Hybrid sampling Algorithm Based on Ant Colony Optimization and k-means Clustering

Mrs. S. Santha Subbulaxmi
PhD Research Scholar, Madurai Kamaraj University, Madurai, Tamil Nadu, India
santhamd3@gmail.com

Dr. G. Arumugam
Professor & Head of Department (Retd.), Computer Science Department,
Madurai Kamaraj University, Madurai, Tamil Nadu, India
gurusamyarumugam@gmail.com

Abstract - Class imbalance is very common in many real-world datasets and is an active research topic in machine learning and data mining. In this paper, a hybrid sampling algorithm is proposed to deal with class imbalance with minimum loss of existing information and additional computing power. The proposed algorithm undersamples the majority class data by identifying within-class sub-concepts and then deriving an optimal subset of majority class data using ant colony optimization. A representative training dataset is derived by combining the optimal subset of majority class data and minority class data. New synthetic instances are created in the minority class when imbalance still exists in the representative training dataset. The selection of majority class instances to solve the imbalanced data classification problem is novel in the proposed hybrid sampling algorithm. The proposed algorithm is examined on 15 imbalanced datasets. The experiment results revealed that the proposed algorithm achieves good performance in the aspects of performance measures, G-Mean and AUC. The proposed algorithm is compared with the popular sampling ensemble algorithms, and the results revealed that the proposed algorithm outperforms the popular sampling ensemble algorithms.

***Keywords*:** Imbalanced data; Classification; k-means clustering; Hybrid sampling; Ant colony optimization.

1. Introduction

Classification is important in data mining and machine learning and is an active research topic. The literature to date contains many classification algorithms, with logistic regression, decision trees, and support vector machines (SVMs) being some of the traditional ones. However, traditional classification algorithms struggle to achieve high classification accuracy when the data distribution is skewed. Data skewness is due to class imbalance in the dataset, and class imbalance exists when the classes of the data distribution have instances in different ratios. A class that contains only fewer instances is known as a minority class, while one that contains many instances is known as a majority class. Traditional classification algorithms are generally biased toward majority-class instances. However, because of their rarity, the relatively few minority-class instances can be very important in classification. The cost of misclassifying these rare instances is not the same as that for majority-class instances, thereby making classification more complicated. Other challenges in classifying imbalanced data include small sample size, within-class sub-concepts, class overlap, and high dimensionality.

Imbalanced datasets are very common in many real-world applications, including detecting oil spills from satellite radar images [1,2], identifying post-operative life expectancy in lung-care patients [3], detecting software defects [4], classifying network traffic [5], and diagnosing faults in wind turbines [6]. The literature contains much research into handling class imbalance, and many methods have been proposed for classifying imbalanced data. These can be categorized as being either data-preprocessing methods, algorithmic methods, or cost-sensitive methods. Data-preprocessing methods include data sampling, data cleaning, and feature selection. Data-sampling techniques are used to rebalance an imbalanced data distribution and can be categorized as oversampling, undersampling, or hybrid sampling.

Oversampling techniques replicate or create new synthetic instances in minority classes. The synthetic minority oversampling technique (SMOTE) [7] is very popular; SMOTE generates synthetic instances by interpolating linearly among minority instances that lie together. Other highly popular oversampling algorithms are ADASYN [8], Borderline–SMOTE [9], Safe-Level-SMOTE [10], and MWMOTE [11]. However, the key disadvantage of oversampling is that it increases the amount of computing power required. Undersampling techniques reduce the number of majority-class instances by either removing such instances or selecting among them. The selection or removal of such instances can be done in random or through some informative measure. Random undersampling (RUS) chooses majority-class instances randomly and derives representative majority-

class subsets, whereas focused undersampling [12] undersamples the majority-class instances that are located on the majority–minority borders. However, the key disadvantage of undersampling is that useful information can be lost upon removing majority-class instances. Instead, hybrid sampling combines oversampling and undersampling.

Popular data-cleaning techniques include Tomek links [13], the condensed nearest-neighbor rule [14], one-sided selection (OSS) [15], and Wilson's edited nearest-neighbor rule [16]. Feature-selection techniques are used to derive relevant features for learning by reducing the number of irrelevant features; for example, Dietterich et al. [17] identified the features that helped in class separability and selected only those features for learning. Algorithmic methods are used to change the learning process of a classification algorithm and make it suitable for imbalanced data; algorithmic modifications include a new splitting criterion [18] and adjusting the offset entropy in decision trees [19]. Cost-sensitive algorithms introduce misclassification costs in the training of the classifier or in the training-data instances: for SVMs, Veropoulos et al. [20] adopted the different-error-cost approach and assigned different misclassification costs to each class; Ting [21] used an instance-weighting method to assign higher weights to the instances of classes that have higher misclassification costs.

This paper proposes a novel hybrid sampling method that uses k-means clustering and ant colony optimization (ACO) to classify imbalanced data. The rest of the paper is structured as follows. We discuss the related research literature in Section 2 and give preliminary details of the proposed research work in Section 3. We describe our proposed approach in Section 4, and in Section 5 we explain the experimental framework and discuss the results. Finally, we present our conclusions in Section 6.

2. Related Work

In this section, we discuss previous research on imbalanced-data classification ensembles, cluster-based undersampling, and metaheuristic-optimized imbalanced-data classification algorithms.

SMOTEBagging [22] generates different training datasets for learning and solves the problem of classifying imbalanced data. It sets a resampling rate (10–100%) for each iteration to replicate the existing minority-class instances. SMOTE is used to generate synthetic instances for the remaining requirement, thereby creating a diverse ensemble. SMOTEBoost [23] is a combination of SMOTE [7] and boosting [24], and AdaCost [25] is used in the AdaBoost algorithm [24]. UNDERBagging [26] applies random undersampling to the majority class and creates different majority subsets; a representative training set is created based on the majority subset and minority, and bagging is applied to the representative training set to create the ensemble. RUSBoost [27] sets the initial weights of the training instances and randomly undersamples the majority-class instances; it updates the instances' weights based on the learning and builds the ensemble. Other popular ensemble-based imbalanced-data classification algorithms are EasyEnsemble [28], BalanceCascade, RandomBalanace Boosting [29], RandomBalance Bagging [29], and StochasticEnsemble [30].

A cluster-based undersampling algorithm [31] groups all instances into k clusters and undersamples the majority-class instances in each cluster to obtain a balanced dataset. ClusterOSS [32] groups the majority-class instances and performs undersampling by identifying the instances closest to the cluster center. The Tomek-link algorithm is finally applied on the under-sampled dataset to solve the imbalanced data classification algorithm. The diversified sensitivity-based undersampling algorithm [33] samples the majority class based on the data distribution to achieve diversity in sampling. First, it clusters the majority-class instances by setting the number of clusters as the number of minority-class instances; the number of minority-class clusters is set as the square root of the number of minority-class instances. It then undersamples the instances using a stochastic sensitivity measure, and a radial-basis-function neural network is used to train the training instances. The algorithm outputs a balanced dataset that achieves the highest sensitivity. Meanwhile, the ensemble AnyNovel [34] adopts a two-step cluster-formation technique; it uses supervised learning to form the first level clusters and then unsupervised clustering to detect the sub-concepts within each class cluster, and it can handle drift and singular outliers.

ACO [35] is used to derive an optimal majority-class subset to solve the imbalanced data classification problem. Cluster-based evolutionary undersampling [36] combines clustering and a genetic algorithm to classify imbalanced data. Yu et al. [37] proposed an extreme learning machine based on optimal decision output compensation to improve the recognition of minority-class instances by using particle swarm optimization, and grey-wolf optimization [38] optimizes the regularization parameters of the extreme learning machine and solves the problem of imbalanced-data classification.

The literature inspires us to use grouping and subgrouping to identify within-class sub-concepts. To use the search space efficiently and obtain optimal results, we use the power of metaheuristics, and we propose herein a hybrid sampling algorithm based on ACO and k-means clustering.

3. Preliminaries

Developed by Dorigo and colleagues [39–42], the ACO algorithm is a bio-inspired metaheuristic algorithm that mimics the foraging behavior of ants to find a solution. When searching for food, an ant explores its surroundings randomly and seeks a pathway between its nest and the food source; on its return trip to the nest, the ant lays pheromones on its pathway. Collectively, the ants prefer pathways with higher pheromone concentrations, so in any iteration, the ants' most-favorable pathway is the one with the highest pheromone concentration, and it is the pathway where most ants have traveled. The most-favorable pathway is the best solution in that iteration. After each iteration, each pathway is either intensified or weakened by updating the pheromones, and thus the best pathways have more chance of success in the next iteration. Once the ACO has converged, all the ants have chosen the same path, and the optimal solution is attained. The ACO framework is as follows: step 1: initialize the parameters; step 2: construct solutions based on the parameters; step 3: evaluate the fitness of the solution; step 4: find the best solution; step 5: update the parameters; step 6: return the optimal solution.

The ACO algorithm is used to find optimal solutions in many applications (e.g., the traveling-salesman problem, vehicle routine, frequency assignment) and solve imbalanced-data classification. The ACOSampling algorithm [36] derives an optimal training dataset for a skewed DNA microarray dataset by filtering out those majority instances with less information. Herein, we use the same mathematical terms for the ACO parameters as those in ACOSampling, and the pheromone updating is calculated as

$$\tau_{ij}(t+1) = \rho \times \tau_{ij}(t) + \Delta\tau_{ij}, \tag{3.1}$$

where ρ is the evaporation coefficient and $\Delta\tau_{ij}$ is the increase in pheromones for an excellent pathway. The i represents the ith majority sample in the training set and j represents the pathway. The evaporation coefficient ρ controls the decrease of pheromones. The pheromones in the pathways of the best 10% ants are added, and these pathways are stored in the set E. The increase in pheromones for an excellent pathway is calculated as

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{0.1 \times ant\_n} \times \text{fitness} & \text{if pathway}_{ij} \in \text{E} \\ 0 & \text{if pathway}_{ij} \notin \text{E} \end{cases}, \tag{3.2}$$

where ant_n is the number of ants in the colony. The ACO algorithm searches the entire search space of the solutions and returns an optimal solution by choosing the best solution components with the help of the ACO parameters.

4. Proposed Algorithm

This section presents our proposed hybrid sampling algorithm based on ACO and k-means clustering. It is designed to address the class imbalance in binary-class numeric datasets. ACOSampling [35] uses ACO to derive an optimal subset of the majority-class data by selecting majority-class instances. Motivated by ACOSampling, our proposed method uses ACO and a single constraint. First, it groups and subgroups the majority-class instances and then uses ACO to choose the subgroups. It then applies the constraint that the total number of instances in the chosen subgroups should not be less than the number of minority-class instances; if the number of instances in the selected majority-class subsets is greater than the number of minority-class instances, then new synthetic instances are created and an optimal representative training dataset (RT) is derived to build the classifier.

The proposed hybrid sampling algorithm based on ACO and k-means clustering returns a hybrid sampling classifier based on the given inputs, the training dataset, the validation dataset, the size ant_n, number of ants in the ant colony, the iteration number, and the base classifier. The whole procedure of the proposed algorithm is outlined in Fig. 1.

The given training dataset is scaled with the Z score and then divided into majority- and minority-class data. The proposed algorithm uses ACO to derive an optimal subset of the majority-class data. To obtain the optimal majority-class subset, it divides the majority-class data into groups and then subgroups (SG). The silhouette method is used to calculate the number k of groups to form in the majority-class data, and the k-means clustering algorithm is used to create those k groups. For each group, we calculated the number of subgroups (GI) to be formed. For each group, GI is computed by the square root of the number of instances in the particular group. The subgroups (SG) are then formed within that group by calculating the Euclidean distance of each instance with its centroid and then cut the numerical variable, Euclidean distance into GI quantile intervals. Therefore, in that group, the instances that lie within the interval ranges of the quantile intervals form GI subgroups. All the k groups are subgrouped by following the same approach.

The proposed hybrid sampling algorithm presents a solution to imbalanced data classification problem by choosing the solution components based on ACO parameters. Here the solution components refer to the subgroups (SG) of the majority class, and solution refers to the classifier built on the ACO derived representative training dataset. The ACO parameters, the evaporation coefficient ρ, and pheromone matrix of size SG are initialized. In an iteration, for each ant in the colony, the ACO algorithm proposes a solution by choosing the subgroups (Sel_SG) randomly based on the pheromone levels. The number of instances in the selected subgroups is denoted

as Sel_SG_Size. If Sel_SG_Size equals the number of instances in the minority-class data, then an RT is created by combining instances in Sel_SG and minority-class data instances. If Sel_SG_Size is greater than the number of instances in the minority-class data, then synthetic instances are created in the minority class to make equal the number of instances in both classes. An RT is created based on the instances in Sel_SG, the minority-class data, and the newly created synthetic data. The new synthetic instances to be generated for each minority instance is calculated as

$$\text{perc.over} = (\text{Mi\_Size} / (\text{Sel\_SG\_Size} - \text{Mi\_Size})) \quad (4.1)$$

Each ant in the colony of size ant_n derives ant_n RTs. For each RT, a classifier is built based on the given base classifier and the RT. The built classifier is evaluated with the validation dataset, and the performance of the classifier is measured. The geometric mean (G-Mean) and area under the receiver operating characteristic (ROC) curve (AUC) are used to calculate the fitness value for each ant as

$$\text{Fitness} = (\text{G-Mean} + \text{AUC}) / 2. \quad (4.2)$$

If Sel_SG_Size is less than the number of instances in the minority-class data, then the fitness value is set as −1. The maximum fitness value of all ant_n ants in a given iteration is taken as the best fitness value, and the RT of the ant with the best fitness value is taken as the best RT (BRT). The pheromone matrix is then updated using Eq. (3.1). At the end of each iteration, the BRT is selected; the BRT returned at the iteration_number is the optimal RT, and the classifier based on it is returned.

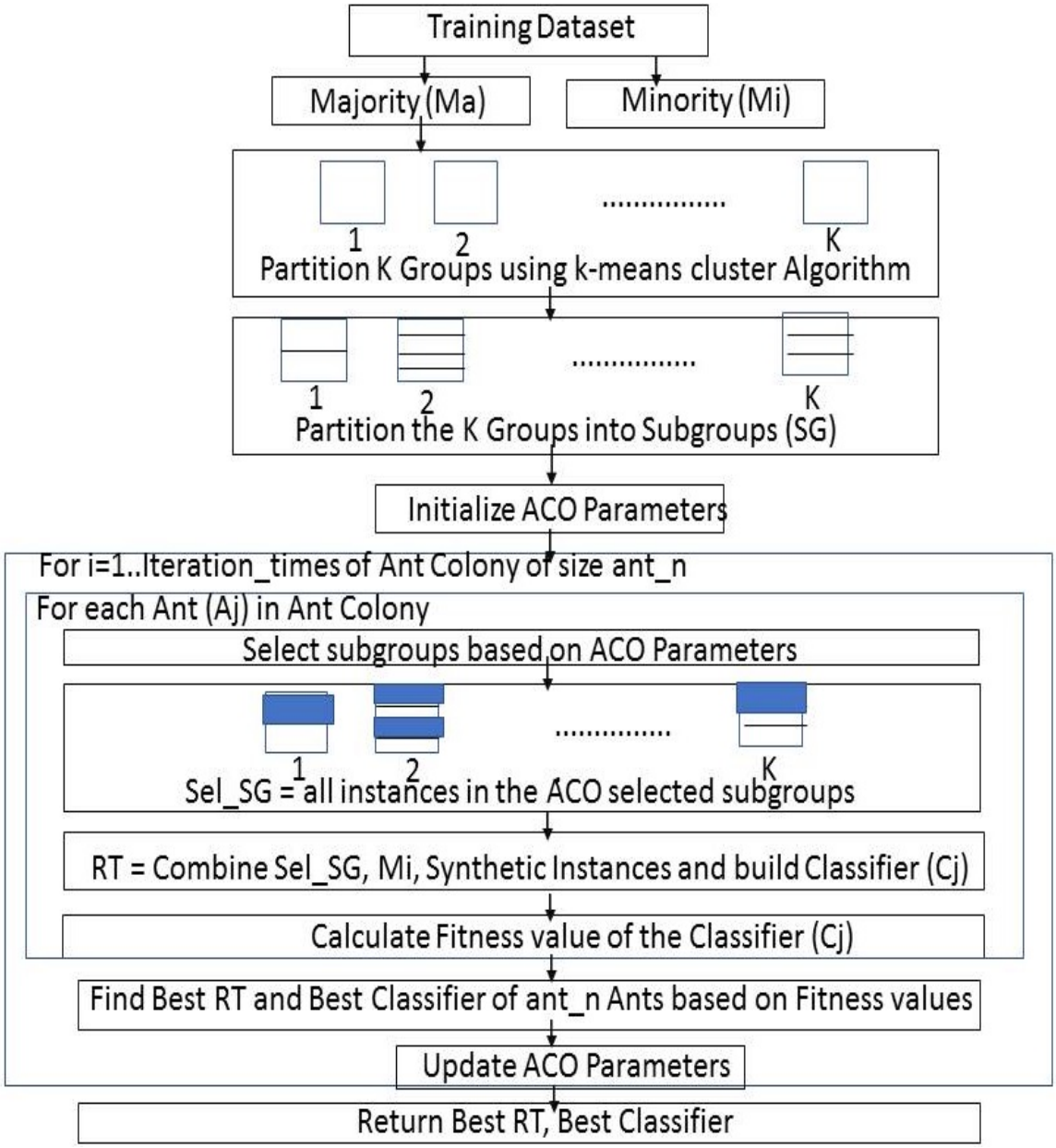

Fig. 1. Flowchart of hybrid sampling algorithm based on ant colony optimization and k-means clustering

5. Experimental Framework and Results

This section describes the experimental framework used to evaluate our proposed algorithm, and we discuss the obtained performance results. To develop our proposed algorithm and to conduct the experiments, we used the R software environment (ver. 4.0.2). The k-means algorithm was implemented by using the R package "Cluster," and the subgroups were created using the cut2 function in the R package "Hmisc v4.4-1." In the experiments, an SVM was used as the base classifier and was implemented by using the R package "e1071" with the SVM kernel (svm.ker) set as "radial." We compared our proposed algorithm with the popular sampling ensemble algorithms UnderBagging, RUSBoost, SMOTEBagging, and SMOTEBoost, which are detailed in the respective references given in Section 2. These algorithms were implemented using the R package "ebmc."

In this proposed algorithm, the evaporation coefficient *p* is initialized as 0.8, the pheromone matrix of size SG is initialized as 1, the number of iterations is defined as 50, and the size ant_n of the ant colony is defined as 20.

The performance of each classifier is estimated from the metrics G-Mean and AUC. The value of G-Mean reflects the balance between the classification performances of the majority and minority classes and is calculated as the square root of the product of precision and recall; a low value of G-Mean represents poor performance of the minority class. The ROC graph is a plot of the false positive rate on the horizontal axis and the true positive rate on the vertical axis, and AUC is a single scalar value that represents the excepted performance of the ROC curve, reflecting the strength of the classifier. In the experiments, the ROC and AUC are measured using the R package "PROC".

The benchmarking datasets used in the experiments are obtained from the KEEL (Knowledge Extraction based on Evolutionary Learning) data repository [43]. A total of 15 datasets are used from KEEL data repository. The dataset name, number of features, total number of instances, and imbalance ratio are summarized in Table 1. The number of features ranges from 11 to 18, the number of instances ranges from 214 to 1484, and the imbalance ratio ranges from 1.86 to 44. The datasets were partitioned into training (60%), validation (20%), and test (20%) datasets. In the experiments, we treated the target variable as 1 for minority class and 0 for majority class.

Experiments were conducted to assess the proposed algorithm with the 15 imbalanced datasets from the KEEL repository. In the experiments, the proposed algorithm achieved G-Mean and AUC performance scores above 0.9, thereby showing that it can handle class imbalance in a skewed data distribution.

The proposed algorithm was also compared with the popular sampling ensemble algorithms UnderBagging, RUSBoost, SMOTEBagging, and SMOTEBoost. The metrics G-Mean and AUC of the proposed algorithm and popular sampling ensemble algorithms are measured, and the assessment results are given in Table 2.

Fig 2 shows a radar chart of the G-Mean scores of the proposed algorithm and the popular sampling ensemble algorithms. The minimum G-Mean value achieved by the proposed algorithm is 0.906344 and maximum G-Mean value achieved by the proposed algorithm is 1.

The G-Mean value achieved by the UnderBagging algorithm ranges from 0.552506 and 0.987096. The G-Mean value of the proposed algorithm is greater than the UnderBagging algorithm for all the 15 datasets. The UnderBagging algorithm didn't able to perform well with the 7 datasets: glass0, glass6, yeast-2_vs_8, ecoli-0-2-6-7_vs_3-5, ecoli-0-1-4-7_vs_5-6, led7digit-0-2-4-5-6-7-8-9_vs_1, and winequality-white-3_vs_7. The G-Mean value achieved by the UnderBagging algorithm are less than 0.9 for these 7 datasets whereas the proposed algorithm is able to achieve G-Mean value higher than 0.9.

The G-Mean value achieved by the RUSBoost algorithm ranges from 0.561951 and 0.984251. The G-Mean value of the proposed algorithm is greater than the RUSBoost algorithm for all the 15 datasets. The RUSBoost algorithm didn't able to perform well with the 8 datasets: yeast3, yeast-2_vs_8, winequality-white-3_vs_7, glass6, glass0, ecoli-0-6-7_vs_5, ecoli-0-2-6-7_vs_3-5, and ecoli-0-1-4-7_vs_5-6. The G-Mean value achieved by the RUSBoost algorithm are less than 0.9 for these 8 datasets whereas the proposed algorithm is able to achieve G-Mean value higher than 0.9.

The G-Mean value achieved by the SMOTEBagging algorithm ranges from 0.542897 and 0.987096. The G-Mean value of the proposed algorithm is greater than the SMOTEBagging algorithm for all the 15 datasets. The SMOTEBagging algorithm didn't able to perform well with the 7 datasets: yeast-2_vs_8, winequality-white-3_vs_7, led7digit-0-2-4-5-6-7-8-9_vs_1, glass6, glass0, ecoli-0-2-6-7_vs_3-5, and ecoli-0-1-4-7_vs_5-6. The G-Mean value achieved by the SMOTEBagging algorithm are less than 0.9 for these 7 datasets whereas the proposed algorithm is able to achieve G-Mean value higher than 0.9.

The G-Mean value achieved by the SMOTEBoost algorithm ranges from 0.5699 and 0.986754. The G-Mean value of the proposed algorithm is greater than the SMOTEBoost algorithm for all the 15 datasets. The SMOTEBoost algorithm didn't able to perform well with the 9 datasets: yeast3, yeast-2_vs_8, winequality-white-3_vs_7, glass6, glass-0-1-2-3_vs_4-5-6, glass0, ecoli-0-6-7_vs_5, ecoli-0-2-6-7_vs_3-5, and ecoli-0-1-4-7_vs_5-6. The G-Mean value achieved by the SMOTEBoost algorithm are less than 0.9 for these 9 datasets whereas the proposed algorithm is able to achieve G-Mean value higher than 0.9.

Fig 3 shows a radar chart of the AUC scores of the proposed algorithm and the popular sampling ensemble algorithms. The minimum AUC value achieved by the proposed algorithm is 0.91871 and maximum AUC value achieved by the proposed algorithm is 1. The AUC value achieved by the UnderBagging algorithm ranges from 0.54432 and 0.997207. The AUC value of the proposed algorithm is greater than the UnderBagging algorithm for all the 15 datasets. The UnderBagging algorithm didn't able to perform well with the 9 datasets: yeast3, yeast-2_vs_8, yeast-0-2-5-7-9_vs_3-6-8, led7digit-0-2-4-5-6-7-8-9_vs_1, glass0, ecoli-0-6-7_vs_5, ecoli-0-6-7_vs_5, ecoli-0-2-6-7_vs_3-5, and ecoli-0-1-4-7_vs_5-6. The AUC value achieved by the UnderBagging algorithm are less than 0.9 for these 9 datasets whereas the proposed algorithm is able to achieve AUC value higher than 0.9.

The AUC value achieved by the RUSBoost algorithm ranges from 0.572464 and 0.997207. The AUC value of the proposed algorithm is greater than the RUSBoost algorithm for all the 15 datasets. The RUSBoost algorithm didn't able to perform well with the 8 datasets: yeast3, yeast-2_vs_8, yeast-0-2-5-7-9_vs_3-6-8, led7digit-0-2-4-5-6-7-8-9_vs_1, glass0, ecoli-0-6-7_vs_5, ecoli-0-2-6-7_vs_3-5, and ecoli-0-1-4-7_vs_5-6. The AUC value achieved by the RUSBoost algorithm are less than 0.9 for these 8 datasets whereas the proposed algorithm is able to achieve AUC value higher than 0.9.

The AUC value achieved by the SMOTEBagging algorithm ranges from 0.530039 and 0.988372. The AUC value of the proposed algorithm is greater than the SMOTEBagging algorithm for all the 15 datasets. The SMOTEBagging algorithm didn't able to perform well with the 9 datasets: yeast3, yeast-2_vs_8, yeast-0-2-5-7-9_vs_3-6-8, winequality-white-3_vs_7, led7digit-0-2-4-5-6-7-8-9_vs_1, glass0, ecoli-0-6-7_vs_5, ecoli-0-6-7_vs_5, and ecoli-0-2-6-7_vs_3-5. The AUC value achieved by the SMOTEBagging algorithm are less than 0.9 for these 9 datasets whereas the proposed algorithm is able to achieve AUC value higher than 0.9.

The AUC value achieved by the SMOTEBoost algorithm ranges from 0.725 and 0.989691. The AUC value of the proposed algorithm is greater than the SMOTEBoost algorithm for all the 15 datasets. The SMOTEBoost algorithm didn't able to perform well with the 6 datasets: winequality-white-3_vs_7, glass6, glass0, ecoli-0-6-7_vs_5, ecoli-0-2-6-7_vs_3-5, and ecoli-0-1-4-7_vs_5-6. The AUC value achieved by the SMOTEBoost algorithm are less than 0.9 for these 6 datasets whereas the proposed algorithm is able to achieve AUC value higher than 0.9.

It is witnessed clearly that the proposed algorithm performs better than the four popular sampling ensemble algorithms in a higher margin. The popular undersampling algorithm ACOSampling [36] is the motivation for this research work. It derives an optimal majority class subset by choosing instances in the majority class. The proposed algorithm focuses on the majority class's within-class sub-concepts and aims to select the majority-class data's optimal subset. Within the entire search space of the within-class sub-concepts, the ACO algorithm selects the optimal combination of within-class sub-concepts of the majority-class data and minority class data. An RT is derived based on the optimal subsets of majority and minority class data. New synthetic instances are created in the minority-class data only if there is a class imbalance in the RT. The selection of the optimal subset of the majority-class data is novel and unique, and so the proposed hybrid sampling framework differs from the existing algorithms. The experiment results have also proven that the proposed algorithm is also able to deal with the imbalanced datasets.

Table 1. Details of datasets.

Sl.No.	Dataset	No. of attributes	No. of examples	IR
1	wisconsin	9	683	1.86
2	glass0	9	214	2.06
3	glass-0-1-2-3_vs_4-5-6	9	214	3.2
4	vehicle0	18	846	3.25
5	new-thyroid2	5	215	5.14
6	glass6	9	214	6.38
7	yeast3	8	1484	8.1
8	yeast-2_vs_8	8	482	23.1
9	ecoli-0-6-7_vs_5	6	220	10
10	yeast-0-2-5-7-9_vs_3-6-8	8	1004	9.14
11	ecoli-0-2-6-7_vs_3-5	7	224	9.18
12	ecoli-0-6-7_vs_5	6	220	10
13	ecoli-0-1-4-7_vs_5-6	6	332	12.28
14	led7digit-0-2-4-5-6-7-8-9_vs_1	7	443	10.97
15	winequality-white-3_vs_7	11	900	44

Table 2. Evaluation of performances of proposed hybrid sampling algorithm and popular sampling ensemble algorithms in terms of G-Mean and AUC.

Sl.No.	Dataset	Metric	Proposed algorithm	Under Bagging	RUSBoost	SMOTE Bagging	SMOTE Boost
1	wisconsin	G-Mean	0.982905	0.976584	0.974245	0.976584	0.965507
		AUC	0.9875	0.978721	0.981481	0.978721	0.97561
2	glass0	G-Mean	0.906344	0.811246	0.791695	0.811246	0.811246
		AUC	0.955084	0.780051	0.760101	0.780051	0.780051
3	glass-0-1-2-3_vs_4-5-6	G-Mean	1	0.942809	0.942809	0.942809	0.881917
		AUC	1	0.97619	0.97619	0.97619	0.954545
4	vehicle0	G-Mean	0.992395	0.982003	0.982003	0.960883	0.960883
		AUC	0.98	0.975775	0.975775	0.967445	0.967445
5	new-thyroid2	G-Mean	1	0.984251	0.984251	0.984251	0.984251
		AUC	1	0.944444	0.944444	0.944444	0.944444
6	glass6	G-Mean	1	0.707107	0.707107	0.707107	0.698638
		AUC	1	0.988372	0.988372	0.988372	0.738095
7	yeast3	G-Mean	0.958425	0.912412	0.891406	0.909787	0.870738
		AUC	0.91871	0.835476	0.807837	0.883867	0.918606
8	yeast-2_vs_8	G-Mean	1	0.552506	0.561951	0.542897	0.57735
		AUC	1	0.54432	0.572464	0.530039	0.989691
9	ecoli-0-6-7_vs_5	G-Mean	1	0.987096	0.57735	0.987096	0.816497
		AUC	1	0.875	0.97561	0.875	0.9875
10	yeast-0-2-5-7-9_vs_3-6-8	G-Mean	0.968848	0.940331	0.908968	0.917	0.927601
		AUC	0.925747	0.877619	0.824612	0.858095	0.914697

11	ecoli-0-2-6-7_vs_3-5	G-Mean	0.982905	0.562352	0.5699	0.562352	0.5699
		AUC	0.9875	0.641026	0.725	0.641026	0.725
12	ecoli-0-6-7_vs_5	G-Mean	1	0.973329	0.973329	0.986754	0.986754
		AUC	1	0.75	0.75	0.833333	0.833333
13	ecoli-0-1-4-7_vs_5-6	G-Mean	0.92582	0.749495	0.749495	0.845154	0.749495
		AUC	0.991667	0.87541	0.87541	0.983607	0.87541
14	led7digit-0-2-4-5-6-7-8-9_vs_1	G-Mean	0.993808	0.894082	0.906084	0.894082	0.929622
		AUC	0.944444	0.743333	0.785173	0.743333	0.931327
15	winequality-white-3_vs_7	G-Mean	1	0.707107	0.707107	0.705118	0.705118
		AUC	1	0.997207	0.997207	0.747191	0.747191

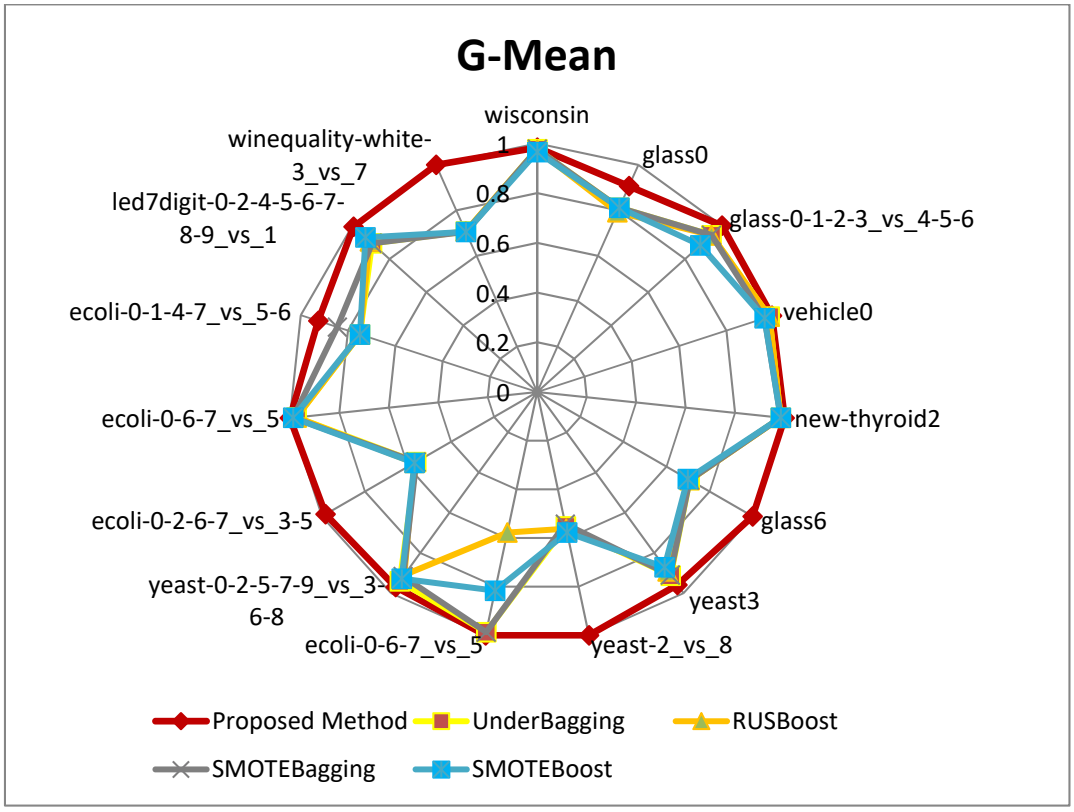

Fig. 2. G-Mean values of proposed hybrid sampling algorithm and popular sampling ensemble algorithms.

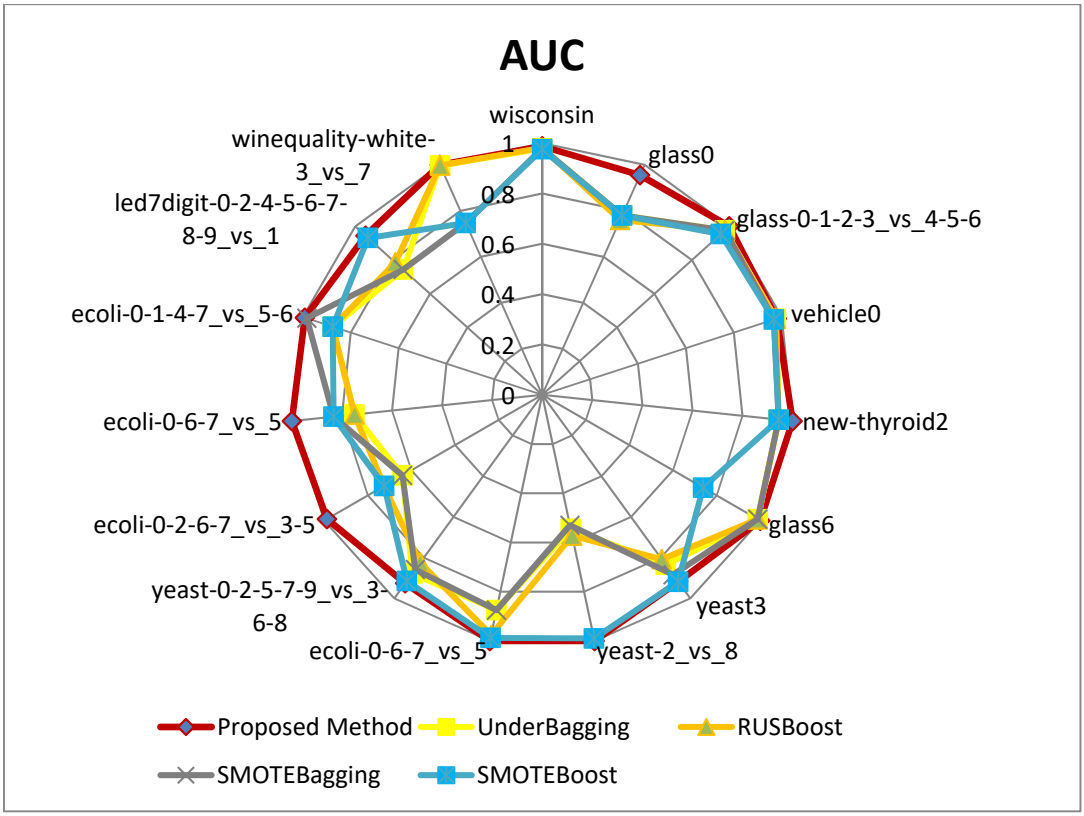

Fig. 3. AUC values of proposed hybrid sampling algorithm and popular sampling ensemble algorithms

Conclusion

Skewed data distributions are prevalent in many datasets with real-world applications and pose a considerable challenge to researchers. In attempting to address this challenge, oversampling algorithms used to increase the computational effort and undersampling algorithms used to remove important information. The proposed hybrid sampling algorithm combines both: based on the prevailing data distribution, it undersamples the majority-class data and oversamples the minority-class data if required. The ACO algorithm is used to derive a representative training dataset. The experimental results of the proposed algorithm against popular sampling ensemble algorithms showed that it achieves better performance. The present work is a unique approach to dealing with skewed distributions.

References

[1] Kubat, M.; Matwin, S. (1997): Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, pp. 179–186.
[2] Kubat, M.; Holte, R.; Matwin, S. (1997): Learning when negative examples abound. In: van Someren, M., Widmer, G. (eds.) Proceedings of the 9th European Conference on Machine Learning, Springer, Berlin/Heidelberg, pp. 146–153.
[3] Zieba, M.; Tomczak, J.M.; Lubicz, M.; Swiatek, J. (2014): Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. Appl. Soft Comput. 14, Part A, pp. 99–108.
[4] Rodriguez, D., Herraiz, I.; Harrison, R.; Dolado, J.; Riquelme, J.C. (2014): Preliminary comparison of techniques for dealing with imbalance in software defect prediction. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, ACM, New York, pp. 43:1–43:10.
[5] Wei, H.; Sun, B.; Jing, M. (2014): Balancedboost: a hybrid approach for real-time network traffic classification. In: 23rd International Conference on Computer Communication and Networks (ICCCN), Shanghai, pp. 1–6.
[6] Wu, Z.; Lin, W.; Ji, Y. (2018): An integrated ensemble learning model for imbalanced fault diagnostics and prognostics. IEEE Access 6, pp. 8394–8402.
[7] Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. (2002): SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, pp. 321–357.
[8] He, H.; Bai, Y.; Garcia, E.A.; Li, S. (2008): ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the 2008 IEEE International Joint Conference Neural Networks (IJCNN'08), Hong Kong, pp. 1322–1328.
[9] Han, H.; Wang, W.Y.; Mao, B.H. (2005): Borderline–SMOTE: a new over-sampling method in imbalanced data sets learning. In: Proceedings of the 2005 International Conference on Intelligent Computing (ICIC'05), Hefei. Lecture Notes in Computer Science, vol. 3644, pp. 878–887.
[10] Bunkhumpornpat, C.; Sinapiromsaran, K.; Lursinsap, C. (2009): Safe-level-SMOTE: safe-level-synthetic minority over-sampling TEchnique for handling the class imbalanced problem. In: Proceedings of the 13th Pacific–Asia Conference on Advances in Knowledge Discovery and Data Mining PAKDD'09, Bangkok, pp. 475–482.
[11] Barua, S.; Islam, M.M.; Yao, X.; Murase, K.(2014): MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans. Know. Data Eng. 26(2), pp. 405–425.
[12] Japkowicz N. (2000): Learning from imbalanced data sets: a comparison of various strategies. AAAI Tech Report WS-00-05.
[13] Tomek, I.; (1976): Two modifications of CNN. IEEE Trans. Syst. Man Commun. 6, pp. 769–772.

[14] Hart, P.E. (1968): The condensed nearest neighbor rule. IEEE Trans. Inf. Theory 14, pp. 515–516.
[15] Kubat, M.; Holte, R.C.; Matwin, S. (1997): Learning when negative examples abound. In: van Someren, M., Widmer, G. (eds.) Proceedings of the 9th European Conference on Machine Learning (ECML'97). Lecture Notes in Computer Science, Springer, Berlin/New York vol. 1224, pp. 146–153.
[16] Wilson, D.L. (1972): Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. 2(3), pp. 408–421.
[17] Dietterich, T.; Kearns, M.; Mansour, Y. (1996): Applying the weak learning framework to understand and improve C4.5. In: Proc 13th International Conference on Machine Learning, pp. 96–100.
[18] Marcellin, S.; Zighed, D.A.; Ritschard, G. (2006): Detection of breast cancer using an asymmetric entropy measure. In: COMPSTAT Proceedings In Computational Statistics, pp. 975–982.
[19] Drummond, C.; Holte, R.C. (2003): C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II, held in conjunction with ICML.
[20] Veropoulos, K.; Campbell, C.; Cristianini, N. (1999): Controlling the sensitivity of support vector machines. In: Proceedings of the International Joint Conference on AI, Stockholm, pp. 55–60.
[21] Ting, K.M. (2002): An instance-weighting method to induce cost-sensitive trees. IEEE Trans. Knowl. Data Eng. 14(3), pp. 659–665.
[22] Wang, S.; Yao, X. (2009): Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09), Nashville, pp. 324–331.
[23] Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. (2003): SMOTEBoost: improving prediction of the minority class in boosting. In: Knowledge Discovery in Databases (PKDD'03), Springer, Berlin/Heidelberg, pp. 107–119.
[24] Freund, Y.; Schapire. R.E. (1997): A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55(1), pp.119–139.
[25] Fan, W.; Stolfo, S.J.; Zhang, J.; Chan, P.K. (1999): AdaCost: misclassification cost-sensitive boosting. In: International Conference on Machine Learning, pp. 97–105.
[26] Barandela, R.; Valdovinos, R.M.; Sánchez, J.S. (2003): New applications of ensembles of classifiers. Pattern Anal. Appl. 6, pp. 245–256.
[27] Seiffert, C.; Khoshgoftaar, T.; Van Hulse, J.; Napolitano, A. (2010): Rusboost: a hybrid approach to alleviating class imbalance. IEEE Trans. Syst. Man Cybern. A Syst. Hum. 40(1), pp. 185–197.
[28] Liu, X.Y.; Wu, J.; Zhou, Z.H. (2009): Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. B Cybern. 39(2), pp. 539–550.
[29] Díez-Pastor, J.F.; Rodríguez, J.J.; García-Osorio, C.; Kuncheva, L.I. (2015): Random balance: ensembles of variable priors classifiers for imbalanced data. Know. Based Syst. 85, pp. 96–111.
[30] Blagus, R.; Lusa, L. (2017): Gradient boosting for high-dimensional prediction of rare events. Comput. Stat. Data Anal. 113, pp. 19–37.
[31] Yen, S.-J.; Lee, Y.-S. (2009): Cluster-based under-sampling approaches for imbalanced data distributions. Expert Syst. Appl. 36(3), pp. 5718–5727.
[32] Barella, V.; Costa, E.; Carvalho, A.C.P.L.F. (2014): ClusterOSS: a new undersampling method for imbalanced learning. Technical report.
[33] Ng, W.W.Y.; Hu, J., Yeung, D.S.; Yin, S., Roli, F. (2015): Diversified sensitivity-based undersampling for imbalance classification problems. IEEE Trans. Cybern. 45(11), pp. 2402–2412.
[34] Abdallah, Z.S.; Gaber, M.M.; Srinivasan, B.; Krishnaswamy, S. (2016): Anynovel: detection of novel concepts in evolving data streams. Evol. Syst. 7(2), pp. 73–93.
[35] Yu, H.; Ni, J.; Zhao, J. (2013): ACOSampling: an ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. Neurocomputing 101, pp. 309–318.
[36] Kim, H.; Jo, N.; Shin, K. (2016): Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. Expert Syst. Appl. 59, pp. 226–234.
[37] Yu, H.; Sun, C.; Yang, X.; Yang, W.; Shen, J.; Qi, Y. (2016): ODOC-ELM: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data. Knowl.-Based Syst. 92, pp.55–70.
[38] Thammasakorn, C.; Chiewchanwattana, S.; Sunat, K. (2018): Optimizing weighted ELM based on gray wolf optimizer for imbalanced data classification. In: 10th International Conference on Information Technology and Electrical Engineering (ICITEE) 2018 Jul 24, IEEE-2018, pp. 512–517.
[39] Dorigo, M. (1992): Optimization, learning and natural algorithms (in Italian), Ph.D. Thesis, Dipartimento diElettronica, Politecnico di Milano, Italy.
[40] Dorigo, M.; Gambardella, L.M. (1997): Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. 1(1), 53–66.
[41] Dorigo, M.; Maniezzo, V.; Colorni, A. (1991): Positive feedback as a search strategy. Tech. Report 91-016, Dipartimentodi Elettronica, Politecnico di Milano, Italy.
[42] Dorigo, M.; Maniezzo, V.; Colorni, A. (1996): Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B Cybern. 26(1), pp. 29–41.
[43] Alcalá-Fdez, J.; Fernández, A.; Luengo, J.; Derrac, J.; García, S.; Sánchez, L.; Herrera, F. (2011): KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Logic Soft Comput. 17, pp. 255–287.

Authors Profile

Mrs. Santha Subbulaxmi S, is a Ph.D. research scholar at Computer Science department in Madurai Kamaraj University. Her research interests include data mining, machine learning and optimization algorithms. She received her MCA degree in Computer Science at Madurai Kamaraj University in 1997.

Dr. Arumugam G, received his M.Sc. Applied Sciences (Faculty of Engineering) from University of Madras in 1980, and his Ph.D. degree in Computer Science from the University of Pierre and Marie Curie, Paris, France in 1987 respectively. He worked as Professor & Head of Department at Computer Science Department in Madurai Kamaraj University. His research interests include data mining, text mining, image processing, machine learning and optimization algorithms. He has published 57 referred journal articles and conference articles in these fields.