A novel cost sensitive neural network ensemble for multiclass imbalance data learning

Peng Cao, Bo Li, Dazhe Zhao, Osmar Zaiane

***Abstract*—Traditional classification algorithms can be limited in their performance on imbalanced datasets. In recent years, the imbalanced data learning problem has drawn significant interest. In this work, we focus on designing modifications to neural network, in order to appropriately tackle the problem of multiclass imbalance. We propose a method that combines two ideas: diverse random subspace ensemble learning with evolutionary search, to improve the performance of neural network on multiclass imbalanced data. An evolutionary search technique is utilized to optimize the misclassification cost under the guidance of imbalanced data measures. Moreover, the diverse random subspace ensemble employs the minimum overlapping mechanism to provide diversity so as to improve the performance of the learning and optimization of neural network. Furthermore, the ensemble framework can determine the optimal amount of non-redundant components automatically. We have demonstrated experimentally using UCI datasets that our approach can achieve significantly better result than state-of-the-art methods for imbalanced data.**

I. Introduction

Recently, the class imbalance learning has been recognized as a crucial problem in machine learning and data mining [1, 2]. The issue occurs when the training data is not evenly distributed among classes. Imbalanced data learning is growing in importance and has been identified as one of the 10 main challenges of Data Mining [3]. This problem is also especially critical in many real applications, such as fraud detection and medical diagnoses. In these cases, standard classifiers generally perform poorly. Classifiers usually tend to be overwhelmed by the majority class and ignore the minority class examples. Most classifiers assume an even distribution of examples among classes, and are designed to maximize accuracy, which is not a good metric to evaluate effectiveness in the case of imbalanced training data. Therefore, we need to improve traditional algorithms so as to handle imbalanced data.

Peng Cao is with the Northeastern University, ShenYang, China (corresponding author e-mail: neusoftcp@ gmail.com).

Bo Li is with the Northeastern University, ShenYang, China (e-mail: l.b@neusoft.com).

Dazhe Zhao is with the Northeastern University, ShenYang, China (e-mail: zhaodz@neusoft.com).

Osmar Zaiane is with the University of Alberta, Edmonton, Canada (e-mail: zaiane@ualberta.ca).

Much work has been done in addressing the class imbalance problem. These methods can be grouped in two categories: the data perspective and the algorithm perspective [4]. The methods with the data perspective re-balance the class distribution by re-sampling the data space either randomly or deterministically [5-7]. Cost-sensitive learning tries to learn more characteristics of samples with the minority class by setting a high cost to the misclassification of a minority class samples [8-10]. Most existing imbalance data learning approaches so far are still limited to the binary class imbalance problems. The common approaches have been shown to be less effective or even cause a negative effect in dealing with multiclass tasks [9-10]. The multi-class imbalance problem poses a new challenge for classification task [9, 11]. It is desirable to develop an effective method to handle the multiclass imbalance issue.

We make the traditional cost-insensitive neural network classification algorithm into cost-sensitive by injecting the costs of misclassification into the output of posterior probability (CSNN) to overcome the multiclass imbalanced data classification issue. In the construction of cost sensitive learning, the parameter of misclassification cost plays an indispensable role. However, the appropriate cost cannot be required, and the empirical methods are not workable as the search space is expanded exponentially for the multiple classes as the number of imbalanced classes increases. We propose a new algorithm, called EDS (**E**volutionary search combined with **D**iverse random **S**ubspace ensemble), to help CSNN improving the classification performance on multiclass imbalanced data. In EDS, an evolutionary search method is used as the searching strategy to effectively search the optimum misclassification costs in the multiclass imbalance scenario according to the objective function defined with G-mean [12]. In addition, since combining multiple neural networks can achieve stronger generalization ability [13], we extend the random subspace method [14] to enhance the diversity by employing the minimum overlapping mechanism, so as to avoid overfitting in the procedure of the learning and optimization of CSNN. Furthermore, the ensemble can determine the optimal amount of non-redundant components automatically.

The outline of the paper is as follow: Cost sensitive neural network is described in Section 2. In Section 3, we describe the details of EDS-CSNN algorithm. In Section 4

we give experimental results of the evaluation of our algorithms. We conclude and present some future research directions in Section 5.

II. COST SENSITIVE NEURAL NETWORK

The main disadvantage of re-sampling techniques are that they may cause loss of important information or the model overfitting, as they change the original data distribution. Assigning distinct costs to the training examples seems to be the most effective approach to deal with the class imbalance problem [8-10, 15-16].

The cost-sensitive learning technique takes misclassification costs into account during the model construction, and does not modify the imbalanced data distribution directly. The standard neural network is cost insensitive. In standard neural network classifiers, the class returned is C^* by comparing probability of each class directly for each instance x according to Eq.(1).

$$C^* = \underset{C\in\{1,...,M\}}{argmax}(p_1(C_1 \mid x),..., p_M(C_M \mid x)) \quad (1)$$

where P_i denotes the probability value of each class from the neural network, $\sum_{i=1}^{M} P_i = 1$ and $0 \le P_i \le 1$. M is the number of the class.

The probabilities generated by standard neural network are biased in the imbalanced data distribution. To improve the recognition of the minority class, the probability a sample belongs to a certain class is replaced with the altered probability, which takes the misclassification costs into account, is found to be relatively a good choice in training CSNN [9]. This method uses the training set to construct a neural network, and the cost sensitivity strategy is introduced in the test phase. Given a certain cost matrix, the CSNN returns the class C^*, which is computed by injecting the cost according to Eq.(2).

$$\begin{aligned} C^* &= \underset{C\in\{1,...,M\}}{argmax}(\eta_1 p_1^*(C_1 \mid x),..., \eta_M p_M^*(C_M \mid x)) \\ &= \underset{C\in\{1,...,M\}}{argmax}(\eta_1 cost(C_1) p(C_1 \mid x),..., \eta_M cost(C_M) p(C_M \mid x)) \end{aligned} \quad (2)$$

where P_i^* denotes the class probabilities from the neural network combined with misclassification cost. η_i is a normalization term such that $\sum_{i=1}^{M} P_i^* = 1$ and $0 \le P_i^* \le 1$.

III. THE OPTIMIZED COST SENSITIVE NEURAL NETWORK ON IMBALANCED DATA

A. Threshold moving CSNN on binary class

In the binary class classification, given a certain cost matrix, the CSNN will classify an instance x into minority class (+) if and only if:

$$P(+ \mid x)cost(+) > P(- \mid x)cost(-) \quad (3)$$

Therefore the theoretical threshold for making a decision on classifying instances into minority is obtained as:

$$p(+ \mid x) > \frac{cost(-)}{cost(+)+cost(-)} = \frac{1}{1+C_{rf}} \quad (4)$$

where C_{rf} is ratio of two cost value, C_{rf}= $cost(+)/cost(-)$.

Thus the final decision criterion is only decided by the ratio misclassification cost C_{rf}. In the normal classification without considering the cost, C_{rf} is 1, that means both of the classes have the same weight. In the class imbalance scenario, we need to change the default decision threshold by adjusting the parameter of the C_{rf}. The value of C_{rf} plays a crucial role in the construction of CSNN, but the value is unknown in many domains where it is in fact difficult to specify the precise cost ratio information. It is not exact to set C_{rf} by inverting the ratio of prior distributions between minority and majority class. Therefore, to achieve the best performance on the imbalanced data, we adjust C_{rf} using a heuristic search strategy guided by evaluation measures. This is the method known as threshold moving [8, 16]. Adjusting the decision threshold can move the output threshold toward the inexpensive class such that instances with high costs become harder to be misclassified. The idea is based on the classifier producing probability predictions rather than classification labels, and it can converts any existing cost-**in**sensitive classifiers into cost-sensitive ones.

B. Evolutionary search CSNN on multiple classes

For binary classes (M=2), we can iteratively search the best C_{rf} for which the evaluation measure is maximized. However for a multiclass application (M>2), it is difficult to select the appropriate cost vector in the expanded space. Hence, searching an efficient cost setup becomes a critical issue for applying the cost sensitive neural network to multiclass applications. We propose a method that we call ECSNN (**E**volutionary search CSNN), which employ an evolutionary technique to carry out the meta-learning for searching an optimal class cost setup, which will be applied to the CSNN algorithm trying to improve the classification performance of the imbalanced datasets.

The popularity of evolutionary search has also instigated the development of numerous data mining algorithms [17]. In evolutionary search approach, individuals of a swarm move through a solution space and look for solutions for the data mining task at hand. We utilize the evolutionary search method as the optimization strategy to search the cost. The Artificial Bee Colony (ABC) algorithm is a swarm-based algorithm that was introduced based on the intelligent foraging behavior of honey bee swarms [18]. In the ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of the food source corresponds to the quality (fitness) of the associated solution. It consists in a set of possible solutions x_i that is

represented by the position of the food sources. On the other hand, in order to find the best solution, three classes of bees are used: employed bees, onlooker bees and scout bees. These bees have different tasks in the colony. In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, onlookers and employed bees carry out the exploitation process in the search space, and the scouts control the exploration process. The process is repeated through a predetermined number of cycles, called Maximum Number of Cycle (MCN). From the results obtained in [19], it can be concluded that the performance of the ABC algorithm is better than or similar to that of a Genetic algorithm and Particle swarm optimization although it uses less control parameters and it can be efficiently utilized for solving multimodal and multidimensional optimization problems.

ABC algorithm can optimize the proper class cost parameters based on the posterior probabilities produced by the neural network to maximize the performance measure. According to the specific objective function of artificial bee colony, we can optimize the class cost vector to achieve the optimum classification performance. In the procedure of searching for the best class cost vector, evaluation measures play a crucial role in both assessing the classification performance and guiding the modeling of the classifier. For imbalanced datasets, the evaluation metric should take into account the imbalance. The average accuracy is not an appropriate evaluation metric. We used G-mean as the fitness of the ABC algorithm to guide the optimization process. The G-mean is the geometric mean of accuracies measured separately on each class, which is commonly utilized when performance of both classes is concerned and expected to be high simultaneously. This metric has been used by several researchers for evaluating classifiers on imbalanced datasets.

$$G-mean = (\prod_{i}^{M} Acc_i)^{1/M} \quad (5)$$

We set the cost values for instances in the same class with an identical value. Let $Cost(C_i)$ denote the cost value of class i. Cost values of M classes then make up a cost vector [$Cost(C_1)$, $Cost(C_2)$, …, $Cost(C_M)$]. Since only the ratio of costs is effective, we set the cost of the largest class to 1, thus there are M-1degrees of freedom. This cost vector is encoded in each bee.

In order to optimize class cost parameters with avoiding overfitting, the training data is partitioned into two subsets, training subset and validation subset. The training subset is used to construct the cost sensitive neural network classifier that yields posterior probabilities, while the validation subset is used for obtain the optimal cost vector. The detailed algorithm of ECSNN is shown in **Algorithm 1.** The wrapper paradigm can discover the *bestCostVector* for a dataset based on optimizing evaluation functions. When predicting test instance, all the class posterior probabilities have to be multiplied with the corresponding cost parameter. The algorithm belongs to the post-processing strategy, which does no change the data distribution and distorting the information like re-sampling techniques.

Algorithm 1 *ECSNN*

Input: Training set *TrainingSet*, Test set *TestSet*, Maximum Cycle Number *MCN*; Population size *SN*; Limit *limit*

Training phase:

1. Separate *TrainingSet* into *TrSet* (80%) for training and *ValSet* (20%) for validation
2. Construct a classifier *L* on the *TrSet;*
3. Predict probability estimates on the *ValSet*
4. Initialize the food source positions x_i, i =1...*SN*
5. Obtain the G-mean GM_i with the *CostVector* optimized in x_i
6. *cycle=1*

repeat

7. Produce new solutions x_i' for the employed bee and evaluate them with G-mean
8. Apply the greedy selection process
9. Calculate the probability p_i for the solutions x_i'
10. Produce new solutions v_i for the onlooker from solutions x_i' selected depending on p_i and evaluate them
11. Apply the greedy selection process
12. Determine the abandoned solution for the scout according to the value of *limit*, if exists, and replace it with a new randomly produced solution
13. Memorize the best solution achieved so far; *cycle*++

until cycle=*MCN*

14. Obtain the *bestCostVector* from the best solution

Test phase:

15. Generate probabilities of each instance on the *TestSet* with *L*
16. Obtain the class of each instance with *bestCostVector* using **Eq**.(2)

C. ECSNN combined with diverse random subspace ensemble, EDS-CSNN

The ECSNN algorithm introduced above is based on the outputs of neural network on the whole feature space. The performance of optimization by ABC searching in ECSNN depends on the outputs of posterior probabilities generated by neural network. However, the outputs generated may be not accurate due to irrelevant features or noisy instances. Moreover, the imbalanced problem is often accompanied by high dimensional data in the practical domain [20-21]. All the issues can decrease the performance of ECSNN. Therefore, we propose an ensemble, diverse random subspace, an improvement of random subspace [14], can provide an effective and diverse framework to improve the performance of learning and optimization of ECSNN (EDS-CSNN).

The use of different spaces for ensemble construction has been extensively explained in recent research. Ho showed that the random subspace was able to improve the

generalization error [14]. In the random subspace method, individual classifiers are built by randomly projecting the original data into feature subspaces and training a proper base-learner on these subspaces. Since the random subspace combines multiple classifiers of this type, each with a random bias based on the features it sees, random subspace often prove more effective than learning the base classifier on all of the features.

Since diversity is known to be an important factor affecting the generalization performance of ensemble methods, several means have been proposed to get varied base-classifiers inside an ensemble. We propose an improvement of random subspace method, called diverse random subspace in order to obtain more diversity in each classifier. The common random subspace method is extended by integrating bootstrapping samples to obtain more diversity in each classifier at first. In the bootstrapping method, different training subsets are generated with uniform random selection with replacement. In addition, in the random subspace method, different features in each training subsets are randomly chosen for producing component classifiers. However, this cannot ensure the diversity of each subset since the instances and the features are chosen randomly. Therefore, for improving diversity between each subset, we use a formulation to make sure each subset is diverse. Firstly, we introduce a concept of *overlapping rate:*

$$Overlapping\ rate = \frac{subSet_i \cap subSet_j}{N_{fea} \times N_{ins}} \quad (6)$$

where the *subset* is the subset within a certain subspace, N_{fea} and N_{ins} are the feature size and instance size of each *subset.* All the subsets are of equal size; e.g., in **Figure 1**, the *overlapping rate* is 16%.

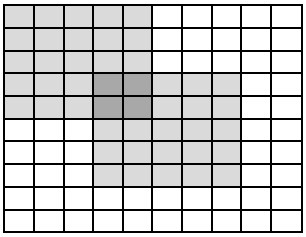

Figure 1 The overlapping rate between two subsets

In addition, we guarantee that the class ratio of each subset follows the one of the original training data distribution. We quantify data diversity between each subset with the data overlapping region, which measures the proportion of feature and instance subspace overlap between the training data of different classifiers in the ensemble. We then introduce a threshold T_{over} to control the intersection between each subset. The overlapping rate of all the subsets needs to be smaller than the threshold T_{over}. Therefore T_{over} is critical to the performance of the ensemble. If it is too large, the subsets lack diversity. If it is too low, the ensemble size is small, diminishing the advantage of ensemble classification. It is a trade-off between the diversity and the required ensemble size.

Through quantifying data diversity between each subset for a component classifier with the data overlapping region which measures the proportion of feature and instance subspace overlap between the training data of different classifiers in the ensemble, we can guarantee the diversity of subsets provided to each classifier, and at the same time provide a way to adaptively determine in an iterative way the number of classifiers in the ensemble. The *GenerateDiverseSets* algorithm can be described as in **Algorithm 2**. The function ***isDiverse***(D_s^k, *DiverseSet*, T_{over}) examines if the new projection D_s^k is diverse enough from the previously collected projections in *DiverseSet* based on the overlapping region threshold T_{over}. The generation of projections stops when there is stagnation – i.e. after enough trials, no new projection is diverse enough from the collected subspaces. Hence, the number of individual classifiers is determined dynamically.

Algorithm 2 *GenerateDiverseSets*

Input: Training set *TrainingSet*, Ratio of bootstrap samples R_s, Ratio of feature subspace R_f, Overlapping region threshold T_{over}, Stagnation rate *sr=100*

1. *change*=0; *DiverseSets*={};
   while *change*<*sr* **do**
2. A bootstrap sample D_s selected with replacement from *TrainingSet* with R_s
3. Generate subset D_s^k by selecting a random subspace with R_f
   if isDiverse(D_s^k, *DiverseSets*, T_{over})==true
4. ***then*** *DiverseSets->add*(D_s^k); *change*=0;
5. ***else*** *change=change*+1;
   end if
   end while

Output: *DiverseSets*

Algorithm 3 *EDS-CSNN*

Input: Training set *TrainingSet*, Test set *TestSet*, Ratio of bootstrap samples R_s, Ratio of feature subspace R_f, Overlapping region threshold T_{over},

Training phase:

1. *DiverseSets* = ***GenerateDiverseSets****(TrainingSet, R_s, R_f, T_{over});*
   for each subset D_k in *DiverseSets*
2. Construct a classifier L_k in D_k
3. Obtain *bestCostVector* according to the Algorithm 1.
4. L_k*->Subspace= subspace*(D_k)
5. L_k*->bestCostVector = bestCostVector*
6. *Ensemble=Ensemble* $\cup$ L_k
   end for

Testing phase:

7. Calculate output from each classifier L_k of *Ensemble* with its *bestCostVector* in its *Subspace* on the *TestSet*
8. Generate the final output by aggregating all the outputs

After obtaining the diverse set, both the construction of neural network and optimization of class costs are conducted on each subspace instead on the whole feature space, then ultimately combine classifiers with different characteristics and achieve improved performance. Therefore, varying the feature subsets gives an opportunity to control the diversity of the feature sets provided to each classifier in the ensemble and capture possible patterns that are informative on classification. The procedure of training and optimization of CSNN can be carried out in parallel to reduce the learning time. In addition, each CSNN classifier is modeled in the reduced subset with fewer instances and features. Hence the computational time is acceptable. **Algorithm 3** illustrates the EDS-CSNN algorithm.

IV. EXPERIMENTS

A. Dataset description

We choose twelve publicly available datasets from different domains with differing levels of class imbalance and number of class. **Table 1** shows the varying characteristics of the datasets, which are comprised of a mixture of continuous and nominal values.

Table 1. The data sets used for experimentation

Dataset	*C*	*Inst.*	*F*	*Class distribution*
Cmc	3	1473	9	629/333/511
Balance	3	625	4	49/288/288
New Thyroid	3	215	5	150/35/30
Car	4	1728	6	1210/384/69/65
Annealing	4	898	38	8/99/684/67/40
Nursery	4	12958	8	4320/328/4266/4044
Page	5	5473	10	4913/329/28/88/115
Ecoli	5	327	7	143/77/52/35/20
Cleveland	5	303	13	164/55/36/35/13
Glass	6	214	9	70/76/17/13/9/29
Satimage	6	6435	36	1533/703/1358/626/707/1508
Yeast	10	1484	9	463/429/244/163/51/44/35/30/20/5

B. Experiment design

In our empirical experiments, the Backpropagation neural network is employed as the base learner in our empirical experiments. The number of input neurons is equal to the number of features for each dataset, and the number of neurons in the hidden layer is set to be 10. The sigmoid function is used as the activation function, and the inner training epochs is set to be 200 with a learning rate of 0.1. All of methods in the comparison were implemented in Java on the platform of the WEKA

Like most of the optimization algorithms, the ABC algorithm also has control parameters to be set before running the algorithm. According to the introduction of ABC above, it is clear that there are three control parameters of basic ABC: the number of food sources which is equal to the number of employed or onlooker bees (*SN*), the value of limit (*limit*) and the maximum cycle number (*MCN*). We set the *SN* twice as many as the number of classes in our experiment. Through experiments, we found that the optimization can get a good and stable solution before the 500 cycles, thus the *MCN* is set to 500. The control parameter *limit* is the core parameter of the algorithm dictating the occurrence of scout bees that are responsible for providing the diversity in the population. We set the *limit* as the square of the number of class, which is generally appropriate.

C. Experiment I: Evaluating the effectiveness of EDS-CSNN

In this experiment, we assess the performance of the EDS algorithm on the CSNN and compared it with original random subspace (ES-CSNN) as well as the single ECSNN. In ES-CSNN and ECSNN, each training set is separated randomly into training subset (70%) for training cost sensitive classifier and validation subset (30%) for adjusting decision threshold. The ensemble size of ES-CSNN is set to 50. In the construction of EDS-CSNN, we enforce the independence of each subset by minimizing the overlapping region among the subsets for each classifier in the ensemble. This approach allows us to determine the ensemble size adaptively with a certain overlapping region threshold. Since ES-CSNN has a limit on the ensemble size, to have a fair comparison, we set the maximum of the ensemble size of diverse random subspace to the same limit, typically fixed at 50. To that end, we selected the first 50 from the *DiverSets* if the limit is exceeded. In diverse random subspace, the ratio parameters are under the default condition where the ratio of bootstrap sampling R_s is 0.7 and the ratio of features R_f is 0.5.

Here we vary the value of T_{over} to exploit the relationship between T_{over} and classification performance. The range of T_{over} is [0.2, 0.5], the step is 0.02. For each value of T_{over}, 10-fold cross validation is conducted to obtain the average value. For space considerations, we only show the results on Page dataset in **Figure 2**. **Figure 2** shows the performance of single ECSNN, ES-CSNN and EDS-CSNN with varying T_{over}.

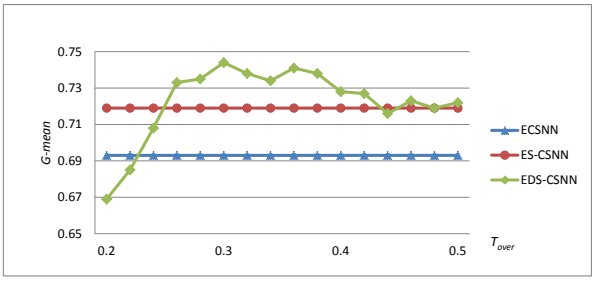

Figure 2. The classification performance when varying the threshold parameter T_{over} on Page dataset

From **Figure** 2, we can see the result of G-mean changes as we vary the value of T_{over} in EDS-CSNN. A

smaller T_{over} represents stronger diversity in the subsets. Increasing the value of T_{over} loosens the restriction of diversity. EDS-CSNN can outperform the single ECSNN and ES-CSNN when T_{over} gets to a certain value. We can see that the best overlapping parameter was 30% for Page dataset and the corresponding G-mean value is 0.744.

What should the value of T_{over} be? Clearly, this value should not be constant as the optimal value of T_{over} depends on the distribution of the dataset. It determines the diversity and number of components, so as to affect the final performance directly. Therefore, we have to estimate the optimal parameter for obtaining the best performance on each dataset. In order to estimate the optimal parameter T_{over} for obtaining the best performance, the threshold T_{over} is chosen by cross validation in the data set. In imbalanced data case, available data instances, mainly instances of the minority classes, are insufficient for traditional cross validation in the training set. For this reason, we randomly divided the original data set into two sets: the training set (80%) and the validation set (20%) for measuring the performance of each T_{over}. This process is repeated 10 times, and then an average G-mean result for each T_{over} is obtained. The output is a T_{over} which obtains the best G-mean value among all tests. We also optimize the optimal T_{over} for diverse random subspace framework methods combined with traditional neural network (NN) and CSNN with default cost setup separately. In the default CSNN, the misclassification cost for class C_i is set to $ImbaRatio_i$ which is the size ratio between the largest class and each class C_i. After obtaining the individual optimal T_{over} for diverse random subspace framework; we compare the performance of the three different types of neural network classifiers working on the single model, original random subspace with a fixed ensemble size, as well as our diverse random subspace separately. All the experiments are carried out by 10-fold cross-validation. For the diverse random subspace framework, the section of the 10 fold cross validation is totally independent from the one of cross validation for obtaining the optimal T_{over}.

It is clear from the experimental results in **Table 2** that EDS-CSNN obtained the best result on the majority datasets. This indicates that diverse random subspace ensemble with evolutionary search optimizing is a very effective strategy for cost sensitive neural network. Diverse random subspace method emphasizes ensemble diversity explicitly during training, so as to enhance the learning of neural network and optimization of cost parameters. Moreover, each CSNN constructed in the individual subset under different subspaces can find potentially interesting local data characteristic and property. Especially for the datasets with high dimensional feature such as Annealing and Satimage, EDS-CSNN offers a great advantage over other solutions. Furthermore, the diverse subset construction can achieve better performance than the complete ensemble on the imbalanced data. The result indicates that the diversity in the ensemble can facilitate class imbalance learning. However, diverse random subspace ensemble cannot achieve the best result on some low dimensional datasets, such as Balance and New Thyroid which have 4 and 5 attributes respectively. That is because the random subspace method is more effective when datasets have a large numbers attributes. Note that even the original random subspace was not the winner on these small dimensional datasets. With a very small number of attributes, each classifier receives a small set of features and thus is weak.

Furthermore, regardless of the single or ensemble model, the artificial bee colony optimization improves the performance of traditional cost sensitive neural network. It demonstrates the misclassification cost based on the prior distributions between two classes is not appropriate, resulting in obtaining an unexpected performance. The optimization method can achieve the optimum misclassification cost under the guidance of G-mean, so as to achieve the best performance. **Table 2** also lists the optimal T_{over} value and the corresponding ensemble size of EDS-CSNN. We found that the size is smaller than 50 on the majority cases. The results reveal that diverse random subspace can generate an ensemble model with smaller sizes but stronger generalization ability.

D. Experiment II: Comparing EDS-CSNN with the-state-of-art methods for imbalanced data learning

After finding out that EDS-CSNN method can improve the classification performance of neural network, we empirically assessed the algorithm against the state-of-the art methods for multiclass imbalanced data learning, such as Editing Nearest Neighbour rule under-sampling (ENN) [7], SMOTEBoost (SMB) [5], Random subspace method combined with SMOTE (SM-RSM) [6], MetaCost (MC) [10] and AdaBoost.NC combined with random over-sampling (ANCOS) [11]. ENN does not require a user specified under-sampling ratio and K is set to 3. For SMB and SM-RSM, the nearest neighbor parameter k is set to 5, the most accepted value in the literature. For the over-sampling methods including SMB, SM-RSM and ROS, the amount of new data for a class C_i is set to be the size difference between the largest class and class C_i. In the setting of MC, the misclassification cost for class C_i is set to $ImbaRatio_i$ which is the size ratio between the largest class and each class C_i. ANCOS approach utilized AdaBoost.NC [22] that combines the strength of negative correlation learning and random over-sampling to address the multiclass imbalanced data classification. The penalty strength parameter in AdaBoost.NC is set 9. The sizes of components are 50 in the all ensemble classifiers. **Table 3** summarizes the performance of the compared algorithms, in which the best performance for each dataset is

highlighted. It is evident from **Table 3** that EDS-CSNN outperforms the current re-sampling and cost-sensitive learning strategies on 8 of 12 datasets.

ENN is the worst method since it is hard to identify the noise when the distribution is complex and imbalanced. Some border points may also be removed as noise while they are useful for training, resulting in loss of information. Both SMB and SM-RSM benefit from the diversity of the ensemble framework. However they manipulate the instances blindly without taking into the majority class consideration, resulting in generating noise instance. The diversity characteristic of ANCOS can improve the generalization performance, but random over-sampling may result in overfitting for minority class. MetaCost performs slightly worse than the other advanced methods. It may be because the ratio misclassification cost based on the size ratio between two classes is not appropriate, which reveals again that the misclassification cost is vital for cost sensitive learning, and needs to be searched by some heuristic methods.

We seek to determine whether our approaches significantly outperform the state-of-the art methods. In order to evaluate the significance of the results, we performed a statistical analysis of our results. Following Demsar's recommendation in [23], we concluded that there is a significant difference among the methods by applying Friedman test. Since the null hypothesis is rejected we have to proceed with further analysis to better understand the behavior of the classification algorithms. We performed a series of Wilcoxon tests and provide the *T* value for our EDS-CSNN against all contenders in **Table 3**. Since there are 12 datasets, *T* should be less than or equal to 13 to reject a null hypothesis in the significance level of 0.05, according to the critical value table. One can conclude that EDS-CSNN statistically outperform the other state-of-the-art methods for imbalanced data methods at a significance level of 0.05.

Table 2. The comparative results of the neural network methods in terms of G-mean on the multiclass class

Datasets	**Single**			**Random subspace**			**Diverse random subspace**				
	NN	**CSNN**	**ECSNN**	**NN**	**CSNN**	**ECSNN**	**NN**	**CSNN**	**ECSNN**		
	G-mean			**G-mean**			**G-mean**		**G-mean**	T_{over}	**size**
Cmc	0.714	0.745	0.723	0.738	0.763	0.744	0.738	0.764	**0.783**	0.44	36
Balance	0	0.214	**0.557**	0	0.232	0.528	0	0.226	0.532	0.32	29
New Thyroid	0.875	0.921	**0.946**	0.889	0.919	0.902	0.887	0.911	0.908	0.22	21
Car	0.752	0.815	0.837	0.766	0.827	0.853	0.807	0.834	**0.879**	0.36	35
Annealing	0.921	0.930	0.928	0.939	0.945	0.948	0.933	0.947	**0.961**	0.5	50
Nursery	0.462	0.637	0.769	0.557	0.644	0.781	0.581	0.639	**0.798**	0.3	29
Page	0.654	0.693	0.703	0.659	0.701	0.722	0.688	0.718	**0.741**	0.3	31
Ecoli	0.816	0.822	0.846	0.831	0.834	**0.867**	0.829	0.834	0.863	0.26	24
Cleveland	0.171	0.186	0.221	0.164	0.191	0.235	0.168	0.197	**0.264**	0.28	43
Glass	0.374	0.424	0.535	0.429	0.438	**0.589**	0.426	0.440	0.578	0.28	28
Satimage	0.786	0.806	0.821	0.827	0.842	0.878	0.833	0.842	**0.894**	0.5	50
Yeast	0	0.232	0.278	0	0.323	0.385	0.254	0	**0.406**	0.38	36

Table 3. G-mean on 12 UCI datasets for several classification methods

Datasets	**MC**	**ANCOS**	**ENN**	**SMB**	**SM-RSM**	**EDS-CSNN**
Cmc	0.685	0.744	0.719	0.749	0.742	**0.783**
Balance	0.348	0.516	0	**0.542**	0.514	0.532
New Thyroid	0.921	**0.929**	0.890	0.922	0.905	0.908
Car	0.821	0.858	0.579	0.848	0.854	**0.879**
Annealing	0.928	0.944	0.928	0.931	0.954	**0.961**
Nursery	0.759	0.808	0.758	**0.811**	0.802	0.798
Page	0.737	0.736	0.684	0.739	**0.741**	**0.741**
Ecoli	0.842	0.847	0.786	**0.871**	0.839	0.863
Cleveland	0.204	0.249	0.027	0.142	0.075	**0.264**
Glass	0.515	0.554	0.425	0.557	0.561	**0.578**
Satimage	0.834	0.865	0.825	0.841	0.864	**0.894**
Yeast	0.138	0.237	0	0.327	0.263	**0.406**
T	1	8	0	10	2	—

E. Experiment III: Evaluating the diverse random subspace ensemble on binary class

In the binary class, there is only one parameter (C_{rf}) need to be optimized. Threshold searching can obtain the optimal C_{rf} for CSNN. We choose five binary datasets to evaluate the performance of the diverse random subspace ensemble with **T**hreshold moving CSNN (TDS-CSNN). Detailed information of the datasets can be found in **Table 4**. From **Table 5** it is apparent that TDS-CSNN obtains the best G-mean as compared to the other considered algorithms on all the datasets except Breast Cancer. It demonstrates that the characteristic of the diversity in diverse random subspace can help in improving the searching of cost parameter and generalization performance for CSNN on the binary class datasets.

Table 4. The binary class datasets used for experimentation

Dataset	*C*	*Inst.*	*F*	*Class distribution*
German	2	1000	24	300/700
Pima	2	768	8	268/500
Sick	2	3772	29	231/3541
Spambase	2	4601	57	1813/2788
Breast Cancer	2	699	9	458/241

Table 5. The G-mean results for several classification methods on the binary class imbalanced datasets

Datasets	MC	ANCOS	ENN	SMB	SM-RSM	TDS-CSNN
German	0.675	0.775	0.651	0.747	0.758	**0.796**
Pima	0.759	0.782	0.755	0.782	0.731	**0.798**
Sick	0.876	0.903	0.844	0.885	0.889	**0.912**
Spambase	0.837	0.861	0.825	0.843	0.855	**0.863**
Breast Cancer	0.957	**0.978**	0.926	0.953	0.972	0.968

V. CONCLUSIONS

In practice, many problem domains have more than two classes with uneven distributions, but there are fewer solutions in multiclass imbalance problems. In this paper, we present EDS framework for cost sensitive neural network in order to advance the classification of multiclass imbalanced data. The key characteristics of EDS are cost searching and ensemble learning. In EDS, artificial Bee Colony algorithm is employed to search the optimal misclassification cost parameters from the available data; and diverse random subspace offers a good framework for imbalanced data learning as it produces varied and complementary base classifiers by explicitly encouraging the diversity of subsets used by each classifier. The proposed method provides an effective solution to deal with the multi-class imbalance problems. The experimental results show that the improved algorithm significantly outperforms existing methods over a variety of datasets. It also demonstrates the optimization of cost setup and ensemble learning are two critical factors to improve the cost sensitive learning.

As a new method for imbalanced learning problems, there are several interesting future research directions for EDS framework. First, we will apply strategy of EDS to other existing cost sensitive classifiers with different fitness functions. Second, we fixed the two ratios of sampling R_s and R_f according to the default settings of bootstrap and random subspace ensemble in our work. We would like to automatically find the optimal value for these parameters based on a given dataset.

REFERENCES

[1] N.V. Chawla, N. Japkowicz, A. Kolcz, "Editorial: special issue on learning from imbalanced data sets," SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets 6 (1):1-6, 2004.

[2] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, "Handling imbalanced datasets: A review," GESTS International Transactions on Computer Science and Engineering 30(1):25-36, 2006.

[3] Q. Yang, X. Wu, "10 challenging problems in data mining research," Int'l J. Information Technology and Decision Making 5(4):597–604, 2006.

[4] H. He, E.A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, 21(9):1263-1284, 2009.

[5] N.V. Chawla, A. Lazarevic, L. Hall, K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," Proc. of 7th European Conf. Principles and Practice of Knowledge Discovery in Databases, pp. 107-119, 2003.

[6] T. Hoens, N.V. Chawla, "Generating Diverse Ensembles to Counter the Problem of Class Imbalance," Proc. of 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 488-499, 2010.

[7] D.L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," IEEE Transactions on Systems, Man and Cybernetics, (3), pp. 408-421, 1972.

[8] V.S. Sheng, C.X. Ling, "Thresholding for making classifiers cost-sensitive," Proc. of 21th National Conference on Artificial Intelligence, pp. 476-481, 2006.

[9] Z. H. Zhou, X. Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," IEEE Transactions on Knowledge and Data Engineering, 18(1), 63-77, 2006.

[10] P. Domingos, "MetaCost: A general method for making classifiers cost-sensitive," Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 155–164, 1999.

[11] S. Wang, X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 42, pp. 1119-1130, 2012.

[12] M. Kubat, R. Holte, S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," Machine Learning, 30, pp.195–215, 1998.

[13] Z.H. Zhou, J. Wu, W. Tang, "Ensembling neural networks: many could be better than all," Artificial intelligence, 137(1), pp. 239-263, 2002.

[14] T. Ho, "The random subspace method for constructing decision forests," Pattern Analysis and Machine Intelligence 20 (8): 832–844, 1998.

[15] C. Elkan, "The Foundations of Cost-Sensitive Learning," Proc. of 17th International Joint Conference of Artificial Intelligence, pp.973-978, 2001.

[16] F. Provost, "Machine learning from imbalanced data sets 101," Proc. of AAAI Workshop on Imbalanced Data, 2000.

[17] D. Martens, B. Baesens, T. Fawcett T, "Editorial Survey: Swarm Intelligence for Data Mining," Machine Learning, 82(1):1-42, 2011.

[18] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm," Journal of Global Optimization 39, pp. 459–471, 2007.

[19] D. Karaboga, B. Akay, "A comparative study of artificial bee colony algorithm," Applied Mathematics and Computation, 214, pp.108–132, 2009.

[20] V.H. Jason, T.M. Khoshgoftaar, A. Napolitano, R. Wald, "Feature selection with high-dimensional imbalanced data," Proc. of IEEE International Conference on Data Mining Workshops:507-514 2009.

[21] B. Rok, L. Lara, "Class prediction for high-dimensional class-imbalanced data," BMC bioinformatics 11, 2010.

[22] S. Wang, H. Chen, X. Yao, "Negative Correlation Learning for Classification Ensembles," Proc. of International Joint Conference on Neural Networks (IJCNN), pp. 2893-2900, 2010.

[23] J. Demsar, "Statistical comparisons of classifiers over multiple datasets," Journal of Machine Learning Research, 7(1):1-30, 2006.