers are adjusted depending on its distance from the centroid of the minority class. In this way, the resulting weights of safe zone observations are higher than those of overlapping observations which in turn are higher than noisy observations. Thus, the computation of weights using this approach helps ANN handle class imbalance as the total weight of minority class equals that of the majority class. Also, the weights of the overlapping observations decreasing with the extent of overlapping helps in increasing the true positive rate. Further, assigning least weight to outliers minimizes the shift of the decision boundary towards the majority class, resulting in reduction of false positive rate. Thus, the proposed method efficiently handles class imbalance and class overlap simultaneously.

Table 13 Summary of Significant test results

Feature Percentage	Proposed method	Existing Methods
AUC	W+ = 276, W- = 0, W = 0	ANN
	W+ = 255, W- = 21, W = 21	SMOTE
	W+ = 275, W- = 1, W = 1	ENN
	W+ = 260, W- = 16, W = 16	SMOTE_ENN
	W+ = 275, W- = 1, W = 1	Tomek
	W+ = 261.5, W- = 14.5, W = 14.5	SMOTE_Tomek
	W+ = 276, W- = 0, W = 0	ANN
	W+ = 255, W- = 21, W = 21	SMOTE
G-Mean	W+ = 275, W- = 1, W = 1	ENN
	W+ = 221, W- = 32, W = 32	SMOTE_ENN
	W+ = 276, W- = 0, W = 0	Tomek
	W+ = 260, W- = 16, W = 16	SMOTE_Tomek
	W+ = 253, W- = 23, W = 23	ANN
	W+ = 240, W- = 36, W = 36	SMOTE
F-Measure_1	W+ = 241.5, W- = 34.5, W = 34.5	ENN
	W+ = 245, W- = 8, W = 8	SMOTE_ENN
	W+ = 243, W- = 33, W = 33	Tomek
	W+ = 242.5, W- = 33.5, W = 33.5	SMOTE_Tomek
	W+ = 17.5, W- = 258.5, W = 258.5	ANN
	W+ = 233, W- = 20, W = 20	SMOTE
F-Measure_0	W+ = 66.5 , W- = 170.5, W = 170.5	ENN
	W+ = 248, W- = 5, W = 5	SMOTE_ENN
	W+ = 20, W- = 235, W = 235	Tomek
	W+ = 230.5, W- = 21.5, W = 21.5	SMOTE_Tomek

8 Conclusion

In this study, we have proposed an overlap sensitive neural network for handling class imbalance along with class overlapping and presence of noisy observations. The method incorporates different costs of misclassification by computing different weights for observations depending on its location in the feature space. Twelve simulated data sets that vary with respect to class imbalance and class overlap were analyzed and the results show that the proposed method outperforms the other methods in terms of different metric measures. Further, the method tested on 23 publicly available data sets also shows superior performance of the proposed method on various performance measures such as F-measure, G-mean and AUC. Thus, this approach of training the ANN efficiently handles the problem of class imbalance and class overlap.

References

Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple Valued Logic Soft Comput 17:255–287

Alibeigi M, Hashemi S, Hamzeh A (2012) DBFS: an effective density based feature selection scheme for small sample size and high dimensional imbalanced data sets. Data Knowl Eng 81:67–103

Alshomrani S, Bawakid A, Shim S-O, Fernández A, Herrera F (2015) A proposal for evolutionary fuzzy systems using feature weighting: dealing with overlapping in imbalanced datasets. Knowl-Based Syst 73:1–17

Barua S, Islam MM, Yao X, Murase K (2012) Mwmote-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans Knowl Data Eng 26(2):405–425

Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor Newsl 6(1):20–29

Batista GE, Prati RC, Monard MC (2005) Balancing strategies and class overlapping. In: International symposium on intelligent data analysis. Springer, Berlin, pp 24–35

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit 30(7):1145–1159

Burez J, Van den Poel D (2009) Handling class imbalance in customer churn prediction. Expert Syst Appl 36(3):4626–4636

Ceci M, Pio G, Kuzmanovski V, Džeroski S (2015) Semi-supervised multi-view learning for gene network reconstruction. PLoS ONE 10(12):e0144031

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Chawla NV, Japkowicz N, Kotcz A (2004) Special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl 6(1):1–6

Cleofas-Sánchez L, García V, Marqués A, Sánchez JS (2016) Financial distress prediction using the hybrid associative memory with translation. Appl Soft Comput 44:144–152

Cui Y, Jia M, Lin T-Y, Song Y, Belongie S (2019) Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 9268–9277

Das B, Krishnan NC, Cook DJ (2013) Handling class overlap and imbalance to detect prompt situations in smart homes. In: 2013 IEEE 13th international conference on data mining workshops. IEEE, pp 266–273

Elkan C (2001) The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. vol 17. Lawrence Erlbaum Associates Ltd, pp 973–978

Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. Comput Intell 20(1):18–36

Guo H, Viktor HL (2004a) Boosting with data generation: improving the classification of hard to learn examples. In: International conference on industrial, engineering and other applications of applied intelligent systems. Springer Berlin, pp 1082–1091

Guo H, Viktor HL (2004b) Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. ACM SIGKDD Explor Newsl 6(1):30–39

Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. Springer, Berlin, pp 878–887

He H, Garcia EA (2008) Learning from imbalanced data. IEEE Trans Knowl Data Eng 9:1263–1284

He H, Bai Y, Garcia EA, Li S (2008) Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE international joint conference on neural networks, 2008. IJCNN 2008. IEEE world congress on computational intelligence. IEEE, pp 1322–1328

Huang J, Ling CX (2005) Using AUC and accuracy in evaluating learning algorithms. IEEE Trans Knowl Data Eng 17(3):299–310

Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intelligent data analysis 6(5):429–449

Jo T, Japkowicz N (2004) Class imbalances versus small disjuncts. ACM SIGKDD Explor Newsl 6(1):40–49

Lee HK, Kim SB (2018) An overlap-sensitive margin classifier for imbalanced and overlapping data. Expert Syst Appl 98:72–83

Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp 2980–2988

López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci 250:113–141

McClelland JL, Rumelhart DE, Hinton GE (1988) The appeal of parallel distributed processing. Morgan Kaufmann, Burlington

Piras L, Giacinto G (2012) Synthetic pattern generation for imbalanced learning in image retrieval. Pattern Recognit Lett 33(16):2198–2205

Prati RC, Batista GE, Monard MC (2004) Class imbalances versus class overlapping: an analysis of a learning system behavior. In: Mexican international conference on artificial intelligence. Springer, Berlin, pp 312–321

Provost FJ, Fawcett T et al (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In: KDD-97 Proceedings, vol. 97. American Association for Artificial Intelligence, pp 43–48

Qu Y, Su H, Guo L, Chu J (2011) A novel SVM modeling approach for highly imbalanced and overlapping classification. Intell Data Anal 15(3):319–341

Richardson A (2010) Nonparametric statistics for non-statisticians: a step-by-step approach by Gregory W. Corder, Dale I. Foreman. Int Stat Rev 78(3):451–452

Shahee SA, Ananthakumar U (2018a) An adaptive oversampling technique for imbalanced datasets. In: Industrial conference on data mining. Springer, Berlin, pp 1–16

Shahee SA, Ananthakumar U (2018b) Synthetic sampling approach based on model-based clustering for imbalanced data. Int J Artif Intell Soft Comput 6(4):348–364

Shahee SA, Ananthakumar U (2019) An effective distance based feature selection approach for imbalanced data. Appl Intell 5:1–29

Simard PY, Steinkraus D, Platt JC et al (2003) Best practices for convolutional neural networks applied to visual document analysis. In: Icdar. vol 3

Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378

Tang Y, Gao J (2007) Improved classification for problem involving overlapping patterns. IEICE Trans Inf Syst 90(11):1787–1795

Tang W, Mao K, Mak LO, Ng GW (2010) Classification for overlapping classes using optimized overlapping region detection and soft decision. In: 2010 13th international conference on information fusion. IEEE, pp 1–8

Tax DM, Duin RP (2004) Support vector data description. Mach Learn 54(1):45–66

Thanathamathee P, Lursinsap C (2013) Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and adaboost techniques. Pattern Recogn Lett 34(12):1339–1347

Tharwat A (2018) Classification assessment methods. Appl Comput Inform 17(1):168–192

Ting KM (2002) An instance-weighting method to induce cost-sensitive trees. IEEE Trans Knowl Data Eng 3:659–665

Tomek I (1976) Two modifications of CNN. IEEE Trans Syst Man Cybernet 6:769–772

Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybernet 3:408–421

Xiong H, Wu J, Liu L (2010) Classification with classoverlapping: a systematic study. In: Proceedings of the 1st international conference on E-business intelligence (ICEBI2010). pp Atlantis Press

Yin L, Ge Y, Xiao K, Wang X, Quan X (2013) Feature selection for high-dimensional imbalanced data. Neurocomputing 105:3–11

Zhou L (2013) Performance of corporate bankruptcy prediction models on imbalanced dataset: the effect of sampling methods. Knowl-Based Syst 41:16–25

Zikeba M, Tomczak SK, Tomczak JM (2016) Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. Expert Syst Appl 58:93–101