



Optimization Based Undersampling for Imbalanced Classes

Fatih SAĞLAM^{1,*}, Mervenur SÖZEN², Mehmet Ali CENGİZ³

¹*Ondokuz Mayıs University, Faculty of Science and Literature, Department of Statistics, 55100, Samsun, Türkiye*

fatih.saglam@omu.edu.tr, ORCID: 0000-0002-2084-2008

²*Ondokuz Mayıs University, Faculty of Science and Literature, Department of Statistics, 55100, Samsun, Türkiye*

mervenur.pala1@gmail.com, ORCID: 0000-0001-5603-5382

³*Ondokuz Mayıs University, Faculty of Science and Literature, Department of Statistics, 55100, Samsun, Türkiye*

macengiz@omu.edu.tr, ORCID: 0000-0002-1271-2588

Received: 21.02.2021

Accepted: 25.10.2021

Published: 31.12.2021

Abstract

The classification methods consider the probability of predicting the majority class to be high when the number of class observations is different. To address this problem, there are some methods such as resampling methods in the literature. Undersampling, one of the resampling methods, creates balance by removing data from the majority class. This study aims to compare different optimization methods to determine the most suitable observations to be taken from the majority class while undersampling. Firstly, a simple simulation study was conducted and graphs were used to analyze the discrepancy between the resampled datasets. Then, different classifier models were constructed for different imbalanced data sets. In these models, random undersampling, undersampling with genetic algorithm, undersampling with differential evolution algorithm, undersampling with an artificial bee colony, and under-sampling with particle herd optimization were compared. The results were given rank numbers differing depending on the



classifiers and data sets and a general mean rank was obtained. As a result, when undersampling, artificial bee colony was seen to perform better than other methods of optimization.

Keywords: Imbalanced classes; Classification; Undersampling; Optimization.

Dengesiz Sınıflamada Optimizasyona Dayalı Azörnekleme

Öz

Sınıflama yöntemleri, sınıf gözlemlerinin sayısı farklı olduğunda çoğunluk sınıfını tahmin etme olasılığının yüksek olduğunu düşünür. Bu sorunu gidermek için literatürde yeniden örnekleme yöntemleri gibi bazı yöntemler bulunmaktadır. Yeniden örnekleme yöntemlerinden biri olan azörnekleme, çoğunluk sınıftan verileri silerek denge oluşturur. Bu çalışma, az örnekleme yapılırken çoğunluk sınıftan alınacak en uygun gözlemleri belirlemek için farklı optimizasyon yöntemlerini karşılaştırmayı amaçlamaktadır. İlk olarak, basit bir simülasyon çalışması yapılmış ve yeniden örneklenen veri setleri arasındaki farklılığı analiz etmek için grafikler kullanılmıştır. Daha sonra, farklı dengesiz veri setleri için farklı sınıflayıcı modelleri oluşturulmuştur. Bu modellerde rastgele azörnekleme, genetik algoritma ile azörnekleme, diferansiyel evrim algoritması ile azörnekleme, yapay arı kolonisi ile azörnekleme ve parçacık sürüsü optimizasyonu ile azörnekleme karşılaştırılmıştır. Sonuçlara sınıflandırıcılara ve veri setlerine göre değişen sıra numaraları verilmiş ve genel bir ortalama sıra elde edilmiştir. Sonuç olarak, yetersiz örnekleme yapıldığında, yapay arı kolonisinin diğer optimizasyon yöntemlerinden daha iyi performans gösterdiği görülmüştür.

Anahtar Kelimeler: Dengesiz sınıflar; Sınıflama; Azörnekleme; Optimizasyon.

1. Introduction

Class imbalance is one of the major problems in machine learning. Class imbalance leads to bias in the learning process from the data set. This bias causes incorrect predictions and makes it difficult to evaluate the model. Resampling is one of the most frequently used methods for dealing with this problem. In this method, balance is achieved by increasing the number of data of minority class observations and/or by reducing the number of data of majority class observations.

Undersampling methods are the methods that provide balance by reducing the majority class observations. Oversampling methods are the methods that provide balance by increasing the minority class observations. The most basic one of the undersampling methods is random undersampling (RUS) method. It is often used in literature to handle class imbalance problem.

For example, Chen et al. [1] reduced the aliasing artifacts and improved image quality by using a hybrid scheme in the form of RUS-based singular value decomposition and compressed sensing. Liu and Tsoumakas [2] used the RUS to improve the learning method of the Ensemble of Classifier Chains against class imbalance. Noise detection is a popular approach to select samples to be removed. Tomek Link [3] is one of these noise detection methods frequently used in the literature to select samples to be discarded in undersampling applications [4-6]. Edited Nearest Neighbor [7] is another noise detection method that can be used for undersampling. It uses three-nearest neighbor and single-nearest neighbor sequentially to reduce the number of samples. Laurikkala [8] proposed a neighborhood cleaning rule to undersample the majority class. The method uses Edited Nearest Neighbor to remove noisy samples not only in the majority class but also in the minority class.

Removal of observations to achieve class balance causes undesirable loss of information available. To minimize the loss of information, it would be more appropriate to select the observations by non-random methods. Many methods have been suggested in the literature for this purpose [9-13]. The purpose of this study is to use optimization to select observations that are to be removed in undersampling. Optimization is the process of achieving the most appropriate solution for a specific purpose under certain constraints. In other words, optimization is the process of obtaining the best result under given conditions. Different optimization techniques have been developed to solve the problems encountered. There are two types of algorithms for solving optimization problems. One of these is the classical algorithms that can obtain the optimal solution by scanning the entire solution space. The other is the heuristic algorithms which intuitively reach the solution in a short time without scanning the solution's entire space. Genetic Algorithm (GA) [14], Particle Swarm Optimization (PSO) [15], Differential Evolution Algorithm (DE) [16], and Artificial Bee Colony (ABC) [17] are some examples of heuristic algorithms.

Methods of optimization were used in various ways when applying methods of resampling. García and Herrera [18] have proposed eight different methods for selecting the observations to be taken in undersampling with evolutionary algorithms. Inspired by this method, Roshan and Asadi [19] increased the performance of bagging by detecting the best performing bonds with the multipurpose evolutionary algorithm. Yu et al. [20] improved the performance of the Support Vector Machines classifier in imbalanced class data sets by setting an optimized decision limit.

In the introductory section of this study, class imbalance problem and optimization methods are mentioned and related studies in the literature are given. In the second section, the problem of class imbalance is explained. Then, random sampling and sampling methods with optimization are introduced. In the third section, the performance criteria used in the model evaluation while

there is a class imbalance are explained. In the fourth section, models with different data sets and different classifiers are formed. In these models, performance results are obtained with and without resampling. In the fifth section, the averages of all results are given and the significance of the difference between these results is tested. Finally, the results are compared and suggestions are made in the sixth section.

2. Materials and Methods

2.1. Class imbalance problem

Algorithms, where the dependent variable is categorical and where these categories are estimated using certain independent variables are referred to as classification. In classification, the categories included in the dependent variable take the class name. Classification methods were developed to determine the order in the observation data and categorize the sample based on this order [21]. Most classification methods establish a model assuming that the number of observations of these classes are equal. In this case, when the number of observations in the classes varies, the models show bias in favor of the class with many observations. This is called class imbalance problem. In almost all actual data, the number of observations of classes is not equal, i.e., there is a class imbalance. However, the term class imbalance is used for situations where a significant imbalance rate exists. This is called between-classes imbalance and is a condition often encountered [22]. When there is an inter-classes imbalance, the class with fewer observations is called the positive class, and the class with more observations is called the negative class. There are some methods used to solve the problem of class imbalance. These methods include the following:

- In resampling methods, data derivation or data reduction may be applied until the data is balanced [23].
- Different weights can be given to the observations.
- Different weights can be given to the classes.
- By using ensemble algorithms, positive classes can be predicted more easily [24].

Resampling methods are methods in which the data set is modified to make the observation classes in the data set more balanced [25].

The resampling method can be grouped under three titles. These are oversampling, undersampling, and hybrid methods. In oversampling methods, the aim is to increase the balance of the positive class data in a certain way. In undersampling methods, some observations from

the majority class observations are supposed to be balanced either randomly or in a controlled way. In hybrid methods, undersampling and oversampling methods are used in conjunction.

2.2. Random undersampling

The most basic of undersampling methods is the random undersampling method. In this method, a certain amount of data is randomly selected from the negative class and the remainder is removed from the data set. Thus, balance can be ensured. RUS deletes the observation from the original data set. The procedure for RUS is given in Algorithm 1. Random observations from T_{neg} of the entered T training set are selected and removed. However, discarding existing information in this method causes loss of information. This is also undesirable. Consequently, RUS is a simple method of adjusting the balance of the T data set [25].

Algorithm 1: RUS (Exact Balance)

Input	:	Training dataset, T
Function	:	
		1. T_{neg} and $T_{pos} \leftarrow$ datasets belonging to classes
		2. N_{neg} and $N_{pos} \leftarrow$ number of observations in T_{neg} and T_{pos} respectively
		3. $L = N_{neg} - N_{pos}$ (Number of samples required for exact balance)
		4. $U \leftarrow L$ -length series randomly selected within $1, 2, \dots, N_{neg}$ series
		5. $T_{neg}^{un} \leftarrow T_{neg}$ after discarding samples selected in U
Output	:	$T_{un} \leftarrow T_{neg}^{un}$ and T_{pos} datasets together

2.3. Undersampling with optimization

The purpose of this method is to determine the most useful observations to be taken from samples of negative class. For this, it is necessary to determine the data set, which provides the best separation in the T training data set. Algorithm 2 specifies the degree to which the separation is accomplished. This algorithm is the objective function of the optimization method used in Algorithm 3. Vector Z with length N_{neg} to be tested in optimization are the parameters to be optimized. These parameters are set as $0 \leq Z_i \leq 1, i = 1, 2, \dots, N_{neg}$ in optimization method. Z parameters are rounded in the objective function. Thus, all parameters take values of either 0 or 1. This is because the method of optimization to be used may not optimize discrete parameters. Data sets of negative and positive classes are determined and two data sets are notated as T_{neg} and T_{pos} are obtained. Then, negative observations in T_{neg} , which have a value of 1 in the Z vector, are removed and T_{neg}^{un} is obtained. Then T_{neg}^{un} and T_{pos} are combined and the new undersampled data set, T^{un} is obtained. A decision tree ($cp = 0.01$), K , is trained in T_{neg}^{un} . The

performance of K in dataset T^{un} is used to determine the performance of the resampling method. For this, independent variables of T^{un} are used to obtain the predictions of K and area under ROC curve (AUC) is used to measure the performance. The subset T^{un} of T , maximizing the AUC level obtained from the estimation gives us the optimal subset. The performance of T^{un} obtained with this method depends on multiple factors. The optimization method to be used is, obviously, one of the factors. Different optimization methods may find different subsets. Iteration numbers in optimization methods may change the results. The type of the classifier (CART decision tree in this case) and the evaluation criteria (AUC in this case) are also important factors.

Undersampling with the optimization method optimizes too many parameters. Therefore, it takes too much time compared to resampling methods without optimization. Too many parameters need too many combinations and more iterations are required to optimize them. Each iteration also takes longer because the parameter number is high.

Algorithm 2: Objective Function of undersampling with optimization

Input	:	Train data set, T ; Z vector of length N_{neg} with values in range $[0,1]$
Function	:	<ol style="list-style-type: none"> 1. $Z = [Z + 0.5]$ (rounding) 2. T_{neg} and $T_{pos} \leftarrow$ datasets of classes 3. $T_{neg}^{un} \leftarrow T_{neg}$ observations corresponding to value of 1 in the vector Z 4. $T_{un} \leftarrow T_{neg}^{un}$ and T_{pos} datasets together 5. $K \leftarrow$ decision tree model trained in T^{un} 6. $Pr(Y = positive\ class X = x) \leftarrow$ positive class probabilities of K
Output	:	$Perf \leftarrow$ AUC calculated using $Pr(Y = positive\ class X = x)$

Algorithm 3: Undersampling with optimization

Input	:	Train data set, T
Function	:	<ol style="list-style-type: none"> 1. T_{neg} and $T_{pos} \leftarrow$ datasets of classes 2. N_{neg} and $N_{pos} \leftarrow$ number of observations in T_{neg} and T_{pos} respectively 3. Optimize vector $0 \leq Z_i \leq 1, i = 1, 2, \dots, N_{neg}$ using the objective function in Algorithm 2 3. $Z \leftarrow$ optimized parameters 4. $Z_{opt} = [Z + 0.5]$ (rounding) 5. $T_{neg}^{un} \leftarrow T_{neg}$ observations corresponding to the value of 1 in the vector Z_{opt}

Output	: $T_{un} \leftarrow T_{neg}^{un}$ and T_{pos} datasets together
--------	--

2.4 Performance criteria

In order to determine the classification models developed for a data set with class imbalance problem, the most suitable performance criterion should be selected. Although there are many performance criteria in the literature, F_1 , G-mean and AUC are frequently used in imbalanced data sets instead of misleading criteria such as accuracy or error rate [26, 27, 28, 29, 30]. F_1 and G-mean are calculated using confusion matrix. The confusion matrix is given in Table 1. Some of the performance criteria that were calculated through confusion matrix are presented in Table 2. Since accuracy is biased towards the negative class, imbalanced data sets produce biased results. While sensitivity and precision consider positive class performance to be very high, specificity gives the negative class more importance. G-mean is the geometric mean of sensitivity and specificity. F_1 score is the harmonic mean of sensitivity and precision. Since G-mean and F_1 consider both classes, they give a more realistic and general performance in imbalanced datasets.

Table 1: Confusion matrix

		Prediction	
		Positive	Negative
Truth	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Table 2: Some performance criteria used to evaluate classification models

Criterion	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
True positive rate (Sensitivity)	$\frac{TP}{TP + FN}$
True negative rate (Specificity)	$\frac{TN}{TN + FP}$
Positive prediction value (Precision)	$\frac{TP}{TP + FP}$
G-mean	$\sqrt{\frac{\text{Sensitivity} \times \text{Specificity}}{2}}$
F_1	$\frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Sensitivity}}}$

AUC is the most commonly used criterion for graphical performance. Calculation of AUC can be given as follows:

$$\overline{AUC} = \int_0^1 \overline{ROC}(t) dt \quad (1)$$

3. Simulation

Figure 1 shows the flow chart of forming the model with or without resampling. The study was conducted on simulation data and then 11 different data sets which are available online [31]. Table 3 shows the structural characteristics of the data sets. Undersampling methods were applied to achieve balance in the train dataset by removing observations from the negative class. Methods of undersampling used are; RUS, undersampling with genetic algorithm (US with GA), undersampling with differential evolution algorithm (US with DE), undersampling with an artificial bee colony (US with ABC), undersampling with particle flock optimization (US with PSO).

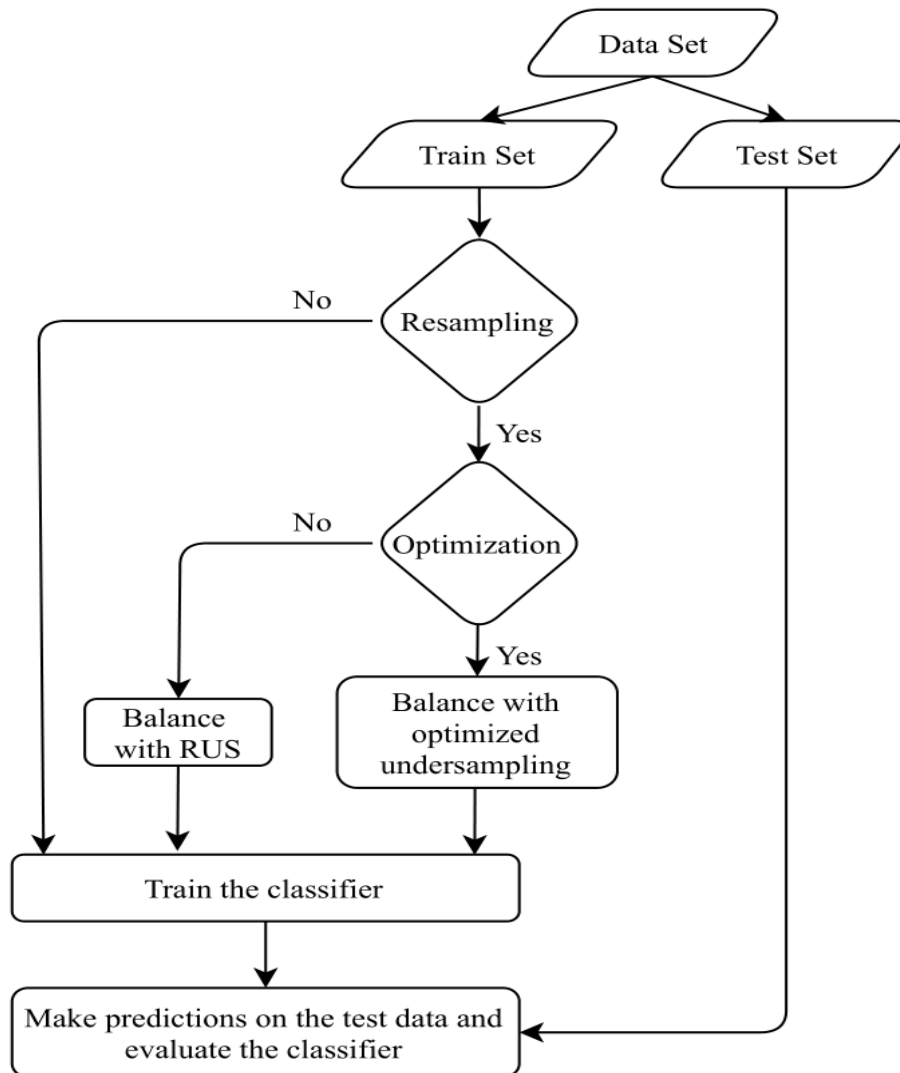


Figure 1: Flow chart of modelling with and without

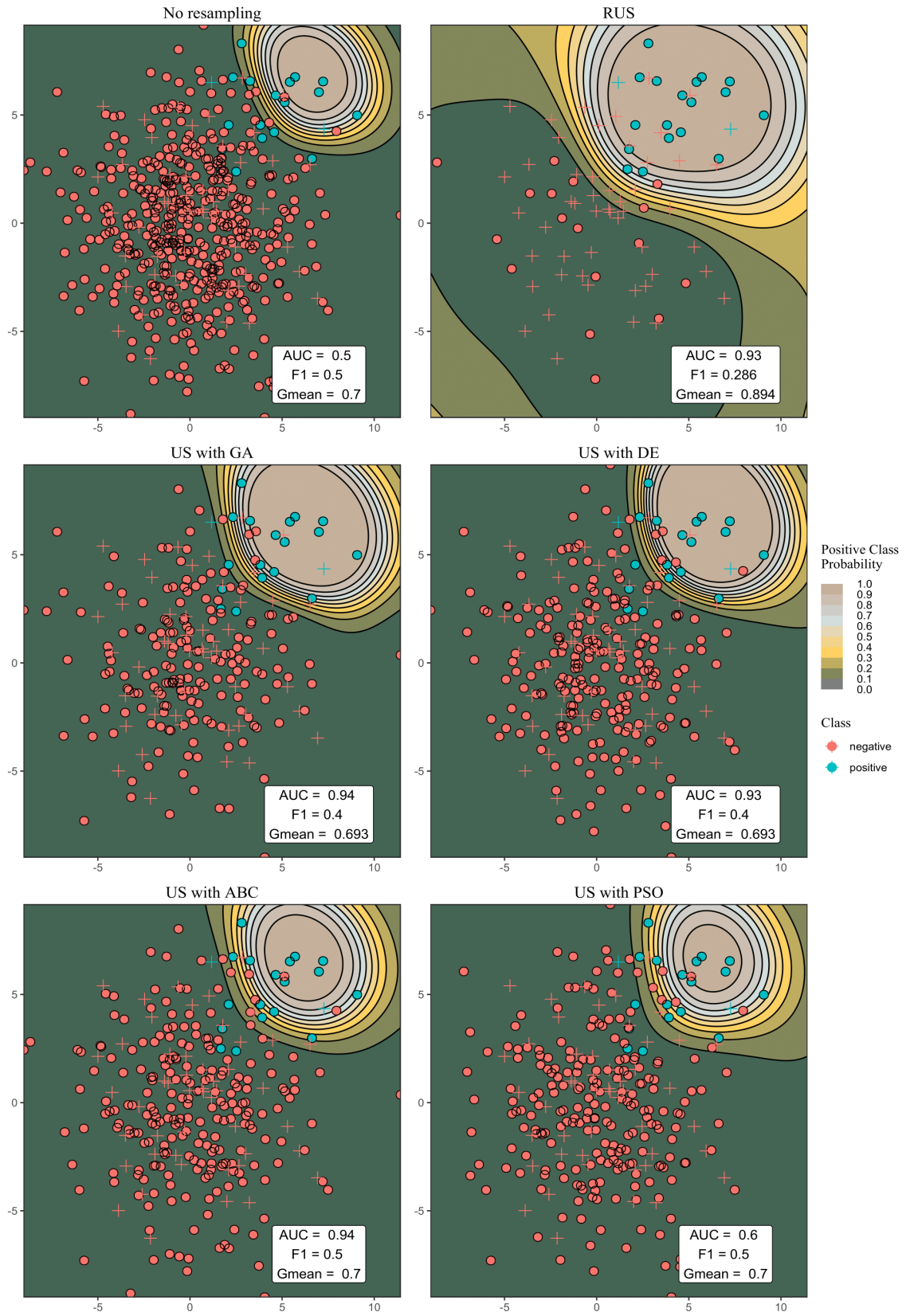


Figure 2: Decision boundaries of radial SVM in the simulation data set (“+” signs indicate test set, circles indicate training set)

The maximum number of iterations was set to 100 in all optimization methods. The class observations were made in the simulation study as two dimensional $X_{neg} \sim N(0,3)$ with length 500 and $X_{pos} \sim N(5,2)$ with length 20. The training and test data set were stratified as 90% and 10% respectively. The radial SVM model was formed on the imbalanced training set. Secondly, the balance was achieved with RUS, and the model was formed. Subsequently, four different optimization methods were used and undersamplings were applied. Figure 2 shows the decision boundaries and performances of the models. Although according to the original data set, there was no increase in F_1 score, US with GA and US with ABC reached 0.94 AUC. Compared to other methods, US with PSO was more inefficient in the simulation study. In the datasets undersampled by optimization methods, it is seen that less data is discarded from the majority class compared to the RUS method. If we consider that the US with GA, US with DE, and US with ABC methods perform close to or better than RUS, we see that we can achieve the desired results with less information loss. But the same is difficult to say for US with PSO.

Table 3: Structural properties of data sets

Dataset	Independent Variable Number	N	N_{pos}	N_{neg}	Imbalance Ratio (N_{neg}/N_{pos})
breast	9	277	81	196	2.420
bupa	5	345	145	200	1.379
cleveland	5	296	34	262	7.706
ecoli	5	335	34	301	8.853
haberman	3	305	81	224	2.765
Sonar	60	120	23	97	4.217
glass0	9	214	70	144	2.057
kyphosis	3	81	17	64	3.765
newthyroid1	5	215	35	180	5.143
Seeds	7	199	66	133	2.015
Vertebral	6	310	100	210	2.100

Eleven different data sets and 5 different classifiers were used to analyze the performances of these methods. AUC, F_1 and G-mean criteria were used to evaluate the performance of the models. The 10-fold cross-validity was repeated 10 times and the performance criteria were averaged. In addition, the performance of the methods was graded from 1 to 6 with $10 \times 10 = 100$ models set up for each case. The sequence numbers were properly synchronized when the same performances were achieved in this ranking. To give an example, all $(1 + 2 + 3) / 3 = 2$ were assigned rank numbers when the three methods were in the first place, and the subsequent methods were assigned 4-5-6 rank numbers. The obtained rank numbers were averaged for 100 models and added to the results.

In the study, J48 decision tree ($C = 0.25$, $M = 2$), K nearest neighborhood (KNN, $K = 5$), radial support vector machines (SVM, $\sigma = 1$, $C = 0.25$), naive bayes and logistic regression classifiers were used.

The AUC measurements of the J48 classifier are given in Table 4 and the average rank in Table 5. In the 11 data sets, the best results were obtained once without resampling, four times in RUS, three times in US with GA, one time in US with DE, and two times in US with ABC. The best ranking was achieved with US with GA, and the worst with US with PSO when the rankings were analyzed. While GA was the optimization method that gave the best result in AUC rankings, the method that gave the worst result was PSO.

Table 4: AUC values for J48 classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.6168±0.11	0.6069±0.1225	0.6047±0.1292	0.629±0.1082	0.6192±0.1145	0.6262±0.1214
bupa	0.6168±0.09	0.6285±0.1023	0.6126±0.0972	0.614±0.0981	0.6123±0.0962	0.617±0.0932
cleveland	0.5209±0.0575	0.5878±0.1374	0.5288±0.1267	0.5455±0.1177	0.5379±0.1145	0.525±0.1075
ecoli	0.7921±0.1858	0.7988±0.1271	0.7949±0.1511	0.7888±0.1471	0.803±0.1659	0.7587±0.1748
haberman	0.5727±0.0953	0.6269±0.0979	0.6137±0.1087	0.6147±0.1144	0.6193±0.1008	0.58±0.0995
Sonar	0.6982±0.1976	0.7125±0.1931	0.7627±0.1898	0.7598±0.1786	0.7164±0.2081	0.7492±0.1917
glass0	0.7939±0.0975	0.7874±0.0912	0.8049±0.1035	0.7973±0.089	0.7868±0.1028	0.7999±0.1031
kyphosis	0.5905±0.1948	0.7181±0.1765	0.6371±0.2016	0.6015±0.1819	0.6461±0.193	0.5886±0.1787
newthyroid1	0.9501±0.0899	0.937±0.0734	0.9569±0.0784	0.95±0.0808	0.94±0.0912	0.9547±0.0735
Seeds	0.918±0.0833	0.8965±0.0821	0.9024±0.079	0.9085±0.0709	0.9125±0.0811	0.9165±0.0762
Vertebral	0.8384±0.0862	0.8385±0.0755	0.8399±0.0814	0.8389±0.0792	0.8474±0.0863	0.8261±0.0924

Table 5: AUC mean ranks for J48 classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	3.68	3.565	3.75	3.2	3.445	3.36
bupa	3.56	3.24	3.615	3.525	3.545	3.515
cleveland	3.715	2.785	3.65	3.39	3.685	3.775
ecoli	3.245	3.58	3.51	3.66	3.35	3.655
haberman	3.855	2.98	3.215	3.42	3.31	4.22
Sonar	3.655	3.97	3.3	3.295	3.535	3.245
glass0	3.505	3.685	3.275	3.57	3.715	3.25
kyphosis	3.965	2.67	3.425	3.72	3.305	3.915
newthyroid1	3.02	4.205	3.29	3.535	3.535	3.415
Seeds	3.18	3.935	3.57	3.645	3.415	3.255
Vertebral	3.385	3.68	3.47	3.505	3.195	3.765
Mean	3.524	3.481	3.461	3.497	3.458	3.579

The F_1 measurements of the J48 classifier are given in Table 6 and the average rank in Table 7. In the 11 datasets, the best results were obtained ten times without resampling and one time in US with GA. In terms of rank averages, the best ranking was reached when no resampling was performed, while the worst ranking was achieved when RUS was applied. While the optimization method that gave the best result in F_1 rankings was ABC, the method that gave the worst result was GA.

Table 6: F_1 values for J48 classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.8206±0.046	0.7038±0.0929	0.7608±0.0756	0.7708±0.0749	0.755±0.084	0.7637±0.0753
bupa	0.7063±0.0706	0.6644±0.107	0.5852±0.1437	0.5763±0.1495	0.6267±0.1624	0.6363±0.1305
cleveland	0.9337±0.0162	0.7415±0.1415	0.9102±0.0424	0.9255±0.0258	0.9137±0.037	0.9173±0.0362
ecoli	0.9614±0.0196	0.8683±0.0629	0.954±0.0271	0.9549±0.0251	0.9533±0.027	0.9563±0.028
haberman	0.8084±0.0537	0.7604±0.075	0.7859±0.0684	0.7791±0.0895	0.7821±0.0734	0.7803±0.0716
Sonar	0.5043±0.2811	0.4947±0.2217	0.5827±0.2566	0.5709±0.2669	0.5506±0.2752	0.5758±0.2785
glass0	0.848±0.0655	0.7997±0.0696	0.8285±0.0821	0.8136±0.0798	0.8278±0.0729	0.8218±0.0814
kyphosis	0.8298±0.0831	0.7222±0.1746	0.7931±0.1185	0.7841±0.1371	0.7891±0.1377	0.7907±0.1235
newthyroid1	0.9837±0.0219	0.9601±0.0354	0.9787±0.0249	0.976±0.0244	0.9753±0.0276	0.9755±0.026
Seeds	0.89±0.0949	0.8404±0.1068	0.8523±0.099	0.8563±0.091	0.8784±0.0931	0.8662±0.0997
Vertebral	0.8668±0.0547	0.8377±0.0659	0.8496±0.0561	0.8429±0.0588	0.8637±0.0551	0.8416±0.0605

Table 7: F_1 mean ranks for J48 classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	1.945	4.82	3.66	3.325	3.615	3.635
bupa	2.48	3.07	4.21	4.325	3.38	3.535
cleveland	2.415	5.74	3.44	2.915	3.315	3.175
ecoli	2.625	5.835	3.16	3.215	3.245	2.92
haberman	2.555	4.22	3.45	3.57	3.525	3.68
Sonar	3.765	4.145	3.305	3.24	3.435	3.11
glass0	2.735	4.2	3.39	3.9	3.34	3.435
kyphosis	2.805	4.28	3.615	3.585	3.35	3.365
newthyroid1	3.025	4.345	3.305	3.495	3.385	3.445
Seeds	2.93	3.965	3.79	3.775	3.175	3.365
Vertebral	2.885	3.82	3.56	3.885	2.92	3.93
Mean	2.742	4.404	3.535	3.566	3.335	3.418

The G-mean measurements of the J48 classifier are given in Table 8 and the average rank in Table 9. In the 11 datasets, the best results were obtained two times without resampling, six times in RUS, two times in US with GA, and one time in US with DE in 11 data sets. When examining the averages of the rankings, it was seen that the best ranking was achieved when applying RUS, and the worst ranking was achieved when no resampling was applied. While GA was the optimization method that gave the best result in G-mean rankings, the method that gave the worst result was PSO.

Table 8: G-mean values for J48 classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.51±0.1689	0.5718±0.1185	0.5602±0.1376	0.5841±0.1292	0.5822±0.1175	0.5698±0.1514
bupa	0.5841±0.0857	0.5897±0.1153	0.5581±0.119	0.555±0.1207	0.5619±0.1108	0.569±0.1021
cleveland	0.0203±0.1001	0.5257±0.2173	0.1597±0.2633	0.1614±0.2603	0.1577±0.2675	0.1378±0.2534
ecoli	0.704±0.233	0.7902±0.1074	0.7423±0.1893	0.738±0.2026	0.7228±0.2308	0.7113±0.2235
haberman	0.4054±0.2724	0.6052±0.1263	0.5796±0.1355	0.5679±0.153	0.5795±0.1466	0.5355±0.1505
Sonar	0.6133±0.3036	0.6766±0.236	0.7191±0.2513	0.6998±0.2701	0.6691±0.2825	0.6947±0.2794
glass0	0.7812±0.0946	0.7706±0.0801	0.7916±0.1016	0.7765±0.1209	0.7741±0.0982	0.774±0.0999
kyphosis	0.3025±0.3502	0.6312±0.2596	0.4504±0.3615	0.3735±0.3632	0.4316±0.3576	0.3402±0.362
newthyroid1	0.9434±0.1019	0.9314±0.0818	0.9419±0.0994	0.9314±0.1059	0.9239±0.113	0.9338±0.0981
Seeds	0.9114±0.0753	0.8821±0.0835	0.8863±0.0783	0.8912±0.0709	0.9046±0.0735	0.8974±0.0789
Vertebral	0.7471±0.0987	0.8054±0.076	0.7801±0.0818	0.7854±0.0837	0.7559±0.0982	0.7657±0.0903

Table 9: G-mean rank means for J48 classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	4.22	3.47	3.67	3.12	3.275	3.245
bupa	3.345	3.125	3.625	3.83	3.53	3.545
cleveland	4.355	1.745	3.71	3.645	3.715	3.83
ecoli	3.485	3.4	3.395	3.53	3.55	3.64
haberman	4.04	2.84	3.295	3.545	3.245	4.035
Sonar	3.83	4.01	3.295	3.23	3.49	3.145
glass0	3.43	3.84	3.245	3.5	3.43	3.555
kyphosis	4.095	2.655	3.275	3.635	3.455	3.885
newthyroid1	2.965	4.035	3.295	3.575	3.605	3.525
Seeds	2.95	3.97	3.795	3.77	3.17	3.345
Vertebral	4.21	2.755	3.235	3.255	3.915	3.63
Mean	3.72	3.259	3.44	3.512	3.489	3.58

The AUC measurements of the KNN classifier are given in Table 10 and the average ranks are given in Table 11. In the 11 datasets, the best results were obtained five times without resampling, four times with RUS, and two times in US with GA. When examining the averages of the rankings, it was seen that the best ranking was achieved without resampling and the worst ranking was achieved when RUS was applied. While the optimization method which gave the best result in AUC rankings was ABC, the method that gave the worst result was PSO.

Table 10: AUC values for KNN classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.6473±0.1052	0.6617±0.1104	0.6553±0.101	0.6584±0.0966	0.6573±0.0943	0.6611±0.1091
bupa	0.6614±0.0842	0.6648±0.0862	0.6518±0.0853	0.6613±0.085	0.6611±0.0874	0.6573±0.0785
cleveland	0.5378±0.1342	0.5581±0.1596	0.5418±0.1415	0.551±0.1387	0.5492±0.1483	0.5322±0.1363
ecoli	0.8913±0.0988	0.9134±0.0595	0.9092±0.0768	0.9156±0.0737	0.91±0.0804	0.9099±0.0764
haberman	0.6531±0.0961	0.6529±0.1051	0.6443±0.1013	0.651±0.1044	0.648±0.1051	0.6456±0.105
Sonar	0.917±0.1157	0.8745±0.1331	0.9036±0.127	0.8969±0.1315	0.9046±0.1217	0.9023±0.1257
glass0	0.8749±0.0823	0.8528±0.0779	0.8541±0.0834	0.8526±0.0819	0.864±0.0857	0.8454±0.0867
kyphosis	0.6472±0.2394	0.6596±0.2136	0.6611±0.237	0.647±0.2407	0.6448±0.2327	0.6402±0.2424
newthyroid1	0.9919±0.0409	0.9793±0.0459	0.9878±0.0405	0.9878±0.0288	0.9869±0.0414	0.987±0.0412
Seeds	0.9693±0.0391	0.9725±0.0336	0.9734±0.0337	0.9731±0.0319	0.969±0.0395	0.97±0.0354
Vertebral	0.9092±0.0525	0.9036±0.0477	0.9044±0.0548	0.9074±0.0495	0.9086±0.0541	0.907±0.0505

Table 11: AUC rank means for KNN classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	3.88	3.29	3.505	3.435	3.61	3.28
bupa	3.36	3.4	3.775	3.415	3.535	3.515
cleveland	3.52	3.325	3.53	3.47	3.335	3.82
ecoli	3.555	3.705	3.435	3.23	3.44	3.635
haberman	3.31	3.42	3.785	3.42	3.49	3.575
Sonar	2.745	4.41	3.385	3.635	3.31	3.515
glass0	2.455	3.81	3.75	3.895	3.03	4.06
kyphosis	3.49	3.495	3.325	3.495	3.575	3.62
newthyroid1	2.69	4.28	3.48	3.61	3.49	3.45
Seeds	3.53	3.545	3.435	3.29	3.465	3.735
Vertebral	3.155	3.855	3.62	3.63	3.21	3.53
Mean	3.245	3.685	3.548	3.502	3.408	3.612

The F_1 measurements of the KNN classifier are given in Table 12, and the average ranks are given in Table 13. In the 11 datasets, the best results were obtained ten times without resampling and one time in RUS. When examining the average rankings, it was seen that the best ranking was achieved without resampling and the worst ranking was achieved when RUS was applied. While the optimization method which gave the best result in F_1 rankings was ABC, the method that gave the worst result was DE.

Table 12: F_1 values for KNN classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.8164±0.0446	0.7128±0.0781	0.7607±0.0682	0.7591±0.072	0.7488±0.0757	0.7597±0.0773
bupa	0.7151±0.064	0.6658±0.0834	0.6084±0.1053	0.5954±0.1084	0.6768±0.1044	0.6336±0.1002
cleveland	0.9341±0.0134	0.7179±0.0885	0.9036±0.0336	0.9041±0.0334	0.9033±0.0306	0.9092±0.0286
ecoli	0.9561±0.0236	0.8896±0.0506	0.9525±0.0244	0.9527±0.0277	0.9528±0.0261	0.953±0.027
haberman	0.8228±0.0467	0.7198±0.0718	0.7768±0.0571	0.7694±0.0669	0.7742±0.0657	0.771±0.0621
Sonar	0.7201±0.2115	0.5336±0.1536	0.6404±0.2064	0.6122±0.1911	0.6593±0.2196	0.62±0.1903
glass0	0.8368±0.0693	0.771±0.0898	0.7869±0.1011	0.7832±0.0854	0.8124±0.0847	0.7744±0.0918
kyphosis	0.8549±0.0627	0.6627±0.1778	0.8271±0.0997	0.8094±0.1187	0.8239±0.0966	0.806±0.1107
newthyroid1	0.9776±0.0213	0.961±0.0343	0.9757±0.0246	0.9758±0.0261	0.9752±0.026	0.9743±0.0263
Seeds	0.8397±0.1078	0.8703±0.0954	0.8586±0.0993	0.8619±0.1029	0.8522±0.0981	0.8566±0.0992
Vertebral	0.8795±0.0507	0.8464±0.056	0.8551±0.0599	0.8589±0.0573	0.8741±0.0543	0.8657±0.0562

Table 13: F_1 rank means for KNN classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	1.815	4.795	3.565	3.545	3.79	3.49
bupa	1.91	3.205	4.425	4.78	2.73	3.95
cleveland	1.6	6	3.465	3.425	3.405	3.105
ecoli	2.625	5.86	3.275	3.06	3.105	3.075
haberman	1.625	5.4	3.345	3.62	3.445	3.565
Sonar	2.14	5.055	3.27	3.775	3.09	3.67
glass0	2.13	4.355	3.68	3.855	2.895	4.085
kyphosis	2.47	5.285	3.14	3.315	3.23	3.56
newthyroid1	3.155	4.355	3.36	3.315	3.4	3.415
Seeds	3.855	3.18	3.47	3.42	3.55	3.525
Vertebral	2.665	4.405	3.925	3.625	2.995	3.385
Mean	2.363	4.718	3.538	3.612	3.24	3.53

The G-mean measurements of the KNN classifier are given in Table 14, and the average ranks are given in Table 15. In the 11 data sets, the best results were achieved once without resampling, seven times in RUS, and two times in US with DE. When the rank averages are examined, it was seen that the best ranking is reached when RUS is applied and the worst ranking is reached when no resampling is applied. While the optimization method which gave the best result in G-mean rankings was DE, the method that gave the worst result was PSO.

Table 14: G-mean values for KNN classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.491±0.1433	0.6063±0.1029	0.5965±0.121	0.6024±0.1235	0.5889±0.1112	0.5852±0.1324
bupa	0.6066±0.0887	0.62±0.0858	0.6017±0.0882	0.5993±0.0876	0.6065±0.0915	0.6116±0.0845
cleveland	0±0	0.4971±0.2218	0.0627±0.172	0.0399±0.1362	0.0227±0.115	0.036±0.1321

ecoli	0.6932±0.2592	0.8305±0.1195	0.7691±0.2262	0.7959±0.1877	0.7573±0.2195	0.7694±0.2189
haberman	0.4515±0.1882	0.5998±0.1053	0.5593±0.1238	0.5744±0.1242	0.5837±0.123	0.5486±0.1303
Sonar	0.8406±0.1701	0.7461±0.1368	0.8062±0.1682	0.7983±0.1487	0.8127±0.1711	0.8013±0.165
glass0	0.7489±0.1184	0.7562±0.0934	0.7609±0.1004	0.7613±0.0898	0.7484±0.1086	0.7436±0.0973
kyphosis	0.0932±0.2581	0.538±0.3016	0.2994±0.3863	0.3243±0.3748	0.2402±0.3525	0.2942±0.357
newthyroid1	0.8628±0.1572	0.8996±0.1166	0.8929±0.1299	0.8905±0.1346	0.8896±0.1277	0.8888±0.1323
Seeds	0.8704±0.0871	0.9037±0.0733	0.8915±0.0774	0.8963±0.0809	0.884±0.0806	0.8901±0.0772
Vertebral	0.8097±0.082	0.8301±0.0622	0.8246±0.0711	0.8324±0.0637	0.814±0.0776	0.8286±0.0722

Table 15: G-mean rank means for KNN classifier

Dataset	No resampling	RUS	GA	DE	ABC	PSO
breast	4.765	3.25	3.135	3.15	3.31	3.39
bupa	3.5	3.115	3.605	3.745	3.55	3.485
cleveland	4.095	1.38	3.78	3.875	3.975	3.895
ecoli	3.905	3.205	3.49	3.275	3.65	3.475
haberman	4.91	2.84	3.435	3.185	2.955	3.675
Sonar	2.17	4.995	3.28	3.775	3.105	3.675
glass0	3.34	3.565	3.325	3.365	3.54	3.865
kyphosis	4.315	2.585	3.395	3.35	3.785	3.57
newthyroid1	3.425	4.015	3.38	3.345	3.4	3.435
Seeds	3.885	3.17	3.46	3.42	3.55	3.515
Vertebral	3.76	3.38	3.59	3.275	3.665	3.33
Mean	3.825	3.227	3.443	3.433	3.499	3.574

The AUC measurements of the radial SVM classifier are given in Table 16, and the average ranks are given in Table 17. In the 11 datasets, the best results were obtained five times without resampling, four times in RUS, and two times in US with GA. When the averages of the rankings were examined, it was seen that the best ranking was achieved without resampling and the worst ranking was achieved when RUS was applied. While the optimization method which gave the best result in AUC rankings was ABC, the method that gave the worst result was PSO.

Table 16: AUC values for radial SVM classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.6454±0.1174	0.5424±0.1658	0.6282±0.1137	0.6305±0.1238	0.6092±0.1343	0.6226±0.1318
bupa	0.7124±0.0737	0.6899±0.0824	0.6839±0.081	0.6822±0.0833	0.7022±0.0734	0.6884±0.0779
cleveland	0.6946±0.1735	0.5804±0.2161	0.6666±0.1737	0.6711±0.168	0.6599±0.1717	0.6379±0.1879
ecoli	0.8878±0.1397	0.9284±0.0562	0.9062±0.11	0.9088±0.1041	0.8936±0.1273	0.903±0.1093
haberman	0.6543±0.1122	0.6688±0.1087	0.6884±0.1157	0.6928±0.1084	0.6965±0.1052	0.6904±0.1055
Sonar	0.6496±0.3671	0.4933±0.2564	0.3483±0.3553	0.4027±0.3668	0.481±0.3894	0.3379±0.34
glass0	0.8801±0.0767	0.8676±0.0796	0.8641±0.0795	0.8674±0.0809	0.8743±0.0791	0.8716±0.0772
kyphosis	0.8211±0.1889	0.7748±0.2553	0.8464±0.1699	0.8277±0.1926	0.8426±0.1741	0.7962±0.2171
newthyroid1	0.9945±0.0127	0.9955±0.0141	0.9956±0.0156	0.9966±0.0114	0.9971±0.012	0.996±0.0123
Seeds	0.9852±0.0206	0.9828±0.0231	0.9844±0.0214	0.9813±0.0279	0.9838±0.0218	0.9842±0.0227
Vertebral	0.8843±0.0664	0.8831±0.0585	0.8841±0.0645	0.8894±0.0602	0.8827±0.0646	0.8813±0.063

Table 17: AUC rank means for radial SVM classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	2.99	4.085	3.57	3.315	3.67	3.37
bupa	2.495	3.755	3.93	3.9	3.02	3.9
cleveland	2.685	3.855	3.605	3.485	3.655	3.715
ecoli	3.155	3.99	3.4	3.345	3.565	3.545

haberman	4.295	4.105	3.175	3.215	2.92	3.29
Sonar	2.645	3.11	3.99	3.85	3.535	3.87
glass0	2.92	3.64	4.01	3.625	3.165	3.64
kyphosis	3.41	3.91	3.125	3.505	3.275	3.775
newthyroid1	3.755	3.51	3.445	3.44	3.39	3.46
Seeds	3.285	3.675	3.435	3.73	3.505	3.37
Vertebral	3.085	4.015	3.515	3.135	3.385	3.865
Mean	3.156	3.786	3.564	3.504	3.371	3.618

The F_1 measurements of the radial SVM classifier are given in Table 18 and the average ranks are given in Table 19. In the 11 datasets, the best results were obtained nine times without resampling and two times in RUS. When the averages of the rankings were examined, it was seen that the best ranking was achieved without resampling and the worst ranking was achieved when RUS was applied. While the optimization method which gave the best result in F_1 rankings was ABC, the method that gave the worst result was DE.

Table 18: F_1 values for radial SVM classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.8194±0.0214	0.5094±0.1433	0.7538±0.127	0.7804±0.0703	0.7335±0.1058	0.7759±0.0791
bupa	0.753±0.0551	0.7251±0.06	0.5706±0.156	0.5056±0.1405	0.6801±0.1355	0.6579±0.1484
cleveland	0.9352±0.0138	0.597±0.1404	0.9313±0.0197	0.9314±0.0166	0.9321±0.0157	0.9299±0.0193
ecoli	0.9552±0.0211	0.9344±0.0313	0.9548±0.0242	0.9529±0.0251	0.9536±0.0228	0.9548±0.0234
haberman	0.84±0.0313	0.7209±0.0815	0.7989±0.0522	0.7946±0.06	0.7956±0.062	0.7996±0.0599
Sonar	0±0	0.2256±0.1296	0±0	0±0	0±0	0±0
glass0	0.865±0.0446	0.8364±0.0753	0.8303±0.0676	0.8327±0.066	0.8588±0.0546	0.835±0.07
kyphosis	0.8749±0.0802	0.7525±0.1731	0.8602±0.1082	0.858±0.1002	0.8582±0.1009	0.8478±0.0995
newthyroid1	0.9779±0.0259	0.9202±0.0464	0.9369±0.0461	0.9425±0.041	0.9445±0.0451	0.9426±0.0442
Seeds	0.8808±0.0942	0.8906±0.0961	0.882±0.091	0.8792±0.0918	0.877±0.0921	0.8854±0.0866
Vertebral	0.8796±0.0401	0.8493±0.0577	0.8592±0.0498	0.8614±0.0488	0.8748±0.0466	0.8643±0.0479

Table 19: F_1 rank means for radial SVM classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	1.97	5.79	3.38	3.03	3.715	3.115
bupa	1.95	2.86	4.49	5.225	2.98	3.495
cleveland	2.68	6	3.025	3.145	3.04	3.11
ecoli	2.915	5.03	3.155	3.455	3.325	3.12
haberman	1.86	5.44	3.43	3.48	3.46	3.33
Sonar	3.88	1.6	3.88	3.88	3.88	3.88
glass0	2.59	3.755	4.14	3.855	2.735	3.925
kyphosis	2.925	4.71	3.13	3.285	3.305	3.645
newthyroid1	1.87	4.645	3.775	3.605	3.485	3.62
Seeds	3.55	3.29	3.49	3.6	3.65	3.42
Vertebral	2.56	4.425	3.925	3.72	2.815	3.555
Mean	2.614	4.322	3.62	3.662	3.308	3.474

The G-mean measurements of the radial SVM classifier are given in Table 20, and the average ranks are given in Table 21. The best results were obtained two times without resampling and nine times in the RUS. When the rank averages were examined, it was seen that the best ranking was achieved when RUS was applied and the worst ranking was achieved when no

resampling was applied. While the optimization method which gave the best result in G-mean rankings was ABC, the method that gave the worst result was PSO.

Table 20: G-mean values for radial SVM classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.1087±0.1739	0.4599±0.1615	0.3307±0.2134	0.3666±0.2347	0.3846±0.2455	0.3167±0.2392
bupa	0.6257±0.0824	0.6185±0.0792	0.5701±0.1112	0.5489±0.1041	0.6068±0.0874	0.5926±0.1067
cleveland	0.0049±0.049	0.473±0.2252	0.0627±0.1712	0.0531±0.1613	0.0444±0.1523	0.0317±0.1265
ecoli	0.549±0.2918	0.7246±0.2439	0.6533±0.2643	0.6579±0.2568	0.6091±0.2776	0.6315±0.2772
haberman	0.2752±0.2021	0.6192±0.1089	0.5344±0.1509	0.5702±0.1236	0.5697±0.1311	0.5288±0.1673
Sonar	0±0	0.0294±0.1017	0±0	0±0	0±0	0±0
glass0	0.7377±0.1137	0.7699±0.1084	0.7463±0.1042	0.7568±0.0974	0.7544±0.111	0.7468±0.1112
kyphosis	0.3236±0.3847	0.6165±0.3033	0.4577±0.4007	0.4429±0.3898	0.3801±0.3967	0.3884±0.3827
newthyroid1	0.9718±0.0435	0.924±0.0429	0.9368±0.051	0.9418±0.0456	0.9413±0.0527	0.9424±0.0453
Seeds	0.9027±0.0779	0.9163±0.0772	0.9066±0.0742	0.9025±0.0757	0.9013±0.0759	0.9072±0.0726
Vertebral	0.7749±0.0852	0.7962±0.0781	0.7874±0.0743	0.7941±0.0781	0.7844±0.0803	0.7881±0.075

Table 21: G-mean rank means for radial SVM classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	5.085	2.545	3.555	3.165	3.06	3.59
bupa	2.82	3.19	3.865	4.345	3.26	3.52
cleveland	4.095	1.505	3.785	3.8	3.865	3.95
ecoli	3.85	3.385	3.34	3.46	3.62	3.345
haberman	5.6	2.36	3.455	3.03	3	3.555
Sonar	3.54	3.3	3.54	3.54	3.54	3.54
glass0	3.67	3.125	3.78	3.395	3.36	3.67
kyphosis	3.86	2.865	3.37	3.51	3.645	3.75
newthyroid1	2	4.555	3.745	3.585	3.515	3.6
Seeds	3.585	3.17	3.5	3.62	3.675	3.45
Vertebral	3.755	3.185	3.57	3.275	3.555	3.66
Mean	3.805	3.017	3.591	3.52	3.463	3.603

The AUC measurements of the Naive Bayes classifier are given in Table 22, and the average rank in Table 23. In the 11 datasets, the best results were obtained seven times without resampling, two times with US with GA, and two times with US with PSO. When the averages of the rankings were examined, it was seen that the best ranking was achieved without resampling and the worst ranking was reached when RUS was applied. While the optimization method which gave the best results in AUC rankings was ABC, the method that gave the worst results was DE.

Table 22: AUC values for Naive Bayes classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.72±0.0998	0.7156±0.1009	0.7253±0.1017	0.7215±0.098	0.7235±0.0992	0.7239±0.0962
bupa	0.6041±0.0917	0.5989±0.0938	0.6019±0.0929	0.5898±0.089	0.6032±0.0897	0.6025±0.0889
cleveland	0.6386±0.1591	0.6314±0.1702	0.631±0.1591	0.6321±0.1621	0.633±0.1553	0.6206±0.1584
ecoli	0.92±0.0702	0.9104±0.0808	0.9184±0.0719	0.9199±0.0709	0.9188±0.0708	0.9168±0.0714
haberman	0.6865±0.1091	0.6782±0.1193	0.6816±0.11	0.6796±0.1132	0.6832±0.109	0.6856±0.112
Sonar	0.8886±0.1289	0.8916±0.112	0.8919±0.1248	0.8891±0.1347	0.8896±0.1241	0.8895±0.1173
glass0	0.8926±0.0681	0.8664±0.072	0.8698±0.0827	0.8722±0.0656	0.8844±0.0739	0.8635±0.081
kyphosis	0.7975±0.1888	0.7869±0.1861	0.8001±0.1917	0.777±0.1946	0.8011±0.1902	0.8086±0.1768
newthyroid1	0.9674±0.0399	0.9483±0.0531	0.9519±0.0476	0.9503±0.0514	0.9561±0.043	0.9575±0.043
Seeds	0.9646±0.0385	0.9615±0.04	0.9631±0.0396	0.9628±0.0391	0.9637±0.0393	0.9629±0.0385
Vertebral	0.8397±0.0748	0.8383±0.0772	0.8374±0.0758	0.8406±0.0756	0.8388±0.074	0.8439±0.0725

Table 23: AUC rank means for Naive Bayes classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	3.77	4.035	3.05	3.515	3.355	3.275
bupa	3.295	3.635	3.43	3.72	3.46	3.46
cleveland	3.03	3.455	3.615	3.385	3.645	3.87
ecoli	3.285	3.715	3.51	3.3	3.39	3.8
haberman	3.225	3.62	3.54	3.705	3.525	3.385
Sonar	3.515	3.635	3.42	3.4	3.52	3.51
glass0	2.855	3.755	3.675	3.685	3.04	3.99
kyphosis	3.4	3.68	3.455	3.72	3.395	3.35
newthyroid1	2.935	3.78	3.685	3.735	3.455	3.41
Seeds	3.255	3.765	3.505	3.51	3.4	3.565
Vertebral	3.53	3.695	3.86	3.19	3.675	3.05
Mean	3.281	3.706	3.522	3.533	3.442	3.515

The F_1 measurements of the Naive Bayes classifier are given in Table 24 and the average rank in Table 25. In the 11 datasets, the best results were obtained ten times without resampling and one time with US with PSO. When the averages of the rankings were examined, it was seen that the best ranking was achieved without resampling and the worst ranking was reached when RUS was applied. While the optimization method which gave the best result in F_1 rankings gave ABC, the method that gave the worst result was DE.

Table 24: F_1 values for Naive Bayes classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.8257±0.0372	0.7982±0.0712	0.8211±0.0564	0.8185±0.059	0.8212±0.0545	0.821±0.0512
bupa	0.6746±0.0753	0.604±0.1004	0.526±0.1055	0.5153±0.0964	0.6229±0.1117	0.5537±0.1141
cleveland	0.9266±0.0302	0.7282±0.0763	0.8877±0.0415	0.8864±0.0406	0.8858±0.0422	0.891±0.0399
ecoli	0.937±0.0355	0.9009±0.0467	0.9284±0.0374	0.924±0.0405	0.9299±0.0382	0.9291±0.0386
haberman	0.8311±0.0406	0.771±0.0647	0.7981±0.0563	0.802±0.0546	0.7977±0.0563	0.7991±0.0495
Sonar	0.6375±0.2684	0.5887±0.197	0.6117±0.2594	0.6477±0.2398	0.6293±0.2589	0.6513±0.2555
glass0	0.8051±0.0834	0.7306±0.1047	0.7286±0.1121	0.7378±0.0969	0.7728±0.1121	0.73±0.101
kyphosis	0.8899±0.0626	0.8093±0.1145	0.872±0.0905	0.8549±0.0906	0.8725±0.0967	0.8618±0.09
newthyroid1	0.9695±0.0321	0.9493±0.0402	0.9573±0.0337	0.9555±0.0339	0.9602±0.0309	0.9594±0.029
Seeds	0.8549±0.0914	0.8424±0.0987	0.8502±0.0896	0.8469±0.0895	0.8482±0.0892	0.8506±0.0925
Vertebral	0.8143±0.0649	0.7939±0.0698	0.7983±0.0664	0.7998±0.0697	0.8079±0.0679	0.8043±0.0648

Table 25: F_1 rank means for Naive Bayes classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	3.295	4.235	3.33	3.375	3.385	3.38
bupa	1.645	3.185	4.575	4.67	2.775	4.15
cleveland	1.375	5.95	3.4	3.525	3.52	3.23
ecoli	2.255	5.41	3.25	3.715	3.165	3.205
haberman	1.9	4.765	3.685	3.495	3.65	3.505
Sonar	3.2	4.265	3.65	3.19	3.47	3.225
glass0	2.225	3.945	4.095	3.89	2.87	3.975
kyphosis	2.73	4.715	3.255	3.625	3.19	3.485
newthyroid1	2.72	4.045	3.575	3.76	3.455	3.445
Seeds	3.255	3.77	3.455	3.57	3.565	3.385
Vertebral	2.79	3.965	3.84	3.755	3.215	3.435
Mean	2.49	4.386	3.646	3.688	3.296	3.493

G-mean measurements of the Naive Bayes classifier are given in Table 26 and average ranks are given in Table 27. In the 11 datasets, the best results were obtained four times without resampling and seven times in the RUS. In terms of rank averages, the best ranking was achieved by RUS, and the worst by US with GA. While the optimization method which gave the best result in G-mean rankings was ABC, the method that gave the worst result was the GA.

Table 26: G-mean values for Naive Bayes classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.3682±0.2003	0.6469±0.1213	0.5802±0.1327	0.6015±0.1294	0.5918±0.1413	0.591±0.1232
bupa	0.5828±0.0885	0.5844±0.0925	0.5455±0.0842	0.5413±0.0799	0.5708±0.0908	0.5572±0.0867
cleveland	0.2689±0.294	0.5469±0.207	0.3662±0.2895	0.3865±0.2796	0.3542±0.2772	0.3022±0.295
ecoli	0.7919±0.1345	0.8374±0.1012	0.8026±0.1306	0.8015±0.137	0.799±0.1292	0.8018±0.1356
haberman	0.467±0.1733	0.5992±0.1322	0.5502±0.14	0.5735±0.1484	0.5827±0.1287	0.569±0.1374
Sonar	0.7262±0.26	0.7672±0.1661	0.7163±0.266	0.7496±0.2347	0.7246±0.2469	0.7499±0.2397
glass0	0.7849±0.0842	0.7363±0.0886	0.733±0.0974	0.7414±0.0842	0.7652±0.1006	0.7314±0.0952
kyphosis	0.3715±0.3912	0.6034±0.3323	0.4756±0.3999	0.4635±0.3887	0.4601±0.4029	0.44±0.4073
newthyroid1	0.9632±0.0462	0.9512±0.0378	0.9586±0.0319	0.9569±0.0319	0.9587±0.0341	0.959±0.0312
Seeds	0.8903±0.0719	0.8838±0.0771	0.8895±0.0691	0.8871±0.0707	0.8866±0.0691	0.8877±0.0726
Vertebral	0.7768±0.0757	0.77±0.0711	0.77±0.0733	0.7728±0.0747	0.7738±0.0762	0.7733±0.0722

Table 27: G-mean rank means for Naive Bayes classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	5.435	2.48	3.41	3.205	3.245	3.225
bupa	2.865	2.955	4.025	4.03	3.485	3.64
cleveland	3.615	2.29	3.69	3.565	3.835	4.005
ecoli	2.915	3.97	3.49	3.745	3.375	3.505
haberman	4.94	2.8	3.625	3.155	3.155	3.325
Sonar	3.245	4.065	3.66	3.235	3.53	3.265
glass0	2.535	3.74	3.99	3.78	2.97	3.985
kyphosis	3.69	3.195	3.43	3.55	3.51	3.625
newthyroid1	2.81	3.995	3.525	3.74	3.485	3.445
Seeds	3.255	3.77	3.455	3.57	3.565	3.385
Vertebral	3.14	3.795	3.69	3.645	3.325	3.405
Mean	3.495	3.369	3.635	3.565	3.407	3.528

The AUC measurements of the Logistic Regression classifier are given in Table 28, and the average rank in Table 29. In the 11 datasets, the best results were obtained five times without resampling, one time in US with ABC and five times in US with PSO. When the averages of the rankings were examined, it was seen that the best ranking was achieved without resampling and the worst ranking was achieved when RUS was applied. While the optimization method which was the best results in AUC rankings was ABC, the method that gave the worst results was DE.

Table 28: AUC values for logistic regression classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.7215±0.1073	0.7128±0.105	0.7224±0.1056	0.7252±0.1096	0.7202±0.1072	0.7225±0.1064
bupa	0.7107±0.0884	0.7092±0.0903	0.7042±0.0886	0.7036±0.0896	0.7077±0.0887	0.7047±0.0884
cleveland	0.7403±0.1578	0.7225±0.1555	0.7328±0.1556	0.7363±0.155	0.7377±0.1597	0.7359±0.1527
ecoli	0.9067±0.0708	0.8986±0.0737	0.9048±0.0732	0.9038±0.0723	0.9056±0.0719	0.9036±0.0718
haberman	0.6807±0.1273	0.6784±0.1221	0.6807±0.1348	0.6806±0.1225	0.6792±0.1232	0.6814±0.1272
Sonar	0.7314±0.1561	0.5458±0.212	0.6897±0.1929	0.5949±0.1948	0.6628±0.1994	0.6175±0.1964

glass0	0.8216±0.0858	0.8114±0.086	0.8134±0.0861	0.8106±0.0866	0.8225±0.083	0.8156±0.0865
kyphosis	0.8357±0.1641	0.8168±0.1701	0.8361±0.1628	0.8296±0.1722	0.8357±0.1632	0.8395±0.1625
newthyroid1	0.9778±0.054	0.9878±0.0442	0.9864±0.0513	0.985±0.0422	0.9861±0.0365	0.9932±0.0316
Seeds	0.9888±0.0336	0.9719±0.0486	0.9698±0.0526	0.9752±0.0503	0.9826±0.0379	0.9833±0.0363
Vertebral	0.9312±0.0425	0.9305±0.0402	0.9302±0.0442	0.932±0.0433	0.9299±0.0445	0.9319±0.0421

Table 29: AUC rank means for logistic regression classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	3.485	4	3.405	3.295	3.45	3.365
bupa	3.14	3.38	3.62	3.795	3.5	3.565
cleveland	3.115	3.895	3.8	3.4	3.31	3.48
ecoli	3.195	3.785	3.485	3.575	3.42	3.54
haberman	3.455	3.685	3.285	3.475	3.555	3.545
Sonar	2.655	4.27	3.13	3.985	3.31	3.65
glass0	3.195	3.55	3.695	3.77	3.13	3.66
kyphosis	3.31	3.85	3.455	3.58	3.52	3.285
newthyroid1	3.68	3.51	3.355	3.61	3.57	3.275
Seeds	2.995	3.875	3.79	3.57	3.29	3.48
Vertebral	3.47	3.595	3.48	3.335	3.71	3.41
Mean	3.245	3.763	3.5	3.581	3.433	3.478

The F_1 measures of the Logistic Regression classifier are given in Table 30, and the average rank in Table 31. In the 11 datasets, the best results were obtained ten times without resampling and one time with US with PSO. When the averages of the rankings were examined, it was seen that the best ranking was achieved without resampling and the worst ranking was reached when RUS was applied. While the optimization method which gave the best result in F_1 rankings was ABC, the method that gave the worst result was DE.

Table 30: F_1 values for Logistic Regression classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.8355±0.0474	0.7303±0.0773	0.785±0.0733	0.786±0.0807	0.7734±0.0833	0.781±0.0824
bupa	0.7337±0.0657	0.6787±0.08	0.584±0.0908	0.5675±0.1012	0.6892±0.1035	0.6255±0.0915
cleveland	0.9412±0.0142	0.7924±0.0638	0.9264±0.0317	0.9274±0.03	0.9252±0.0302	0.929±0.0262
ecoli	0.9573±0.0191	0.8928±0.0526	0.9534±0.0216	0.953±0.0223	0.9569±0.0202	0.9532±0.0213
haberman	0.8449±0.0253	0.7911±0.073	0.8417±0.0433	0.8364±0.055	0.8376±0.0458	0.842±0.0457
Sonar	0.4384±0.243	0.326±0.184	0.411±0.2199	0.3397±0.1981	0.4052±0.2274	0.3542±0.2112
glass0	0.8265±0.0646	0.7616±0.076	0.7734±0.0761	0.7651±0.0804	0.807±0.064	0.7778±0.0803
kyphosis	0.8743±0.0901	0.7844±0.1455	0.8561±0.1082	0.8579±0.1093	0.8626±0.1112	0.8551±0.1101
newthyroid1	0.9856±0.0221	0.9812±0.0283	0.986±0.023	0.9859±0.0227	0.9856±0.0211	0.9884±0.0198
Seeds	0.9341±0.0779	0.9274±0.0798	0.9253±0.0742	0.9236±0.0779	0.9263±0.0778	0.9199±0.0792
Vertebral	0.888±0.0464	0.8697±0.0531	0.872±0.0589	0.8704±0.0548	0.8834±0.0467	0.8773±0.0545

Table 31: F_1 rank means for Logistic Regression classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	1.735	5.15	3.375	3.325	3.79	3.625
bupa	1.75	2.755	4.92	5.09	2.62	3.865
cleveland	2.07	6	3.235	3.155	3.35	3.19
ecoli	2.64	5.705	3.235	3.28	2.765	3.375
haberman	3.09	5.07	2.99	3.34	3.455	3.055
Sonar	2.82	3.99	3.375	3.875	3.22	3.72
glass0	2.165	4.24	3.81	4.105	2.785	3.895
kyphosis	2.995	4.695	3.365	3.335	3.24	3.37
newthyroid1	3.51	3.745	3.475	3.445	3.545	3.28
Seeds	3.245	3.46	3.49	3.565	3.46	3.78
Vertebral	2.65	3.965	3.815	4.04	3.085	3.445
Mean	2.606	4.434	3.553	3.687	3.21	3.509

The G-mean measurements of the Logistic Regression classifier are given in Table 32, and the average of the rankings are given in Table 33. In the 11 datasets, the best results were obtained two times without resampling, seven times in RUS, one time in US with DE, and one time in US with PSO. When the rank averages are examined, it was seen that the best ranking was achieved when RUS was applied and the worst ranking was reached when no resampling was applied. While the optimization method which gave the best result in G-mean rankings was ABC, the method that gave the worst result was PSO.

Table 32: G-mean values for Logistic Regression classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	0.5676±0.1264	0.6443±0.0986	0.6534±0.1153	0.6659±0.1156	0.6579±0.1107	0.6533±0.1217
bupa	0.6277±0.0944	0.6512±0.0817	0.6028±0.0728	0.5917±0.0789	0.6219±0.0877	0.6188±0.0776
cleveland	0.1495±0.2492	0.6424±0.1875	0.3951±0.3086	0.3728±0.3211	0.3499±0.313	0.3277±0.3099
ecoli	0.6304±0.2396	0.8087±0.1089	0.6991±0.2258	0.7334±0.1961	0.6913±0.2328	0.7038±0.2171
haberman	0.3366±0.1824	0.6211±0.1288	0.5238±0.1594	0.5384±0.1656	0.528±0.1766	0.5237±0.1607
Sonar	0.5984±0.2861	0.5068±0.242	0.5932±0.2585	0.5143±0.2604	0.5719±0.2703	0.5313±0.2745
glass0	0.6978±0.1216	0.7263±0.086	0.7201±0.1008	0.7238±0.0948	0.7111±0.0915	0.7196±0.1032
kyphosis	0.4359±0.3947	0.6308±0.3119	0.4907±0.3897	0.5615±0.3737	0.5065±0.3945	0.5442±0.3718
newthyroid1	0.9531±0.0893	0.9611±0.0886	0.9575±0.0909	0.9633±0.0836	0.9661±0.0616	0.9714±0.0575
Seeds	0.9503±0.0619	0.9477±0.061	0.9452±0.059	0.9436±0.0611	0.9461±0.0601	0.9413±0.0612
Vertebral	0.8248±0.0714	0.8393±0.0655	0.8318±0.0734	0.8336±0.0697	0.8287±0.0682	0.8317±0.0748

Table 33: G-mean rank means for Logistic Regression classifier

Dataset	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
breast	4.925	3.59	3.14	2.89	3.215	3.24
bupa	3.325	2.42	4.045	4.315	3.51	3.385
cleveland	4.6	1.94	3.39	3.42	3.805	3.845
ecoli	3.855	2.935	3.69	3.4	3.43	3.69
haberman	5.275	1.94	3.47	3.33	3.395	3.59
Sonar	2.895	3.995	3.315	3.835	3.29	3.67
glass0	3.8	3.255	3.4	3.435	3.615	3.495
kyphosis	3.595	3.445	3.705	3.375	3.45	3.43
newthyroid1	3.52	3.765	3.465	3.435	3.535	3.28
Seeds	3.245	3.46	3.49	3.565	3.46	3.78
Vertebral	3.73	3.36	3.525	3.435	3.535	3.415
Mean	3.888	3.1	3.512	3.494	3.476	3.529

4. Results and Discussion

The rank means for all cases is given in Table 34. Kruskal Wallis H test was used to test whether the difference between the rank means found for all cases was statistically significant. Since there was a significant difference between all cases, Dunn test was performed for pairwise comparisons. Significance levels of all these statistical tests are given in Table 35. The following inferences were made on the basis of these results:

- For AUC, F_1 and G-mean, there is a significant difference between the rank averages according to resampling status.
- For AUC, F_1 and G-mean, there is no significant difference between the no resampling and RUS rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the no resampling and GA rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the RUS and GA rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the no resampling and DE rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the RUS and DE rank averages.
- For AUC, there is no significant difference between the GA and DE rank averages. For F_1 , there is a significant difference between the GA and DE rank averages. For G-mean, there is no significant difference between the GA and DE rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the no resampling and ABC rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the RUS and ABC rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the GA and ABC rank averages.
- For AUC and F_1 there is a significant difference between the DE and ABC rank averages. There is no significant difference between the DE and ABC rank averages for G-mean.
- For AUC, F_1 and G-mean, there is a significant difference between the no resampling and PSO rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between RUS and PSO rank averages.
- For AUC and G-mean, there is a significant difference between GA and PSO rank averages. For F_1 , there is no significant difference between GA and PSO rank averages.
- For AUC, there is no significant difference between the DE and PSO rank averages. For F_1 and G-mean, there is a significant difference between the DE and PSO rank averages.
- For AUC, F_1 and G-mean, there is a significant difference between the ABC and PSO rank averages.

Table 34: Rank means for all cases

Criterion	No resampling	RUS	US with GA	US with DE	US with ABC	US with PSO
AUC	3.290	3.684	3.519	3.523	3.422	3.560
F_1	2.563	4.453	3.579	3.643	3.278	3.485
G-mean	3.747	3.194	3.524	3.505	3.467	3.563

Table 35: Kruskal-Wallis H test and pairwise comparisons for the mean of all cases

Test		p_{AUC}	p_{F_1}	p_{Gmean}	
Kruskall-Wallis H Test		<0.001*	<0.001*	<0.001*	
	1 - 2	<0.001*	<0.001*	<0.001*	
	1 - 3	<0.001*	<0.001*	<0.001*	
	2 - 3	<0.001*	<0.001*	<0.001*	
	1 - 4	<0.001*	<0.001*	<0.001*	
	2 - 4	<0.001*	<0.001*	<0.001*	
	3 - 4	0.439	0.012	0.252	
	Dunn Test	1 - 5	<0.001*	<0.001*	<0.001*
		2 - 5	<0.001*	<0.001*	<0.001*
		3 - 5	<0.001*	<0.001*	0.019
4 - 5		<0.001*	<0.001*	0.079	
1 - 6		<0.001*	<0.001*	<0.001*	
2 - 6		<0.001*	<0.001*	<0.001*	
3 - 6		0.070	0.001	0.100	
4 - 6		0.092	<0.001*	0.026	
5 - 6		<0.001*	<0.001*	<0.001*	

5. Conclusion

The problem with class imbalance is that the dependent variable contains different class observation numbers. This situation affects most classification methods. Resampling methods are the most common methods for solving the class imbalance problem. One of the resampling methods is undersampling method. The aim of the undersampling method is to achieve balance by extracting certain observations within the positive class. It is very important to correctly select these observations, as discarding observations causes information loss.

In this study, optimization methods were used in the selection of observations to be extracted in undersampling. The optimization methods used are GA, DE, ABC and PSO. According to the general rank means of these methods, the best optimization method was found to be ABC. The difference between the ABC method and other methods was found to be significant as a result of the Kruskal Wallis H and Dunn tests.

Four optimization methods were used in this study. Since these optimization methods were used in the default settings, they are open to investigating how effective they are at different settings. In addition, it is possible to use different classifiers and different performance criteria in the objective function.

References

- [1] Chen, L., Bao, L., Li, J., Cai, S., Cai, C., Chen, Z., *An aliasing artifacts reducing approach with random undersampling for spatiotemporally encoded single-shot MRI*, *Journal of Magnetic Resonance*, 237, 115-124, 2013.
- [2] Liu, B., Tsoumakas, G., *Dealing with class imbalance in classifier chains via random undersampling*, *Knowledge-Based Systems*, 192:105292, 2019.
- [3] Tomek, I., *Two modifications of CNN*, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6 (11), 769-772, 1976.
- [4] Elhassan, T., Aljourf, M., Al-Mohanna, F., Shoukri, M., *Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method*, *Global Journal of Technology and Optimization*, S1, 2017.
- [5] Pereira, R.M., Costa, Y.M., Silla Jr, C.N., *MLTL: A multi-label approach for the Tomek Link undersampling algorithm*, *Neurocomputing*, 383, 95-105, 2020.
- [6] Devi, D., Purkayastha, B., *Redundancy-driven modified Tomek-link based undersampling: A solution to class imbalance*, *Pattern Recognition Letters*, 93, 3-12, 2017.
- [7] Wilson, D. L., *Asymptotic properties of nearest neighbor rules using edited data*, *IEEE Transactions on Systems, Man, and Cybernetics*, 3, 408-421, 1972. <https://doi.org/10.1109/TSMC.1972.4309137>.
- [8] Laurikkala, J., *Improving identification of difficult small classes by balancing class distribution*, In *Conference on Artificial Intelligence in Medicine in Europe* (pp. 63-66), Springer, Berlin, Heidelberg, 2001. https://doi.org/10.1007/3-540-48229-6_9.
- [9] Bach, M., Werner, A., Palt, M., *The Proposal of Undersampling Method for Learning from Imbalanced Datasets*, *Procedia Computer Science*, 159, 125-134, 2019.
- [10] Lu, W., Li, Z., Chu, J., *Adaptive ensemble undersampling-boost: a novel learning framework for imbalanced data*, *Journal of Systems and Software*, 132, 272-282, 2017.
- [11] Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S., *Clustering-based undersampling in class-imbalanced data*, *Information Sciences*, 409, 17-26, 2017.
- [12] Ofek, N., Rokach, L., Stern, R., Shabtai, A., *Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem*, *Neurocomputing*, 243, 88-102, 2017.
- [13] Körzdörfer, G., Pfeuffer, J., Kluge, T., Gebhardt, M., Hensel, B., Meyer, C.H., Nittka, M., *Effect of spiral undersampling patterns on FISP MRF parameter maps*, *Magnetic Resonance Imaging*, 2019.
- [14] Holland, J.H., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press, 1992.
- [15] Eberhart, R., Kennedy, J., *Particle swarm optimization*, In *Proceedings of the IEEE International Conference on Neural Networks* 4, 1942-1948, 1995.
- [16] Storn, R., Price, K., *Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces*, *Journal of Global Optimization*, 11(4), 341-359, 1997.
- [17] Karaboga, D., *An idea based on honey bee swarm for numerical optimization*, Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 200, 1-10, 2005.
- [18] García, S., Herrera, F., *Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy*, *Evolutionary computation*, 17(3), 275-306, 2009.
- [19] Roshan, S.E., Asadi, S., *Improvement of Bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization*, *Engineering Applications of Artificial Intelligence*, 87, 103319, 2020.
- [20] Yu, H., Mu, C., Sun, C., Yang, W., Yang, X., Zuo, X., *Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data*, *Knowledge-Based Systems*, 76, 67-78, 2015.

- [21] Lavine, B.K., *Clustering and classification of analytical data*, Encyclopedia of Analytical Chemistry: Instrumentation and Applications, 2000.
- [22] He, H., Garcia, E.A., *Learning from imbalanced data*, IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284, 2009.
- [23] Das, B., Krishnan, N.C., Cook, D.J., *RACOG and wRACOG: Two probabilistic oversampling techniques*, IEEE Transactions on Knowledge and Data Engineering, 27(1), 222-234, 2014.
- [24] Fernàndes, E. R., de Carvalho, A.C., *Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning*, Information Sciences, 494, 141-154, 2019.
- [25] Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F., *Learning from imbalanced data sets*, Berlin: Springer, 1-377, 2018.
- [26] Chawla, N.V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W.P., *SMOTE: synthetic minority over-sampling technique*, Journal of Artificial Intelligence Research, 16, 321-357, 2002.
- [27] Kamalov, F., *Kernel density estimation based sampling for imbalanced class distribution*, Information Sciences, 2019.
- [28] Maldonado, S., Weber, R., Famili, F., *Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines*, Information Sciences, 286, 228-246, 2014.
- [29] Moayedikia, A., Ong, K.L., Boo, Y.L., Yeoh, W.G., Jensen, R., *Feature selection for high dimensional imbalanced class data using harmony search*, Engineering Applications of Artificial Intelligence, 57, 38-49, 2017.
- [30] Wong, M.L., Seng, K., Wong, P.K., *Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain*, Expert Systems with Applications, 141, 112918, 2020.
- [31] Dua, D., Graff, C., *UCI Machine Learning Repository* [<http://archive.ics.uci.edu/ml>], Irvine, CA: University of California, School of Information and Computer Science, 2019.