JAISCR, 2014, Vol. 4, No. 2, pp. 133– 148
10.1515/jaiscr-2015-0004

AUTOMATED APPROACH TO CLASSIFICATION OF MINE-LIKE OBJECTS USING MULTIPLE-ASPECT SONAR IMAGES

Xiaoguang Wang[1], Xuan Liu[1], Nathalie Japkowicz[2] and Stan Matwin[3]

[1]*Faculty of Computer Science Dalhousie University, Canada*
e-mail:{x.wang; xuan.liu}@dal.ca

[2]*School of Electrical Engineering & Computer Science University of Ottawa, Canada*
nat@eecs.uottawa.ca

[3]*Faculty of Computer Science, Dalhousie University, Canada*
Institute of Computer Science, Polish Academy of Sciences, Poland
e-mail: stan@cs.dal.ca

Abstract

In this paper, the detection of mines or other objects on the seabed from multiple side-scan sonar views is considered. Two frameworks are provided for this kind of classification. The first framework is based upon the Dempster–Shafer (DS) concept of fusion from a single-view kernel-based classifier and the second framework is based upon the concepts of multi-instance classifiers. Moreover, we consider the class imbalance problem which is always presents in sonar image recognition. Our experimental results show that both of the presented frameworks can be used in mine-like object classification and the presented methods for multi-instance class imbalanced problem are also effective in such classification.

1 Introduction

To acquire high-resolution sonar imagery for the detection of mine like objects (MLO) and other objects of interest on the seabed, side-scan sonar equipped vehicles such as Autonomous Underwater Vehicles (AUVs) are frequently used by military forces or commercial organizations. For this purpose, Automatic Target Recognition (ATR) methods have been successfully applied to detect possible objects or regions of interest in sonar imagery [1]-[9]. Since many of the sonar images are of the same object from different sonar passes, there are multiple views of the same object at different ranges and aspect of the sonar. It is anticipated that the additional information obtained from additional views at an object should improve the classification performance over single-aspect classification. Recent researches [2][4][5] prove this anticipation by experimental result and find that although it is possible to obtain an accurate classification based upon a single image of an object, misclassifications can be reduced if the detection is based upon multiple views of the object.

In this paper, we use two methods combined with data fusion methods and multi-instance classification methods to deal with the class imbalanced problem in multi-view MLO classification. The first method is the cost-sensitive boosting algorithm [48] and the second is a classifier-independent method: over-sampling of multi-views of the minority class.

The remainder of the paper is organized as follows: Section II discusses previous work that has been done both in multi-view based classification and on the class imbalance problem. Section

III presents the fusion methodologies. Section IV presents the data preprocessing method used in our research. In section V, we consider the classification performance of all admissible multiple aspects including double and triple aspects for different types of mine like objects by studying the correct classification rates and the error rates as functions of the angular difference between aspects. In section VI, the class imbalance problem for single aspect mine countermeasure missions (MCM) datasets and a novel solution are presented. In section VII, the class imbalance problem for multi aspects MCM datasets and related concepts are presented and we present a novel cost-sensitive AdaBoost algorithm for this problem. This section also illustrates the efficiency of our algorithm as determined by experimentation, and offers some final remarks. Finally, Section VIII presents the conclusion, followed by the references.

2 Previous work

In these works, B. Zerr et al. [1] [3] firstly described a method to estimate the three-dimensional aspects of underwater objects using a sequence of sonar images. The sonar images are segmented into three kinds of regions: echo, shadow and background. A study they [2] conducted using sonar images of various objects and height profiles as features showed that the highest classification performance when imaging an object twice can be achieved with an angular increment of 90 degrees between the two images. M. Couillard et al. [4] extended this study and considered the classification performance of all admissible secondary aspects for different types of mine like objects by studying the correct classification rates and the error rates as functions of the angular difference between aspects. In their work, two different approaches have been used to combine multiple images of an object. The first one creates a new object for classification by combining the features of the two images to single vector. The second approach is simply to fuse the single aspect classification probabilities obtained from the classifier according to the desired angular increment between the images.

J. Fawcett et al. [5] investigated two approaches for fusing multiple views: fuse-feature and fuse-classification. In the first approach the two feature sets taken at different aspects were combined to form a large feature vector. Then a kernel based classifier was trained with this feature vector. In the second approach, they fused two individual-aspect classifications of two feature vectors using the Dempster-Shafer (DS) theory, which has frequently been used as an alternative to Bayesian theory and fuzzy logic for data fusion.

S. Reed et al. [6] [7] have also investigated the classification of a target by fusing several views using DS theory. They present a model to extend the standard mine/not-mine classification procedure to provide both shape and size information on the object. The difference between their work and others is that they generated the mass functions using a fuzzy functions membership algorithm based on fuzzy logic.

V. Myers and D. P. Williams [8] [9] introduced a model for classifying targets in sonar images from multiple views by using a partially observable Markov decision process (POMDP). This POMDP model allows one to adaptively determine which additional views of an object would be most beneficial in reducing the classification uncertainty.

In other related work, G. Dobeck fused multiple images from different frequency bands [11], J. Tucker et al. [12] fused multiple images from different platforms, and M. Azimi-Sadjadi et al. [13] fused multiple images from multi-aspect target echo classification.

These works have one common point in that they all use fusion methods to combine different views for classification. Although using fusion methods such as Dempster-Shafer fusion of single aspect classification results was shown to be effective in some cases [2][4][5], we can still anticipate a number of challenges and limitations in some ATR application using fusion methods[15]. It is thus necessary to develop other methods to combine different information from multiple views in this research.

In this paper, including the data fusion methodology, we present two frameworks for multi-aspect classification on side scan sonar images. The first one uses the Dempster-Shafer (DS) theory on multiple views of target which is not quite different from the methods mentioned [2][4][5]. In the second framework we use multi-instance method which is a methodology for a combination of the information of multiple views of target.

On the other hand, when applying ATR methods to detect possible MLOs, the number of naturally occurring clutter objects (such as rocks, shipwrecks or fish) that are detected always typically far outweighs the relatively rare event of detecting a mine. This means that the number of non-mine like objects is always much greater than the number of mine like objects. In this situation, the dataset is “imbalanced”. A dataset is imbalanced if the classes are not approximately equally represented. In imbalanced datasets, the number of one class is often much higher than the number of classes and a default classifier always predicts “the majority class”. For MLO classification, no matter whether we make the classification based upon a single image of an object or multiple images of an object, the training data sets are always class imbalanced. Our research shows that in both the cases of learning from single-view or multi-views of the objects, the performance of classifiers always suffered from the class imbalance problem.

For Automatic Target Recognition (ATR) methods used on MCM data sets, D. Williams et al. [10] used infinitely imbalanced logistic regression to solve the class imbalanced problem. That is the only work related to the class imbalanced problem of MLOs classification, especially in the case of multi aspects class imbalanced problem of MLOs classification.

3 Fusion methodologies

Data fusion is a technology which collates information from different sources considering the same scene in an attempt to provide a more complete description. When we try to combine multi-aspects sonar images for classification, the most common numerical fusion techniques used are Bayesian probability theory, Fuzzy systems and Dempster-Shafer theory.

Fuzzy systems contain a wealth of possible fusion operators. However, many of the operators are non-associative and the choice of operators is case dependent, which means the order in which the information is fused has an impact on the final result. Bayesian and Dempster-Shafer models have both been successfully applied but Dempster-Shafer theory provides some features that Bayesian theory does not. One of the most significant features is that Dempster-Shafer theory can consider the union of classes. This feature is used to improve the separability of different classes. Therefore the Dempster-Shafer (DS) method is a popular data fusion method which has been used by other authors for side scan sonar image classification.

The Dempster-Shafer method is based on two ideas: obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster’s rule for combining such degrees of belief when they are based on independent items of evidence.

The Dempster’s rule of combination is a purely conjunctive operation (AND). The combination rule results in a belief function based on conjunctive pooled evidence. This rule can also be used for multi aspect classification.

In DS theory, each unique class makes up a set called the frame of discernment $\theta = \{\omega_1, \omega_2, \ldots, \omega_M\}$. Belief is attributed to hypotheses within the power set through a basic probability assignment, called the mass function m(A).

Suppose that we have two views of target S_1 and S_2 and the mass functions $m_1(S)_1$ and $m_2(S)_2$. Base on the Despster’s rule, the mass after fusion for the set A is:

$$m_{12}(A) = \frac{\sum_{S_1 \cap S_2 = A} m_1(S_1) m_2(S_2)}{1 - \sum_{S_1 \cap S_2 = \emptyset} m_1(S_1) m_2(S_2)} \tag{1}$$

The classification rule for this case is

$$g(x_1, x_2) = \underset{i}{argmax}\, m_{12}(\omega_i) \tag{2}$$

In our research, as many authors, we use Dempster-Shafer theory as a choice for multi-aspect classification. In our algorithm, we use a training dataset for the single-aspect classifier and then save the predicted class labels from the testing data. Using T cross validation we can get a $T \times M$ output matrix. Let $\beta_i(k), k = 1, 2, \ldots, T$, correspond to the ith column of the prediction vector for the kth testing feature vector.

For n output vectors $\beta_i(k), i = 1, 2, \ldots n$ obtained from n single-aspect classifications, the n sets of masses are finally fused using Demsper's rule and the final decision is given by the classification rule $g(x_1, x_{2,..,}x_n)$.

In training datasets of MLO classification, each object has more than one view and each view is saved as an instance in the dataset. Therefore each object has a group of instances which has the same label. We call this group of instances a "bag". Bag is a term originally used in multi-instance learning which will be discussed in next section. In this paper, the n sets of masses, which are also n bags, are fused using Demsper's rule to get the final decision.

4 Multi-instance methodologies

Multi-instance learning (MIL) is another framework choice for multi-aspect classification. MIL is concerned with supervised learning but differs from normal supervised learning in two points: (1) it has multiple instances in an example, and (2) only one class label is observable for all the instances in an example.

The multiple instances learning problem can defined as:

Given:

- a set of bags $B_i, i = 1, \ldots, N$, their classification $c(B)_i\{0,1\}$, and the instances $e_{ij}(j = 1, \ldots, n_i)$ belonging to each bag.
- the existence of an unknown function f that classifies individual instances as 1 or 0, and for which it holds that $c(B_i) = 1$ if and only if there exists $e_{ij}B_i : f(e_{ij}) = 1$ (multi-instance constraint, MIC)

In our experiment we choose two popular multi-instance learning algorithms: the decision tree and the logistic regression methods.

4.1 Multi-instance Tree

Similar to a single-instance decision tree (like C4.5), the multi-instance tree is based on the information gain of a feature of the instance, the difference of the multi-decision tree and the single-decision tree is that instead of using the feature of one instance to develop the information gain, the growing of a multi-instance tree is based on the information gain of a feature to set of instances. The concept of information gain and entropy are extended to bags of instances in the MIL framework. Suppose S is a collection of instances which belong to p(S) positive bags and n(S) negative bags, F is the feature being considered as the splitting criterion and S_n is the collection of instances whose value of feature F is n. The extended information gain and entropy are defined as (3) and (4):

In this paper we use the multi-instance tree inducer (MITI) proposed by Blockeel et al. [16]. It implements the top-down decision tree learning approach known from propositional tree inducers such as C4.5 [17], with two key modifications: (a) nodes are expanded in best-first order guided by a heuristic that aims to identify pure positive leaf nodes as quickly as possible, and (b) whenever a pure positive leaf node is created, all positive bags containing instances in this leaf node are deactivated.

4.2 Multi-instance Logistic Regression (MILR)

For single-instance classification, Logistic Regression [49] assumes a parametric form for the distribution $Pr(Y|X)$, then directly estimates its parameters from the training data. The parametric model assumed by Logistic Regression in the case where Y is a boolean is:

$$Pr(Y = 1 \mid X) = \frac{1}{1+\exp(\omega_0 + \sum_{i=1}^{n} \omega_i X_i)} \quad (5)$$

and

$$Pr(Y = 0 \mid X) = \frac{\exp(\omega_0 + \sum_{i=1}^{n} \omega_i X_i)}{1+\exp(\omega_0 + \sum_{i=1}^{n} \omega_i X_i)} \quad (6)$$

However, the standard logistic regression model [49] does not apply to multi-instance data because the instances' class labels are masked by the "collective" class label of a bag. X. Xu and E. Frank [14] use a two-stage framework to upgrade linear logistic regression and boosting to MI data.

The instance-level class probabilities are given by

$$Pr(y = 1 \mid x) = \frac{1}{1 + exp(-\beta x)}$$

and

$$Pr(y = 0 \mid x) = \frac{1}{1 + exp(\beta x)}$$

$$Entropy_{multi}(S) = -\frac{p(S)}{p(s)+n(S)} \times log_2\left(\frac{p(S)}{p(s)+n(S)}\right) - \frac{n(S)}{p(s)+n(S)} \times log_2\left(\frac{n(S)}{p(s)+n(S)}\right) \quad (3)$$

$$InfoGain_{multi}(S,F) = Entropy_{multi}(S) - \sum_{n \in Values(F)} \frac{p(s_n)+n(s_n)}{p(s)+n(S)} \times Entropy_{multi}(s_n) \quad (4)$$

respectively, where β is the parameter vector to be estimated.

Given a bag b with n instances $x_i \in b$, we assume that the bag-level class probability is either given by

$$Pr(Y \mid b) = \frac{1}{n}\sum_{i=1}^{n} Pr(y \mid x_i) \quad (7)$$

or by

$$log\frac{Pr(y=1 \mid b)}{Pr(y=0 \mid b)} = \frac{1}{n}\sum_{i=1}^{n} log\frac{Pr(y=1 \mid x_i)}{Pr(y=0 \mid x_i)} \quad (8)$$

From (6) we can get (9) and (10).

Based on (9) and (12) we can estimate the parameter vector β by maximizing the bag-level binomial log-likelihood function (11) where N is the number of bags.

As usual, the maximization of the log-likelihood function is carried out via numeric optimization because there is no direct analytical solution. The optimization problem can be solved very efficiently because we are working with a linear model.

5 Class imbalance problem in Multi-Views MLO classification

For classification on single-views of mine like object (MLO) detection, we can apply many existing approaches such as sampling methods [27] [31] or cost-sensitive classification methods [29] [34] [36]. For classification on multi-views of MLO detection, to our knowledge there are very few discussions related to the multi-instance class imbalanced problems.

For the single-instance data imbalance problem, the machine learning community has addressed the issue of class imbalances in two different ways to solve the skewed vector space problem. The first method, which is classifier-independent, is to balance the distributions by considering the representative proportions of class examples in the distribution of the original data. The simplest way to balance a dataset is to under-sample or over-sample (randomly or selectively) the majority class, while maintaining the original minority class population [34]. One of the most common pre-processing methods to balance a dataset, Synthetic Minority Over-sampling Technique (SMOTE) [31], over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining any or all of the k minority class nearest neighbors. Evidence shows that synthetic sampling methods are effective when dealing with learning from imbalanced data [27] [31] [34].

Working with classifiers to adapt datasets is another way to deal with the single-instance imbalanced data problem. The theoretical foundation and algorithms of cost-sensitive methods naturally apply to imbalanced learning problems [29][30]. Thus, for imbalanced learning domains, cost-sensitive techniques provide a viable alternative to sampling methods. Recent research [27][29][34] suggests that assigning distinct costs to the training examples is a fundamental approach of this type, and various experimental studies of this [23][25][36] have been performed using different kinds of classifiers.

The work of [48] provides a cost-sensitive boosting algorithm for imbalanced multi-instance classification. This algorithm makes modifications based on the original Adaboost algorithm [19] for imbalanced multi-instance datasets.

The original AdaBoost [19] iteratively updates the distribution function over the training data. This means that for every iteration $t = 1, \ldots, T$, where T is a given number of the total number of iterations, the distribution function D_t is updated sequentially, and used to train a new hypothesis:

$$Pr(y=1 \mid b) = \frac{\left[\prod_i^n Pr(y=1 \mid x_i)\right]^{\frac{1}{n}}}{\left[\prod_i^n Pr(y=1 \mid x_i)\right]^{\frac{1}{n}} + \left[\prod_i^n Pr(y=0 \mid x_i)\right]^{\frac{1}{n}}} = \frac{\exp\left(\frac{1}{n}\beta\sum_i x_i\right)}{1+\exp\left(\frac{1}{n}\beta\sum_i x_i\right)} \tag{9}$$

$$Pr(y=0 \mid b) = \frac{\left[\prod_i^n Pr(y=0 \mid x_i)\right]^{\frac{1}{n}}}{\left[\prod_i^n Pr(y=1 \mid x_i)\right]^{\frac{1}{n}} + \left[\prod_i^n Pr(y=0 \mid x_i)\right]^{\frac{1}{n}}} = \frac{1}{1+\exp\left(\frac{1}{n}\beta\sum_i x_i\right)} \tag{10}$$

$$LL = \sum_{i=1}^{N} \left[y_i logPr(y=1 \mid b) + (1-y_i) logPr(y=0 \mid b)\right] \tag{11}$$

$$D_{t+1}(i) = \frac{D_t(i)\ exp(-\alpha_t y_i h_t(x_i))}{Z_t} \tag{12}$$

where $\alpha_t = \frac{1}{2} ln\left(\frac{1-\alpha_t}{\alpha_t}\right)$ is the weight updating parameter, $h_t(x_i)$ is the prediction output of hypothesis h_t on the instance x_i, ε_t is the error of hypothesis h_t over the training data, and Z_t is a normalization factor. Here each x_i is an n-tuple of attribute values belonging to a certain domain or instance space X, and y_i is a label in a label set Y.

Schapire and Singer [24] used a generalized version of Adaboost. As shown in [24], the training error of the final classifier is bounded as:

$$\frac{1}{m}\left|\{i : H(x_i) \neq y_i\}\right| \leq \prod_t Z_t \tag{13}$$

where

$$Z_t = \sum_i D_t(i) exp(-\alpha_t y_i h_t(x_i))$$

$$\leq \sum_i D_t(i)\left(\frac{1+y_i h_t(x_i)}{2} e^{-\alpha} + \frac{1+y_i h_t(x_i)}{2} e^{\alpha}\right) \tag{14}$$

Minimizing Z_t on each round, α_t is induced as:

$$\alpha_t = \frac{1}{2} ln\left(\frac{\sum_{i,y_i=h_t(x_i)} D_t(i)}{\sum_{i,y_i\neq h_t(x_i)} D_t(i)}\right) \tag{15}$$

The weighting strategy of AdaBoost identifies samples on their classification outputs as correctly classified or misclassified. However, it treats samples of different classes equally. The weights of misclassified samples from different classes are increased by an identical ratio, and the weights of correctly classified samples from different classes are decreased by an identical ratio.

Given: A multi-instance training dataset with a set of bags χ_i, $i = 1, \ldots, N$, where each bag can consist of an arbitrary number of instances and a given label: $\chi_i = \{x_i^1, x_i^2, \ldots, x_i^{n_i}; y_i\}$, $i = 1, \ldots, N$, $y_i \in \{-1, +1\}$, and each instance $x_i^{n_i}$ is an M-tuple of attribute values belonging to a certain domain or instance space $\mathbb{R}$.

Initialize $D_1(i) = 1/m$.

For $t = 1, \ldots, T$ && the constraint condition η is satisfied

Train a weak learner using distribution D_t.

Get a weak hypothesis $h_t : \chi \to \mathbb{R}$.

Choose $\alpha_t \in \mathbb{R}$.

Update: $D_{t+1}(i) = \frac{D_t(i) K_t(\chi_i, y_i)}{Z_t}$ (14)

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(\chi) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\chi)\right) \tag{15}$$

Figure 1. Cost-sensitive Adaboost for Multi-Instance Learning Algorithm

Since boosting is suitable for cost-sensitive adaption, motivated by [6]'s analysis and methods for choosing α_t, and several cost-sensitive boosting methods [30] [36] [29] for imbalanced single instance learning have been proposed in recent years. The work of [48] applied cost-minimizing techniques to the combination schemes of ensemble methods for imbalanced multi-instance datasets. This learning objective expects that the weighting strategy of a boosting algorithm will preserve a considerable weighted sample size of the minority class. A preferred boosting strategy is one that can distinguish different types of samples, and boost more weights on those samples associated with higher identification importance.

To denote the different identification importance among bags, each bag is associated with a cost item. For an imbalanced multi-instance dataset, there are many more bags with class label $y = -1$ than bags with class label $y = +1$. Using the same learning framework as AdaBoost, the cost items can be fed into the weight update formula of AdaBoost (Eq. (1)) to bias the weighting strategy. The proposed methods are similar to those proposed in Ref. [18]. Fig. 1 shows the proposed algorithms.

In the original adaboost, $K_t(\chi_i, y_i)$ is given as $\exp\alpha(-\alpha_t y_i h_t(\chi_i))$. In Cost-sensitive Adaboost for Multi-Instance Learning Algorithm, the modifications of $K_t(\chi_i, y_i)$ are then given by:

Ab1:

$$K_t(\chi_i, y_i) = \exp(-C_i \alpha_t y_i h_t(\chi_i)) \tag{16}$$

Ab2:

$$K_t(\chi_i, y_i) = C_i \exp(-\alpha_t y_i h_t(\chi_i)) \tag{17}$$

Ab3:

$$K_t(\chi_i, y_i) = C_i \exp(-C_i \alpha_t y_i h_t(\chi_i)) \tag{18}$$

Ab4:

$$K_t(\chi_i, y_i) = C_i^2 \exp(-C_i^2 \alpha_t y_i h_t(\chi_i)) \tag{19}$$

Respectively, for α_t and η, from [48] we can get (20)-(27)

On the other hand, similar to single-view MLO classification, we can also apply bag over-sampling, a classifier-independent method on imbalanced multi-views MLO classification. The Bag_Over_Sampling is a bag level over-sampling approach in which the minority class is over-sampled with replacement.

We have presented two approaches for the class imbalance problem in Multi-views MLO classification. One advantage of these approaches is that both of them are learner independent. Therefore these two approaches can be applied for the multi-instance learning and the DS fusion methods which are presented in previous two sections.

6 Data preprocessing

The first step of this classification task is the segmentation of the sonar images into three distinct regions: highlight or target echo (sound scattered by the target by active sonar), shadow (regions of low acoustic energy created by an object or seabed feature blocking the sound propagation) and background or seabed.

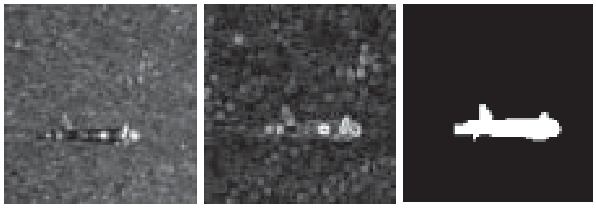

Figure 2. Example of an image processing result on an image provided by the Ocean System Lab, Heriot-Watt University

In mine countermeasure missions (MCM), sonar images collected by AUVs will convey important information about the underwater conditions. How to properly process the sonar images will have a significant impact on the subsequent MLOs detection and classification stages.

In the MCMs, a large part of sonar images collected by AUVs represent the background—seabed. In MLOs detection and classification, we are more interested in the object that lies on the seabed rather than the background. The areas from the images with only background information can be simply discarded. Image segmentation is a widely used image processing technique to detect target objects and segment the original images into small pieces that contain the target objects. The foreground objects are assumed to have a more complex texture than the seabed. Thus, the foreground object areas are obtained by using local range and standard deviation filters.

Instead of dealing with the whole sonar image, image segmentation allows us to only process the smaller pieces, reducing the future computational load. In this step, our goal is to delete image data that contain only background information and reduce the amount of data to be processed. Therefore whether the size, shape and location of the target object are accurately found is not a main concern in this step.

The objective of the image processing procedures at this point is data reduction rather than MLOs detection. Thus, a relatively high false alarm rate is acceptable.

Fig. 2 illustrates the extraction of foreground objects from a sonar image which was provided by the Ocean Systems Lab, Heriot-Watt University.

$$\alpha_{t\_Ab1}=\frac{1}{2}ln\left(\frac{1+\sum_{i,y_i=h_t(\chi_i)}C_iD_t(i)-\sum_{i,y_i\neq h_t(\chi_i)}C_iD_t(i)}{1-\sum_{i,y_i=h_t(\chi_i)}C_iD_t(i)+\sum_{i,y_i\neq h_t(\chi_i)}C_iD_t(i)}\right) \tag{20}$$

$$\eta_{Ab1}:\sum_{i,y_i=h_t(\chi_i)}C_iD_t(i)>\sum_{i,y_i\neq h_t(\chi_i)}C_iD_t(i) \tag{21}$$

$$\alpha_{t\_Ab2}=\frac{1}{2}ln\left(\frac{\sum_{i,y_i=h_t(\chi_i)}C_iD_t(i)}{\sum_{i,y_i\neq h_t(\chi_i)}C_iD_t(i)}\right) \tag{22}$$

$$\eta_{Ab1}:\sum_{i,y_i=h_t(\chi_i)}C_iD_t(i)>\sum_{i,y_i\neq h_t(\chi_i)}C_iD_t(i) \tag{23}$$

$$\eta_{t\_Ab3}=\frac{1}{2}ln\left(\frac{\sum_i C_iD_t(i)+\sum_{i,y_i=h_t(\chi_i)}C_i^2D_t(i)-\sum_{i,y_i\neq h_t(\chi_i)}C_i^2D_t(i)}{\sum_i C_iD_t(i)-\sum_{i,y_i=h_t(\chi_i)}C_i^2D_t(i)+\sum_{i,y_i\neq h_t(\chi_i)}C_i^2D_t(i)}\right) \tag{24}$$

$$\eta_{Ab3}:\sum_{i,y_i=h_t(\chi_i)}C_i^2D_t(i)>\sum_{i,y_i\neq h_t(\chi_i)}C_i^2D_t(i) \tag{25}$$

$$\alpha_{t\_Ab4}=\frac{1}{2}ln\left(\frac{\sum_i C_i^2D_t(i)+\sum_{i,y_i=h_t(\chi_i)}C_i^4D_t(i)-\sum_{i,y_i\neq h_t(\chi_i)}C_i^4D_t(i)}{\sum_i C_i^2D_t(i)-\sum_{i,y_i=h_t(\chi_i)}C_i^4D_t(i)+\sum_{i,y_i\neq h_t(\chi_i)}C_i^4D_t(i)}\right) \tag{26}$$

$$\eta_{Ab4}:\sum_{i,y_i=h_t(\chi_i)}C_i^4D_t(i)>\sum_{i,y_i\neq h_t(\chi_i)}C_i^4D_t(i) \tag{27}$$

Areas that do not have a reasonable size will be ignored.

For object detection tasks, an object should be detected through a single view, no matter where and how it lies on the seabed. Therefore, the features used should be robust to the location and orientation of the object. The grayscale histogram, a simple but informative statistical feature, is considered. In many image recognition systems, many complex features are used, but such features will inevitably increase the computational complexity, impeding the real time detection. The histogram is easy to calculate and robust to rotation. The distribution of the grayscale value can be well described by this feature.

In our experiment, the grayscale value (0-255) is divided into 16 bins with width 16. The grayscale histograms are normalized to the frequency that a pixel value falls into each bin. The MLOs are labeled as the positive examples.

7 Experimental Results Of Multi-aspects Images Classification

7.1 Classification on multi-views of object

In the experiments, we study the classification performances as a function of the number of aspects and compare the experimental result using DS and Multi-instance classifiers.

The binary dataset used in this empirical study is described in TABLE I. which has binary class. The negative examples denote the non_MLOs and the positive examples denote the MLOs. In this experiment each object has three views so we can study the classification performances as a function of the number of views. ROC curves are chosen as the measure technique for the classification. The experimental results are shown in Figure 3 and Figure 4.

Figure 3 shows the ROC curves as a function of the number of aspects using MITI as the classifier and Figure 4 shows the ROC curves using DS with decision tree as the classifier. We can find that for both classifiers, with more views used for classification, the performance is better.

Table 1. Multi-views mlo datasets

Datasets	# objects	# attribute	# positive examples	# negative examples
MLO_a	# 360	# 16	# 180	180

Table 2. Multi-instance class imbalanced datasets

Datasets	# objects	# attribute	# min objects	% min objects	# min instances	% min instances
MLO_1	561	16	58	10.34	116	10.34
MLO_2	555	16	64	11.53	144	12.18
MLO_3	425	16	65	15.29	158	17.67

7.2 Classification on class imbalanced multi-views of object

The datasets utilized in our empirical study are described in TABLE II. The percentage of minority bags varies from 8.27% to 15.29%. All datasets have a binary class. All of these datasets have more than one "view" on an object.

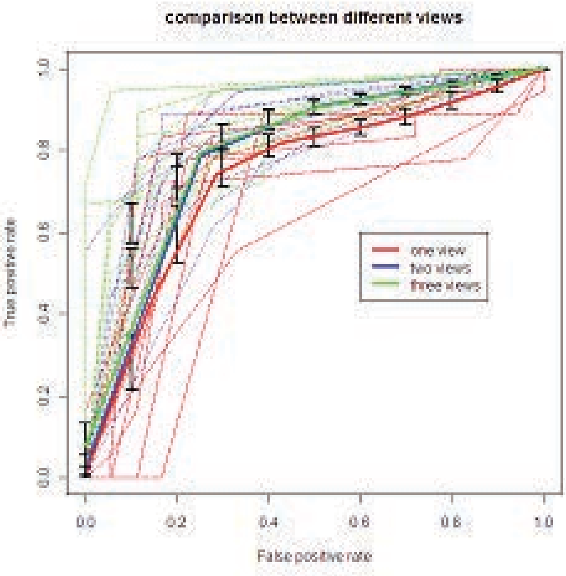

Figure 3. Classification performances as a function of the number of aspects using MITI as the classifier

To manage the significant number of possible combinations of images for multiple views, two fusion approaches are used to fuse the output probabilities.

The first approach is to use a multi-instance learning method to study the classification performances as a function of the number of aspects and the Multi-instance logistic regression classifier is chosen as the multi-aspect classifier. The second approach is fusing the output probabilities from the single aspect classifier. The Dempster-Shafer (DS) method is used to fuse the results as a decision fusion method and the logistic regression classifier is chosen as the single aspect classifier.

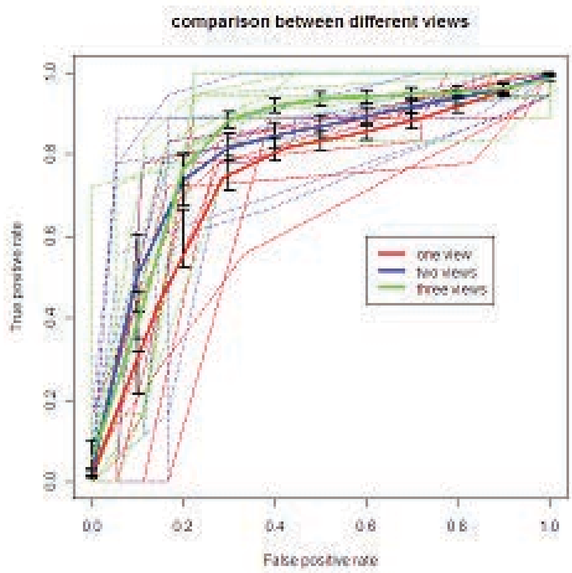

Figure 4. Classification performances as a function of the number of aspects using DS on decision tree as the classifier

Since in learning from extremely imbalanced data, a trivial classifier that predicts every case as the majority class can still achieve very high accuracy, the overall classification accuracy is often not an appropriate measure of performance. We choose Gmean [2] and F-measure as the measures for our algorithm and experiment. The definition of Gmean is listed in Table III.

Specificity: true Negative Rate

$$Acc^{-} = \frac{TN}{TN+FP} \tag{28}$$

Sensitivity: true Positive Rate

Table 3. Confusion matrix

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

Table 4. Multi-instance class imbalanced datasets

Datasets	# objects	# attribute	# cylinder	% manta	# wedding$_c ake$
MLO$_b$	279	16	93	93	93

$$Acc^+ = \frac{TP}{TP+FN} \tag{29}$$

$$Gmean = (Acc^- \times Acc^+)^{1/2} \tag{30}$$

TABLE XII and TABLE XIII show the experimental results of this study. Comparing the aspect classification rates with the two multi-aspect approaches, we see that collecting multiple views produces a significant increase in Gmean and F-measure for classification. Moreover, the multi-instance learning method gets better classification performance than the Dempster-Shafer (DS) method with single aspect classifier on all shapes on the same number of aspects combined.

7.3 Classification on MLOs with multi-views

We have three different shapes of MLOs which are cylinder, manta and wedding_cake shapes. After making a classification of MLOs and non_MLOs, we can keep on making a classification on what kind of shape the MLO belongs to. TABLE IV shows the details of this dataset. TABLE V to TABLE VII show the confusion matrices resulting from single-aspect classification using decision tree, multi-aspects classification using MITI and multi-aspects classification using DS with decision tree respectively.

TABLE VIII to TABLE X give the confusion matrices resulting from single-aspect classification using Logistic Regression, multi-aspects classification using MILR and multi-aspects classification using DS with Logistic Regression respectively.

From these classification results we can see that the classification performance, both on using the multi-instance framework and data fusion framework, were improved by using more “views” in the classification.

7.4 Statistical test method

As Friedman’s test [40] is a non-parametric statistical test for multiple classifiers and multiple domains, we performed it on the results in TABLE XII and TABLE XIII. The null hypothesis for this test is that all the classifiers perform equally, and rejection of the null hypothesis means that there is at least one pair of classifiers with significantly different performance. This test is performed on the multiplication results of Gmean and F-measure.

Friedman’s test result is shown in the TABLE XI.

Since Friedman’s test shows that these classifiers perform differently, we then applied Nemenyi’s post-hoc test [40] to determine which classifier has better performance than others. By comparing their q values [40] with the critical value $q_C = 3.22$, we can determine if one classifier is better than the other one: positive and bigger than q_C –lose; negative and the absolute value larger than q_C –win; other cases –equal.

The scores of all the classifiers in TABLE XII and TABLE XIII are presented in TABLE XIV. The result of 4-2-0 for Ab1 means that this classifier wins 4 times, ties 2 times, and loses zero times. If we set the scores as win=1, equal=0 and lose= -1, the score of each classifier can be calculated. The total score of these classifiers using MITI and DS with decision tree as base learners can also be calculated. From the result we can find that Ab1, Ab3 and Ab4 show better performance in these classifiers dealing with class imbalanced multiple views classification. On the other hand, combined with MITI, cost-sensitive boosting method has the chance to get the best performance in all presented classifiers.

Table 5. The Confusion Matrices Resulting From Single-Aspect Classification Using Decision Tree

(a)	***Single Aspect***		
	Cylinder	Manta	Wedding_Cake
Cylinder	58.9	24.9	16.2
Manta	11.9	73.5	14.6
Wedding_Cake	11.4	16.2	72.4

Table 6. The Confusion Matrices Resulting From Multi-Aspects Classification Using MITI

(b)	***Multi_Aspects***		
	Cylinder	Manta	Wedding_Cake
Cylinder	80.6	11.8	7.5
Manta	8.6	83.3	8.1
Wedding_Cake	10.2	3.2	86.6

Table 7. The Confusion Matrices Resulting From Multi-Aspects Classification Using DS Fusion With Decision Tree

(c)	***Multi_Aspects***		
	Cylinder	Manta	Wedding_Cake
Cylinder	73.6	17.2	9.2
Manta	4.8	84.4	10.8
Wedding_Cake	8.6	7.0	84.4

Table 8. The Confusion Matrices Resulting From Single-Aspect Classification Using Logisitic Regreesion

(a)	***Single Aspect***		
	Cylinder	Manta	Wedding_Cake
Cylinder	58.9	22.7	18.4
Manta	30.8	55.1	14.1
Wedding_Cake	4.9	15.1	80.0

Table 9. The Confusion Matrices Resulting From Multi-Aspects Classification Using MILR

(b)	***Multi_Aspects***		
	Cylinder	Manta	Wedding_Cake
Cylinder	72.6	16.7	10.7
Manta	10.3	76.3	5.4
Wedding_Cake	8.1	3.8	88.1

Table 10. The Confusion Matrices Resulting From Multi-Aspects Classification Using DS Fusion With Logisitic Regreesion

(c)	***Multi_Aspects***		
	Cylinder	Manta	Wedding_Cake
Cylinder	69.4	19.9	10.7
Manta	16.7	75.3	8.0
Wedding_Cake	7.0	8.1	84.9

Table 11. Friedman's test result of TABLE XII

Friedman χ^2	df	p-value	Critical χ^2
22.7143	6	0.000898	12.59
22.7143 > 12.59, hypothesis rejected			

Table 12. Friedman's test result of TABLE XIII

Friedman χ^2	df	p-value	Critical χ^2
16.7143	6	0.000898	12.59
16.7143 > 12.59, hypothesis rejected			

8 Conclusions

In this paper, we have considered the improving in the classification of sidescan sonar images obtained by using feature sets corresponding to multiple sonar views of the same object. There are two basic ways in which the multiple feature sets can be utilized. The first approach consists of fusing the multiple individual classification of the multiple feature vectors with the DS method. The second approach uses multi-instance classification methods to classify multiple feature vectors. Tree methods and Logistic Regression methods were chosen as the base learners for these two approaches in our experiments.

Moreover, class imbalanced problem in MLO classification was also considered in this paper. We presented two frameworks to deal with the multiple views class imbalanced problem in MLO classification. The first framework is a classifier-independent approach which uses bag over-sampling method to increase the minority instance numbers. The second framework is the Cost-sensitive boosting method for multiple views classification.

Our experimental results show that for MLO classification, given multiple views of an object, knowledge of the classification performance of multiple views is needed as by revisiting some of the contacts at suboptimal aspects, the overall survey time can be reduced. Using the multi-aspect side scan sonar images of various mine-like object shapes and non mine-like objects, we constructed secondary view classification curves to be used in conjunction with a path planning algorithm.

We have also studied the classification performances as a function of the number of aspects. Comparing the aspect classification rates with two multi-aspect approaches on different shapes, we see that collecting multiple views produces a significant increase in hit rate and a significant decrease in error rate for all mine shapes. Moreover, the multi-instance learning method gets better classification performance than the Dempster-Shafer (DS) method with the single aspect classifier on all shapes on the same number of aspects combined.

For the multi-views class imbalance problem, we have provided two novel frameworks for this problem: using data generated method or a cost-sensitive boosting method. Based on these methods, we have presented experimental analysis using different learning algorithms with MLO datasets. Experimental evidence derived from standard datasets was presented to support the cost-sensitive optimality of the proposed algorithms. We found that the cost-sensitive boosting with MIL consistently and significantly outperformed all the other methods tested.

References

[1] B. Zerr, B. Stage. Three-dimensional reconstruction of underwater objects from a sequence of sonar images, Proceedings of the IEEE International Conference on Image Processing, pp. 927–930, (1996).

[2] B. Zerr, B. Stage and A. Guerrero, Automatic Target Classification Using Multiple Sidescan Sonar Images of Different Orientations, SACLANT CEN Memorandum SM-309 (1997).

[3] B. Zerr, E. Bovio, B. Stage, Automatic mine classification approach based on AUV manoeuverability and cots side scan sonar, Proceedings of Goats 2001 Conference, La Spezia, Italy, (2001).

[4] M. Couillard, J. Fawcett, M. Davison and V. Myers, Optimizing time-limited multi-aspect classifi-

cation, Proceedings of the Institute of Acoustics 29(6), 89-96 (2007).

[5] J. Fawcett, V. Myers, D. Hopkin, A. Crawford, M. Couillard, B. Zerr. Multiaspect classification of sidescan sonar images: Four different approaches to fusing single-aspect information, Oceanic Engineering, IEEE Journal of 35(4): 863 –876 (2010).

[6] S. Reed, Y. Petillot, J. Bell, Model-based approach to the detection and classification of mines in side scan sonar, Applied Optics 43(2): 237– 246. (2004).

[7] S. Reed, Y. Petillot, J. Bell, Automated approach to classification of mine-like features in sidescan sonar using highlight and shadow information, IEE Proc. Radar, Sonar & Navigation 151 (No.1), 48-56, (2004).

[8] V. Myers, D. P. Williams, A POMDP for multiview target classification with an autonomous underwater vehicle, OCEANS, pp. 1-5, (2010).

[9] V. Myers, D. P. Williams, Adaptive Multiview Target Classification in Synthetic Aperture Sonar Images Using a Partially Observable Markov Decision Process, Oceanic Engineering, IEEE Journal of, On page(s): 45 - 55, Volume: 37 Issue: 1, Jan. (2012)

[10] D. Williams, V. Myers, and M. Silvious, "Mine Classification with Imbalanced Data," IEEE Geoscience and Remote Sensing Letters, Vol. 6, No. 3, pp. 528-532, July 2009.

[11] G. Dobeck, "Fusing sonar images for mine detection and classification," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 3710, 1999, DOI: 10.1117/12.357082.

[12] J. Tucker, N. Klausner, and M. Azimi-Sadjadi, "Target detection in M-disparate sonar platforms using multichannel hypothesis testing," in *Proc. OCEANS Conf.*, Quebec City, QC, Canada, 2008, DOI: 10.1109/ OCEANS.2008.5151818.

[13] M. Azimi-Sadjadi, A. Jamshidi, and G. Dobeck, "Adaptive underwater target classification with multi-aspect decision feedback," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 4394, 2001, DOI: 10.1117/12.445444.

[14] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In Lecture Notes in Computer Science, volume 3056, pages 272–281, April 2004.

[15] Hall, D. L. and Steinberg, A.. Dirty Secrets in Multisensor Data Fusion, http://www.dtic.mil. (2001).

[16] Blockeel, H., Page, D., Srinivasan, A.: Multi-instance tree learning. In: ICML (2005).

[17] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, (1993).

[18] Hosmer, David W.; Lemeshow, Stanley. Applied Logistic Regression (2nd ed.). Wiley. (2000).

[19] Freund,Y., Schapire,R.E.: Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148–156 (1996)

[20] Kubat, M., & Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proceddings of the Fourteenth International Conference on Machine Learning, 179-186 (1997)

[21] Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple instance problem with the axis-parallel rectangles. In: Artificial Intelligence, 89(1-2), 31–71 (1997)

[22] Maron, O., Lozano-Pz T.: A framework for multiple instance learning. In: Proc. of the 1997 Conf. on Advances in Neural Information Processing Systems 10, p.570-576 (1998)

[23] Fan, W., Stolfo,S.J., Zhang,J., Chan,P.K.: AdaCost: Misclassification Cost-Sensitive Boosting. In: Proc. Int'l Conf. Machine Learning, pp. 97-105 (1999)

[24] Schapire,R.E., Singer,Y.: Improved boosting algorithms using confidence-rated predictions. In: Machine Learning, 37 (3) 297–336 (1999)

[25] Ting, K.M.: A Comparative Study of Cost-Sensitive Boosting Algorithms. In: Proc. Int'l Conf. Machine Learning, pp. 983-990 (2000)

[26] Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: ICML (2000)

[27] Japkowicz, N.: Learning from Imbalanced Data Sets: A Comparison of Various Strategies. In: Proc. Am. Assoc. for Artificial Intelligence(AAAI) Workshop Learning from Imbalanced Data Sets, pp. 10-15. (Technical Report WS-00-05) (2000)

[28] Zhang, Q., Goldman, S. A.: EM-DD: An improved multiple instance learning technique. In: Neural Information Processing Systems 14 (2001)

[29] Elkan, C.: The Foundations of Cost-Sensitive Learning. In: Proc. Int'l Joint Conf. Artificial Intelligence, pp. 973-978 (2001)

[30] Ting, K.M.: An Instance-Weighting Method to Induce Cost-Sensitive Trees. In: IEEE Trans. Knowledge and Data Eng., vol. 14, no. 3, pp. 659-665 (2002)

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

[32] Zhang, M.L., Goldman, S.: Em-dd: An improved multi-instance learning technique. In: NIPS (2002)

[33] Andrews, S., Tsochandaridis, I., Hofman, T.: Support vector machines for multiple instance learning. In: Adv. Neural. Inf. Process. Syst. 15, 561–568 (2003)

[34] Batista, G.E.A.P.A., Prati,R.C., Monard,M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. In: ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 20-29 (2004)

[35] Blockeel, H., Page, D., Srinivasan, A.: Multi-instance tree learning. In: ICML (2005)

[36] Sun,Y., Kamel,M.S., Wong, A.K.C., Wang, Y.: Cost-Sensitive Boosting for Classification of Imbalanced Data. In: Pattern Recognition, vol. 40, no. 12, pp. 3358-3378 (2007)

[37] Foulds, J., Frank, E.: Revisiting multiple-instance learning via embedded instance selection. In: W. Wobcke & M. Zhang(Eds), 21st Australasian Joint Conference on Artificial Intelligence Auckland, New Zealand, (pp. 300-310) (2008)

[38] Leistner, C., Saffari, A., and Bischof, H.: MIForests: Multiple Instance Learning with Randomized Trees. In: Proc. ECCV (2010)

[39] Bjerring, L., Frank, E.: Beyond trees: Adopting MITI to learn rules and ensemble classifiers for multi-instance data. In: D. Wang & M. Reynolds (Eds.), AI 2011, LNAI 7106 (pp. 41-50) (2011)

[40] Japkowicz, N., Shah, M.: Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press (2011)

[41] Shawe-Taylor, J. and Cristianini, N.: Further results on the margin distribution. In: Proceedings of the 12th Conference on Computational Learning Theory, 278-285 (1999)

[42] Morik, K., Brockhausen, P., Joachims, T.: Combining Statistical Learning with a Knowledge-Based Approach - A Case Study in Intensive Care Monitoring. In: ICML: 268-277 (1999)

[43] Veropoulos, K., Campbell, C., & Cristianini, N.: Controlling the sensitivity of support vector machines. In: Proceedings of the International Joint Conference on Artificial Intelligence, 55–60. (1999)

[44] Chih-Chung Chang and Chih-Jen Lin.: LIBSVM : a library for support vector machines. In: ACM Transactions on Intelligent Systems and Technology, 2:27:1-27:27, (2011)

[45] Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin.: A practical guide to support vector classification. In: Technical Report, National Taiwan University. (2010)

[46] Bergstra, James; Bengio, Yoshua. :Random Search for Hyper-Parameter Optimization. In: J. Machine Learning Research 13: 281-305. (2012)

[47] Wang, X., Shao, H., Japkowicz, N., Matwin, S., Liu, X., Bourque, A., Nguyen, B.: Using SVM with Adaptively Asymmetric Misclassification Costs for Mine-Like Objects Detection. In: ICMLA (2012)

[48] Wang, X., Matwin, S., Japkowicz, N., Liu, X.: Cost-Sensitive Boosting Algorithms for Imbalanced Multi-instance Datasets. In: Canadian Conference on AI (2013)

[49] Hosmer, David W. Lemeshow, Stanley: Applied Logistic Regression. Wiley. ISBN 0-471-35632-8.(2000)

Table 13. Comparison of all presented algorithms for Class Imbalanced problem with MITI

Datasets	Methods	TPR_{min}	TNR_{min}	Gmean	Precision	Recall	F-measure
MLO_1	Base_Learner	7.8±2.0	97.8±0.2	27.5±3.6	76.6±5.0	41.4±6.9	53.7±7.1
	Bag_Over-Sampling	18.4±4.6	90.8±1.3	40.9±5.1	65.6±5.9	64.7±7.2	65.2±6.5
	Adaboost	9.8±2.9	97.2±0.2	30.8±4.6	75.9±6.2	47.1±8.3	58.1±8.1
	Ab1	22.4±3.4	93.1±0.4	45.7±3.6	75.7±3.0	70.6±4.6	73.1±3.8
	Ab2	12.1±2.9	94.9±0.4	33.8±3.9	69.1±4.3	53.2±6.6	60.1±5.6
	Ab3	54.6±2.3	75.8±1.4	64.3±1.8	69.3±2.1	91.2±0.7	78.7±1.6
	Ab4	54.9±2.7	75.6±1.9	64.4±1.5	69.2±1.3	91.3±0.9	78.7±0.9
MLO_2	Base_Learner	43.9±5.9	96.7±0.3	65.2±4.3	92.9±0.8	85.5±3.1	89.1±2.1
	Bag_Over-Sampling	47.7±4.7	94.7±0.2	67.2±3.4	89.9±1.2	87.3±2.2	88.5±1.7
	Adaboost	45.1±2.3	96.1±0.4	65.8±1.6	92.0±0.7	86.2±1.1	89.0±0.7
	Ab1	68.8±3.6	93.4±0.4	80.1±2.3	91.2±0.9	94.4±0.9	92.7±0.9
	Ab2	59.4±1.6	94.0±0.4	74.7±0.9	90.9±0.3	91.8±0.5	91.3±0.2
	Ab3	84.4±3.1	83.3±1.7	83.8±1.3	83.5±1.1	97.6±0.5	90.0±0.5
	Ab4	84.4±1.0	82.5±0.8	83.4±0.8	82.8±0.7	97.6±0.2	89.6±0.5
MLO_3	Base_Learner	49.5±1.4	95.9±0.5	68.9±1.1	92.4±1.1	84.4±0.7	88.2±0.8
	Bag_Over-Sampling	59.2±3.1	91.9±0.5	73.8±2.1	88.0±1.1	88.9±1.2	88.4±1.1
	Adaboost	56.2±1.5	95.4±0.4	73.2±1.0	92.4±0.7	87.6±0.7	90.0±0.6
	Ab1	72.3±0.5	90.7±0.7	81.0±0.6	88.6±0.9	93.5±0.2	91.0±0.5
	Ab2	67.9±3.8	91.9±0.6	79.0±2.3	89.4±0.8	92.1±1.3	90.7±1.0
	Ab3	87.7±2.1	82.7±1.7	85.2±1.3	83.6±1.4	97.5±0.5	90.0±0.9
	Ab4	91.3±2.1	77.6±2.7	84.2±1.7	80.3±2.0	98.3±0.4	88.4±1.2
MLO_4	Base_Learner	45.0±3.7	96.3±0.3	65.8±2.8	92.2±1.2	89.9±1.4	91.1±1.3
	Bag_Over-Sampling	48.1±2.8	96.1±0.3	68.0±2.0	92.4±0.8	91.1±0.9	91.7±0.8
	Adaboost	49.2±2.1	96.5±0.2	68.9±1.5	93.4±0.4	91.5±0.7	92.4±0.5
	Ab1	67.5±3.4	94.8±0.2	80.0±2.1	92.8±0.6	95.8±0.6	94.3±0.6
	Ab2	54.2±3.5	95.4±0.4	71.9±2.3	92.2±0.7	92.8±0.9	92.5±0.8
	Ab3	91.3±1.3	92.5±0.4	91.9±0.8	92.4±0.5	99.1±0.1	95.7±0.3
	Ab4	79.4±2.1	94.3±0.2	86.5±1.2	93.3±0.3	97.7±0.3	95.5±0.3

Table 14. Comparison of all presented algorithms for class imbalanced problem with DS on decision Tree

Datasets	Methods	TPR_{min}	TNR_{min}	Gmean	Precision	Recall	F-measure
MLO_1	Base_Learner	34.5±1.2	89.5±0.6	55.5±3.6	76.6±1.7	82.0±1.9	79.2±2.2
	Bag_Over-Sampling	41.4±3.1	87.5±0.9	60.2±0.8	76.8±0.8	86.0±0.2	81.1±1.1
	Adaboost	31.0±2.9	88.7±0.2	52.5±4.6	73.3±4.2	79.6±2.3	76.3±2.2
	Ab1	36.2±3.6	88.3±0.9	56.5±2.5	75.5±0.7	83.1±0.8	79.1±0.9
	Ab2	37.9±0.3	87.9±2.1	57.7±1.2	75.8±1.4	84.1±0.6	79.7±1.2
	Ab3	43.1±3.8	85.7±5.4	60.8±3.6	75.1±1.2	86.8±0.5	80.5±1.6
	Ab4	46.6±2.6	82.3±8.9	61.9±6.8	72.5±3.2	88.3±0.1	79.6±1.7
MLO_2	Base_Learner	51.6±2.5	88.0±0.6	67.4±2.1	81.1±1.6	89.1±0.3	84.9±0.6
	Bag_Over-Sampling	57.8±2.3	87.4±1.6	71.1±1.2	82.1±2.1	91.3±0.2	86.4±1.2
	Adaboost	51.6±3.4	90.2±0.3	68.2±2.1	84.1±1.5	89.1±0.5	86.5±0.9
	Ab1	59.4±2.3	89.4±0.6	72.9±1.1	84.9±0.8	91.8±0.3	88.2±0.5
	Ab2	71.9±1.3	84.1±0.6	77.8±1.1	81.9±0.5	95.1±0.1	88.0±0.3
	Ab3	78.1±2.1	81.3±0.8	79.7±1.2	80.7±0.2	96.5±0.2	87.9±0.3
	Ab4	75.0±1.6	81.3±0.9	78.1±0.8	80.0±1.0	95.8±0.1	87.2±0.4
MLO_3	Base_Learner	60.0±2.1	91.1±0.3	73.9±0.6	87.1±0.3	89.3±0.2	88.2±0.2
	Bag_Over-Sampling	63.1±1.6	88.3±0.9	74.6±0.4	84.4±0.2	90.4±0.4	87.3±0.1
	Adaboost	58.5±3.2	92.8±0.2	73.6±2.4	89.0±1.0	88.6±1.1	88.8±0.9
	Ab1	72.3±3.1	89.7±0.7	80.5±2.1	87.6±0.9	93.5±0.7	90.4±0.9
	Ab2	69.2±2.1	88.1±1.3	78.1±2.0	85.3±0.3	92.6±0.2	88.8±0.3
	Ab3	66.2±1.6	88.3±0.9	76.4±1.6	85.0±1.0	91.5±0.1	88.2±0.6
	Ab4	67.7±1.5	86.7±1.2	76.6±1.1	83.5±0.6	92.1±0.3	87.6±0.4
MLO_4	Base_Learner	49.2±3.6	92.3±0.3	67.4±2.6	86.4±0.8	91.5±0.8	88.9±0.6
	Bag_Over-Sampling	55.6±4.3	90.1±0.4	70.8±3.2	84.9±0.9	93.3±1.2	88.9±0.8
	Adaboost	57.1±2.6	94.0±0.2	73.3±2.0	90.5±0.2	93.7±0.1	92.0±0.1
	Ab1	84.1±1.2	89.4±0.2	86.7±1.0	88.8±0.5	98.3±0.2	93.3±0.2
	Ab2	93.4±0.3	84.7±1.0	89.1±0.6	86.0±0.7	99.4±0.0	92.2±0.3
	Ab3	96.8±0.1	81.4±1.5	88.8±0.2	83.9±1.1	99.7±0.0	91.1±0.1
	Ab4	92.1±0.1	82.8±1.3	87.3±0.4	84.2±0.6	99.2±0.1	91.1±0.1

Table 15. Comparison using the statistical test method (sorted by score from high to low)

		Base_ Learner	Bag_Over-Sampling	Adaboost	Ab1	Ab2	Ab3	Ab4
MITI	Gmean×(F-measure)	0-0-6	1-2-3	1-2-3	4-2-0	1-2-3	4-2-0	4-2-0
	Score	-6	-2	-2	4	-2	4	4
DS+Decision tree	Gmean×(F-measure)	0-2-4	0-2-4	0-2-4	3-3-0	3-3-0	3-3-0	3-3-0
	Score	-4	-4	-4	3	3	3	3
Total score		-10	-6	-6	7	1	7	7