ABC-Sampling for balancing imbalanced datasets based on Artificial Bee Colony algorithm

Ali Braytee, Farookh Khadeer Hussain, Ali Anaissi and Paul J. Kennedy
School of Software
Center Quantum Computation and Intelligent Systems
University of Technology Sydney
Sydney, New South Wales 2007
Emails: {ali.braytee, farookh.hussain, ali.anaissi and paul.kennedy}@uts.edu.au

***Abstract*—Class imbalanced data is a common problem for predictive modelling in domains such as bioinformatics. It occurs when the distribution of classes is not uniform among samples and results in a biased prediction of learning towards majority classes. In this study, we propose the ABC-Sampling algorithm based on a swarm optimization method called Artificial Bee Colony, which models the natural foraging behaviour of honeybees. Our algorithm lessens the effects of imbalanced classes by selecting the most informative majority samples using a forward search and storing them in a ranked subset. Then we construct a balanced dataset with a planned undersampling strategy to extract the most frequent majority samples from the top ranked subset and combine them with all minority samples. Our algorithm is superior to a state-of-the-art method on nine benchmark datasets with various levels of imbalance ratios.**

I. Introduction

Recent advances in technology have led to exponential increases in data, from daily business transactions, social media, and microarray data. There is an urgent need for data mining for analysis, capture, search and visualization of data. A serious challenge to data mining in several domains is the imbalanced learning problem, which is caused by unequal distributions of data between classes [4]. Imbalance occurs due to a paucity of cases, for example, patients with a rare disease [15], or difficulties in collecting samples due to high cost or privacy. The class imbalance problem affects classification accuracy because standard learning algorithms tend to predict the dominant class in order to minimize the error. Many standard learning algorithms assume balanced datasets, but this does not necessarily match real-world experience, which can lead to poor performance in practice.

Our Approach. This paper proposes and evaluates a data-independent algorithm. It takes the undersampling approach and is based on artificial bee colony (ABC) [8], a robust optimization method that is part of the swarm intelligence family and simulates foraging behaviour of honeybees. Our ABC-Sampling algorithm classifies imbalanced data by identifying the most informative majority examples. The output is an optimal balanced dataset. Our technique as described in Algorithm 1 has five stages: 1) the dataset is divided into training and testing sets to avoid overfitting; 2) artificial bees leave the hive and initiate "food sources", which represent the selected informative samples from the majority class; 3) food sources, as possible solutions, are sent to a classifier and accuracy is computed; 4) the local optimum is evaluated by choosing the top food source based on fitness value in each iteration; and 5) the global optimum samples are identified by selecting the highest frequency samples. Our method and a state-of-the-art method are evaluated by training a Support Vector Machine (SVM) classifier [13] on the retrieved balanced dataset and evaluating on the test set.

Contributions. Our contributions are:

- We develop an efficient and effective undersampling algorithm to balance the imbalanced datasets based on Artificial Bee Colony optimization method.
- Our approach addresses issues of scalability by evaluating large and highly imbalanced datasets in reasonable computational time.
- We select the optimal parameter values of the algorithm based on experimental results.
- We conduct both theoretical and empirical studies on several binary cases of dataset with different degrees of imbalance and size. Extensive performance studies demonstrate that the proposed algorithm significantly outperforms the state-of-the-art method.

Organization. This rest of this paper is organised as follows. Section II describes the method. Section III presents the experimental results before concluding in Section IV.

Related work. There are two main strategies for tackling the imbalanced learning problem: sampling and cost-sensitive learning. The former modifies the distribution of the imbalanced dataset to make it more balanced. The latter increases the cost of misclassified minority examples and decreases the cost of misclassified majority examples [5]. The sampling approaches can be divided into under- and over-sampling techniques. Undersampling reduces the number of majority samples, whilst over-sampling duplicates minority examples. Drummond et al. [6] shows that undersampling is effective. However, it may exclude important information by removing useful majority samples. Furthermore, random removal of samples, where there are already low numbers of samples is inappropriate in some real applications [17]. Over-sampling replicates minority samples, so it does not add any extra information to the training set. Also, it does not lose information since all majority samples are retained [11]. However, it is more prone to over-fitting because multiple copies of examples due to replication can lead to increased

variance of models [10]. Cost-sensitive learning increases the cost of misclassified examples [5] and minimizes total cost rather than error rate. However, it can be difficult to design a suitable cost function [19]. In summary, multiple strategies have been proposed to solve the imbalanced class problem with accompanying debates on the merits of strategies [2]. Undersampling is often preferred because no extra information is added to the data, but there is no consensus in the literature on an optimal strategy.

Swarm intelligence algorithms simulate the behaviour of natural biological systems. They are composed of a population of single agents that interact with each other. They solve optimization problems by simulating the global behaviour of the internal agents [3]. Yang et al. [16] and Yu et al. [18] applied particle swarm optimization (PSO) and ant colony optimization (ACO) respectively to optimize imbalanced datasets. The common problems of both algorithms are that they need to tune many parameters before and during their execution and they set a random values of parameters without justifying the reasons. Also, both studies do not evaluate the time complexity of their proposed algorithms. Moreover, they do not evaluate their algorithms on large datasets to show the scalability: Yang et al. [16] applied his algorithm on hundreds of samples and Yu et al. [18] to tens of samples only. Schiezaro and Pedrini [12] applied ABC optimization for feature selection, but we believe we are the first to balance imbalanced data with an ABC-based algorithm. Moreover, it can easily be adapted to Oversampling.

The main advantages of Artificial Bee Colony algorithm (ABC) over the other heuristic search algorithms are listed below. ABC has fewer parameters compared to other heuristic search techniques such as genetic algorithm (GA), particle swarm optimization (PSO) and differential evolution (DE). Apart from the maximum number of evaluations and population size, ABC has only one control parameter (limit). But, PSO and GA have three control parameters cognitive factors, social factors, and inertia weight; and crossover rate, mutation rate, and generation gap respectively. Also, DE has at least two parameters (crossover and scaling factor). Moreover, ABC addresses diversity in the search space better than GA and DE [9]. GA and DE use a mutation operator that randomly modifies a part of the current solution. However, ABC balances between the local search process and the global search process. It uses a similar process as in GA and DE to slightly modify a part of the current solution that is useful for local search and the speed of convergence.

On the other hand, it removes the whole solution and generates a new random solution by a scout bee. This mechanism enhances the diversity in solutions of ABC, and it avoids premature convergence in the search. Finally, ABC may replace the global best solution if it reaches the maximum exploitation limit by a new random solution. But in the case of PSO, GA and DE, they keep the best solutions in the population which always contribute in producing new solutions [9].

II. Methods

First we describe the standard artificial bee colony algorithm followed by our proposed variation for undersampling.

A. Artificial bee colony (ABC) algorithm

Artificial Bee Colony (ABC) was proposed for solving optimization problems. It is based on the foraging behaviour of real honeybees, which are classified into two kinds: employed and unemployed foragers. Employed bees exploit food sources and bring information about nectar to the hive to communicate with unemployed bees. Unemployed bees are of two types: onlookers, that wait in the hive for shared nectar information; and scouts, that search for new food nearby [7]. When employed bees bring nectar information from the food source to the hive, they aim to communicate with onlookers so as to choose the best quality nectar among food sources. Bee communication occurs through dance with the direction and duration related to distance and direction of the food. Loaded nectar refers to the food quality. Onlooker bees watch the employed bees to choose the best food source based on nectar quality. Food sources are abandoned once employed bees have fully exploited them, at which time the employed bees become scouts.

In ABC, food sources represent potential problem solutions with each initially exploited by one artificial bee. The nectar information in the food source is the value of the solution. The bee colony is divided in half between employed and onlooker bees. Abandoned food sources are identified after several iterations, i.e. when search does not find better quality neighbouring solutions. Employed bees become scouts in abandoned food sources and start to search for new solutions. The original algorithm has the following fours stages:

a) Initialisation: ABC starts by randomly creating N food sources (i.e. potential solutions) with food source i characterised by a vector $FS_i \in \mathbb{R}^D$ with D, the problem dimensionality. FS_{ij}^{min} and FS_{ij}^{max} define the minimum and maximum values of the jth element of FS_i. Food sources are initialised as

$$FS_{ij}^{'} = FS_{ij} + r(FS_i^{max} - FS_j^{min}) \tag{1}$$

where $rand(0, 1)$ is uniform random number in [0, 1].

b) Employed bee stage: This stage associates a single employed bee with food source i and finds the neighbourhood of the food source (i.e. other potential solutions) using

$$FS_{ij}^{'} = FS_{ij} + \varphi_{ij}(FS_{ij} - FS_{kj}) \tag{2}$$

where $\varphi \in [-1, 1]$ is a uniform random number, i indexes the N food sources, j indexes the elements of FS_i and $k \neq i$ is a uniform random FS other than i. Fitness of FS_i is

$$fitness_i = \begin{cases} (1 + f_i)^{-1}, & \text{if } f_i \geqslant 0 \\ 1 - f_i, & \text{if } f_i < 0 \end{cases} \tag{3}$$

where f_i is the cost (objective) function being maximised.

c) Onlooker stage: Employed bees share their fitness with onlookers who select the maximum fitness among all food sources. The probability of selecting food source i is

$$p_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i} \tag{4}$$

d) Scout stage: Scout bees explore new food sources. Each employed bee has a limit on the number of times it can use the same food source. The count increments if the

fitness value of the current food source is better than the neighbourhood one. Employed bees convert to scouts upon reaching the limit.

B. Sampling strategy

Based on this algorithm, the proposed ABC-Sampling divides into three components. The imbalanced dataset is partitioned into training and test sets in the ratio of 2:1, although the algorithm does not depend on this ratio. Training data is used by the sampling algorithm. Test data is held for evaluation. Finally, ABC-Sampling is applied as below.

C. ABC-Sampling algorithm

In ABC-Sampling, potential solutions are represented by a bit vector. Each food source randomly initiates a bit vector of size D, the number of samples from the majority class in the training set. The bit value in the vector represents the presence or absence of the majority sample in the subset. The algorithm has the following stages.

1. Create M food sources and initialize parameters: Value M is half of the samples in the training set. Randomly initialize the D bit values for each food source. Initialize the maximum number of iterations of the algorithm and the limit counter for abandoning food sources.

2. Compute fitness: Submit the food source samples to a classifier and evaluate accuracy as the fitness value. In this paper, we chose SVM to evaluate the AUC and F-measure accuracy of data points, but other approaches are reasonable [14]. For very high dimensional datasets, we applied feature selection using Balanced Iterative Random Forest [1]. For small datasets, we evaluated AUC and F-measure using leave one out cross-validation technique and for larger datasets we used three-fold cross-validation.

3. Find neighbouring food and compute their fitness: Determine the neighbouring food sources with (2). As the food source values are represented by bits and the perturbation frequency φ is real, the real-valued neighbour values are converted to bits through a sigmoid function. Compute fitness of neighbour samples as in step 2. If the fitness is higher than the current food source, replace the neighbour samples to the food source set. Otherwise, the limit variable for the food source is incremented. If this reaches the maximum, the algorithm eliminates the current food source and increments the abandoned variable to create a new food source by scout bees.

4. Share results and record the best: The employed bees share the resulting fitness values with onlookers, the onlookers select the top average of AUC and F-measure fitness values from food sources and they distribute and re-execute step 3. Select the maximum fitness value of onlookers and save as the global best value in best majority samples set.

5. Launch scout bees: Scout bees set is the number of the exhausted food sources that need to be abandoned. The food source has a parameter counter $LIMIT$, which has been updated during the search. If the value of this variable reaches the maximum limit ($LIMIT_{MAX}$), then the food source will be abandoned and replaced by a new food source that produced by a new scout bee. Scout bees are created by (1) and add it to the original set of food sources.

6. Test for termination: Check if the maximum number of iterations is reached and if so return the best informative samples set. Finally, the algorithm applies undersampling strategy to produce a balanced dataset by selecting a matching number of the top majority samples to the number of minority ones.

Input: training set
Output: $Rankset$ of majority samples
Initialization;
$FoodSourceSize$=$TrainingSamples/2$, FS_Set, $MaxIterations$, $ProbabilitySet$, $LIMIT_{MAX}$, $Abandoned$
for $i = 1 : FoodSourceSize$ **do**
  Initialize position of FS_Set_{ij} by (1) for $j \in [0, 1]$
end
for ***each*** FS *in* FS_Set **do**
  Create an internal set $FS \bigcup minoritySet$;
  Train classifier and evaluate the fitness function;
  Save the fitness value in FS_Acc;
end
for $t = 1 : MaxIterations$ **do**
  for ***each*** FS *in* FS_Set **do**
    FS_Set=ProcessNeighbours (FS,FS_Set)
  end
  Store best local food source ;
  Calculate Probabilities $ProbabilitySet$ using (4) ;
  while $t < FoodSourceSize$ **do**
    if $Random < ProbabilitySet$ **then**
      FS_Set=ProcessNeighbours ($FS\_Set[t]$,FS_Set) ;
    end
  end
  Memorize best global food source ;
  $Global = \max FS\_Acc$;
  $Best\_FS = FS\_Set[Global]$;
  $Rankset = Rankset \bigcup Best\_FS$;
  for $v = 1 : Abandoned$ **do**
    Initialize position of FS_Set_{ij} by (1) for $j \in [0, 1]$
  end
  Find abandoned food sources;
end
Return $Rankset$ of majority samples;

Algorithm 1: ABC-Sampling algorithm

III. Experimental results

A. Datasets

We evaluated on nine imbalanced datasets each with two classes (see Table I). Apart from "childhood leukaemia", these datasets are obtained from UCI Machine Learning Repository[1]. "Childhood leukaemia" is available from the Oncogenomics Section of the Paediatric Oncology Branch at the National Cancer Institute NIH, USA[2]. As the first three datasets, "childhood leukaemia", "Colon" and "Breast", are small and high dimensional, we used leave-one-out cross-validation. The others

[1] http://archive.ics.uci.edu/ml
[2] http://pob.abcc.ncifcrf.gov/cgibin/JK

Function *ProcessNeighbours* ($currentFS, fsSet$)
  Determine the $neighbour$ by (2);
  if $neighbour <> currentFS$ **then**
    Evaluate the fitness function;
    if $currentFS\_Acc < neighbour\_Acc$ **then**
      Replace $currentFS$ with $neighbour$;
    else
      Increment the $LIMIT$ of $currentFS$;
      if $LIMIT \geq LIMIT_{MAX}$ **then**
        Increment the $Abandoned$;
        $fsSet = fsSet - currentFS$;
      end
    end
  end
  Return $fsSet$;

Algorithm 2: Generate and evaluate neighbours

used 3-fold cross validation to reduce computation time and the variance of estimators. Affymetrix childhood leukaemia dataset was generated from U133A platform and collected by The Children's Hospital at Westmead. "Colon" and "Breast" are other Affymetrix microarray dataset from UCI ML Repository. "Blood" was generated by Blood Transfusion Service Center in Taiwan. "Survival" was produced from the survey conducted on the survival of patients who had undergone surgery for breast cancer. "Diabetes" is generated from the study of diabetes in Pima Indian population. "SpamBase" dataset is a large collection of spam and non-spam emails which is collected from postmaster and personal emails. "Australian credit approval" and "Ionosphere" are not highly imbalanced, so we sampled them at 1:5, 1:10 and 1:15, to produce more imbalance.

TABLE I: Evaluated datasets

Dataset	Size	Attributes	Imbalance ratio
Childhood leukaemia	60	22277	1.85
Colon	62	2000	1.82
Breast	77	4869	1.34
Blood	748	5	3.16
Survival	306	3	2.77
Diabetes	768	8	1.86
SpamBase	4601	57	1.537
Australian Credit Approval	414	14	4.97, 10, 14.73
Ionosphere	247	34	5, 9, 15

B. Evaluation and parameter selection

Feature selection is applied to high dimensional biomedical datasets in order to reduce the number of dimensions and to select the most informative features. We use a feature selection technique called Balanced Iterative Random Forest (BIRF) [1]. This technique robustly selects a small number of informative genes. The results have been validated on several training sets to ensure that the genes are globally selected. Evaluation metrics are vital in measuring learning performance. Learning from imbalanced datasets requires the use of Area Under the ROC Curve (AUC), F-measure or similar as they do not assume similar sizes of each class. ABC-Sampling algorithm defines parameters based on parameter selection experiments in Table IV and Figure 1. ABC-Sampling is compared to learning from imbalanced datasets, random undersampling and particle swarm optimization (PSO) [16]. PSO technique searches for an optimal subset of majority samples and combines them with the minority samples for building a balanced classification model. Random undersampling reduces the number of majority samples by selecting a random subset of the majority class equal to the number of minority samples and combines both in the training dataset [17].

C. Results

Performance of ABC-Sampling. In this section, two experiments are conducted. We applied ABC-Sampling on the six biomedical datasets that are either high or low dimensional data. Results obtained are measured on independent test sets to accurately report the generalization of our algorithm. Results are compared to three state-of-the-art methods: PSO [16], random undersampling (labelled "RU") and learning from the original imbalanced dataset ("Baseline"). As shown in Table II, the results demonstrate the superiority of our proposed method "ABC-Sampling" over the state-of-the-art undersampling techniques using two different measure metrics AUC and F-measure. Also, the results show the importance of undersampling methods in improving the classification performance over the evaluated datasets. They attain the highest values compared to classifying the original imbalanced datasets. The second experiment looks at "Australian credit approval" and "Ionosphere" datasets that are artificially imbalanced in different ratios as shown in Table III. Also, it looks at a large dataset "SpamBase". ABC-Sampling outperformed the other methods over all tested levels of imbalance. ABC-Sampling is a strong alternative to existing methods for balancing datasets and leads to excellent learning outcomes. Furthermore, it shows its scalability by achieving a good results on large and highly imbalanced datasets.

Parameter sensitivity. We investigate the parameter sensitivity of our algorithm. ABC optimization algorithm has fewer control parameters compared to other heuristic search algorithms such as PSO and ACO. Apart from the maximum number of iterations, ABC has only one control parameter ($LIMIT_{MAX}$). Therefore, it is less sensitive to the change of parameters. The $LIMIT_{MAX}$ parameter stops the algorithm from being dragged into local optima. As shown in Table IV, the optimal value of parameter $LIMIT_{MAX}$ is 10 based on the accuracy measure of the evaluated datasets. Furthermore, we observed that as the parameter $LIMIT_{MAX}$ increases, the runtime of the algorithm increases and the accuracy of the major cases is decreased. Also, if the value of $LIMIT_{MAX}$ is less than 10, the accuracy is bad due to the early convergence into the local optimum.

We also analysed $MaxIterations$, which specifies the maximum number of iterations of the algorithm. It may affect the accuracy of the classifier. We evaluated the benchmark datasets on different number of iterations to choose the optimal value based on two factors: the highest accuracy and the smallest number of iterations. As shown in Figure 1, iteration 250 attains the highest accuracy compared to the other iteration

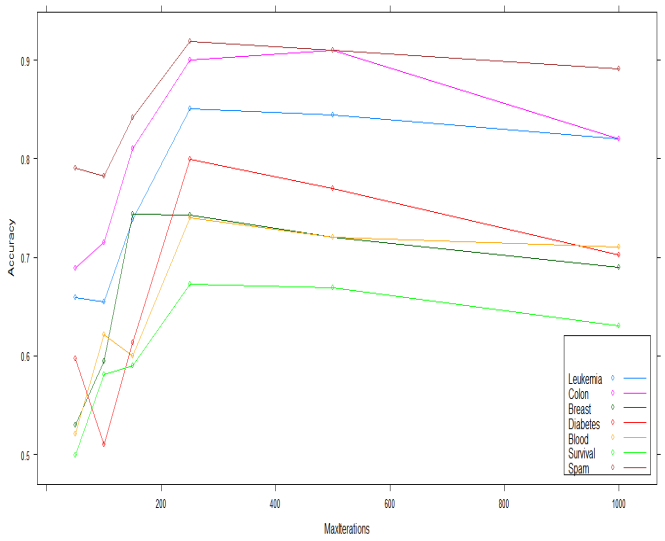

Fig. 1: ABC-Sampling parameter selection ($MaxIterations$)

values. The plots at iteration 500 have similar performance to iteration 250, but using more iterations until termination will increase the time computation of the algorithm.

Time complexity. Heuristic search algorithms may not present better results compared to brute force search methods. However, the time complexity to evaluate all the subsets of the brute force search is O(2^n) where n is the number of data points, which is practically impossible. Therefore, we analysed the time complexity of our algorithm in terms of the number of fitness function evaluations.

The worst case scenario of evaluating ABC-Sampling algorithm is described below. In ABC-Sampling, there is more than one fitness function evaluation for each food source during an iteration. During the initialization and employed stages, the fitness function must be calculated for the whole population size N of food sources. By the same token, during the onlooker stage, it needs to evaluate N fitness functions for top food sources. Therefore, the overall number of fitness function evaluations in the initialization stage is N and in the employed and onlooker stages is $2TN$ where T is the number of iterations. In the scout bees stage, our algorithm selects the food sources that exceed $LIMIT_{MAX}$ trials (labelled as t) to abandon and replace its new food source. Hence the individual needs $t/2$ iterations to exceed the $LIMIT_{MAX}$ trials. As a result, the worst case scenario of evaluating the fitness function in ABC-Sampling algorithm is:

$$N(1 + 3T - \frac{t}{2}) \tag{5}$$

ABC-Sampling has a lower complexity than the standard ABC algorithm due to avoiding execution of the fitness function when the neighbour and current food sources are identical as shown in algorithm 1.

IV. Conclusions

This paper presents ABC-Sampling algorithm based on undersampling strategy and the Artificial Bee Colony optimisation approach. The aim of our method is to balance highly imbalanced datasets to enhance learning. It identifies the most informative majority samples and does not lead to overfitting. We show that ABC-Sampling performs better than state-of-the-art method, it is independent of specific datasets and scales with different levels of imbalance and size of dataset.

As ongoing work, we intend to improve ABC-Sampling algorithm by reducing the execution time of the algorithm, which can be achieved by parallelising computation of fitness for employed bees. Also, we plan to use this method to select the most informative minority samples in order to create a balanced dataset based on planned oversampling strategy.

References

[1] A. Anaissi, P. J. Kennedy, et al. A balanced iterative random forest for gene selection from microarray data. *BMC Bioinformatics*, 14(1):261, 2013.

[2] G. Batista, R. C. Prati, et al. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, 1999.

[4] G. Cohen, M. Hilario, et al. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7–18, 2006.

[5] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 155–164. ACM, 1999.

[6] C. Drummond, R. C. Holte, et al. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Proc. ICML'03 Workshop on Learning from Imbalanced Data Sets*. ACM, 2003.

[7] R. L. Jeanne. The evolution of the organization of work in social insects. *Monitore Zoologico Italiano-Italian J. of Zoology*, 20(2):119–133, 1986.

[8] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[9] D. Karaboga and B. Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied soft computing*, 8(1):687–697, 2008.

[10] M. Kubat, S. Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML*, volume 97, pages 179–186, 1997.

[11] M. M. Rahman and D. Davis. Addressing the class imbalance problem in medical datasets. *Int'l J. of Machine Learning and Computing*, 3(2):224–228, 2013.

[12] M. Schiezaro and H. Pedrini. Data feature selection based on Artificial Bee Colony algorithm. *EURASIP J. on Image & Video Processing*, 2013(1):1–8, 2013.

[13] V. N. Vapnik. *Statistical learning theory*. Wiley, 1998.

[14] G. M. Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19, 2004.

[15] K. S. Woods, C. C. Doss, et al. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 7(06):1417–1436, 1993.

[16] P. Yang, L. Xu, et al. A particle swarm based hybrid system for imbalanced medical data sampling. *BMC Genomics*, 10(Suppl 3):S34, 2009.

[17] S.-J. Yen and Y.-S. Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009.

[18] H. Yu, J. Ni, and J. Zhao. ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. *Neurocomputing*, 101:309–318, 2013.

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

TABLE II: Evaluation results of seven methods on imbalanced datasets. Values are on the test sets.

Methods	Imbalanced Datasets						
	Childhood Leukaemia	Colon	Breast	Diabetes	Blood	Survival	SpamBase
ABC-Sampling							
AUC	**0.851**	**0.9**	**0.743**	**0.799**	**0.74**	**0.73**	**0.919**
F-measure	**0.755**	**0.813**	**0.698**	**0.743**	**0.74**	**0.681**	**0.918**
PSO							
AUC	0.812	0.891	0.732	0.791	0.721	0.691	0.892
F-measure	0.698	0.793	0.621	0.641	0.489	0.577	0.864
RU							
AUC	0.724	0.882	0.72	0.751	0.668	0.631	0.892
F-measure	0.611	0.774	0.613	0.612	0.471	0.481	0.886
Baseline							
AUC	0.792	0.81	0.583	0.621	0.632	0.592	0.77
F-measure	0.628	0.76	0.498	0.58	0.42	0.432	0.73

TABLE III: Classification performance on imbalanced datasets. Values are on the test sets.

	Aus. Credit Approval				Ionosphere			
Ratio	ABC-Sampling	PSO	RU	Baseline	ABC-Sampling	PSO	RU	Baseline
1:5								
AUC	**0.87**	0.821	0.53	0.69	**0.84**	0.712	0.58	0.71
F-measure	**0.853**	0.781	0.44	0.58	**0.829**	0.691	0.44	0.66
1:10								
AUC	**0.86**	0.813	0.45	0.673	**0.71**	0.692	0.46	0.61
F-measure	**0.84**	0.798	0.39	0.591	**0.703**	0.61	0.42	0.587
1:15								
AUC	**0.763**	0.71	0.35	0.615	**0.73**	0.683	0.391	0.62
F-measure	**0.732**	0.681	0.332	0.592	**0.719**	0.642	0.378	0.612

TABLE IV: ABC-Sampling parameter selection.

Methods	$LIMIT_{MAX}$ value					
	5	10	15	20	30	50
Childhood Leukaemia						
AUC	0.615	**0.851**	0.78	0.771	0.851	0.719
F-measure	0.595	**0.755**	0.71	0.677	0.745	0.628
Colon						
AUC	0.63	**0.9**	0.79	0.891	0.8	0.713
F-measure	0.593	**0.813**	0.72	0.691	0.71	0.66
Breast						
AUC	0.512	**0.743**	0.69	0.679	0.669	0.618
F-measure	0.5	0.698	**0.699**	0.631	0.61	0.539
Diabetes						
AUC	0.589	**0.799**	0.712	0.681	0.68	0.699
F-measure	0.511	**0.743**	0.623	0.663	0.661	0.632
Blood						
AUC	0.533	0.74	**0.744**	0.623	0.613	0.613
F-measure	0.413	0.74	**0.741**	0.5	0.51	0.592
Survival						
AUC	0.51	**0.73**	0.689	0.691	0.66	0.541
F-measure	0.413	**0.681**	0.531	0.533	0.543	0.534
SpamBase						
AUC	0.82	**0.919**	0.9	0.91	0.889	0.89
F-measure	0.795	**0.918**	0.87	0.893	0.873	0.882