information

MDPI

Article

A Hybrid MultiLayer Perceptron Under-Sampling with Bagging Dealing with a Real-Life Imbalanced Rice Dataset

Moussa Diallo [1,2], Shengwu Xiong [1,*], Eshete Derb Emiru [1,3], Awet Fesseha [1,4], Aminu Onimisi Abdulsalami [1] and Mohamed Abd Elaziz [5]

1 School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China; moussdiall2015@gmail.com (M.D.); eshetede@whut.edu.cn (E.D.E.); awet.fesseha@mu.edu.et (A.F.); lecturer34@gmail.com (A.O.A.)
2 Department of Mathematics and Computer Science, Ecole Normale Superieure, 241 Bamako, Mali
3 School of computing, Debre Markos University, DebreMarkos 269, Ethiopia
4 School of Natural Science and computing, Mekele University, Mekelle 231, Ethiopia
5 School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China; abd_el_aziz_m@yahoo.com
* Correspondence: xiongsw@whut.edu.cn

Abstract: Classification algorithms have shown exceptional prediction results in the supervised learning area. These classification algorithms are not always efficient when it comes to real-life datasets due to class distributions. As a result, datasets for real-life applications are generally imbalanced. Several methods have been proposed to solve the problem of class imbalance. In this paper, we propose a hybrid method combining the preprocessing techniques and those of ensemble learning. The original training set is undersampled by evaluating the samples by stochastic measurement (SM) and then training these samples selected by Multilayer Perceptron to return a balanced training set. The MLPUS (Multilayer perceptron undersampling) balanced training set is aggregated using the bagging ensemble method. We applied our method to the real-life Niger_Rice dataset and forty-four other imbalanced datasets from the KEEL repository in this study. We also compared our method with six other existing methods in the literature, such as the MLP classifier on the original imbalance dataset, MLPUS, UnderBagging (combining random under-sampling and bagging), RUSBoost, SMOTEBagging (Synthetic Minority Oversampling Technique and bagging), SMOTEBoost. The results show that our method is competitive compared to other methods. The Niger_Rice real-life dataset results are 75.6, 0.73, 0.76, and 0.86, respectively, for accuracy, F-measure, G-mean, and ROC with our proposed method. In contrast, the MLP classifier on the original imbalance Niger_Rice dataset gives results 72.44, 0.82, 0.59, and 0.76 respectively for accuracy, F-measure, G-mean, and ROC.

Keywords: classification algorithms; imbalanced dataset; climate change; rice dataset; office du Niger

check for updates

Citation: Diallo, M.; Xiong, S.; Emiru, E.D.; Fesseha, A.; Abdulsalami, A.O.; Elaziz, M.A. A Hybrid MultiLayer Perceptron Under-Sampling with Bagging Dealing with a Real-Life Imbalanced Rice Dataset. *Information* **2021**, *12*, 291. https://doi.org/10.3390/info12080291

Academic Editor: Arkaitz Zubiaga

Received: 2 June 2021
Accepted: 20 July 2021
Published: 22 July 2021

1. Introduction

Demographic growth in West Africa in general and Mali, in particular, requires abundant agricultural production to cope with this demographic growth. However, agricultural production in this region is traditional, i.e., linked to the weather. Climate change has a considerable impact on agricultural production in this region due to high temperatures [1]. Exploring machine learning technologies to predict agricultural production is an exciting challenge in this climatically unstable region [2]. Since machine learning gives significant results in prediction in certain areas, including recommendation systems, social media, finance, image processing, spam, anti-spam filtering, text classification, speech recognition, medicine, and environment [3], we explore those technologies. Crop production Predicting by machine learning prediction methods using features such as climate data can be a significant challenge. Rice is the most produced and consumed cereal in Mali. In this paper, we study the prediction of rice production using climate data in Mali in the irrigated area called the Niger office. We use the prediction methods of the classification algorithms for rice production from the Niger office. Classifications methods are more often known for solving qualitative problems, while rice production is quantitative. However, this adaptation of the solution is since the Niger office Company uses a threshold to qualify whether rice production is good or bad. This threshold is 6.2 tones per hectare. Thus, if the production is below this threshold, then it is qualified as bad, and if it is greater than or equal to this threshold, then the production is qualified as good. Classification algorithms are used for supervised problems. Traditional classification algorithms are efficient when the training dataset has certain representativeness and balance between the labels [4]. However, these algorithms are not efficient in the case of an imbalanced dataset [4]. After constructing our real-life dataset Niger_Rice dataset of rice production qualification using climatic data, it appears that this dataset is imbalanced according to [5]. The paper [5] defines an imbalanced dataset as being a dataset in which some observations are little compared to others. We use the Niger_Rice real-life dataset in this study and forty-four other imbalance datasets.

Other methods have been used to overcome the limitations of traditional classification algorithms on imbalanced datasets. The paper [6] groups them into four categories: data-level, algorithm-level, cost-sensitive, and ensemble learning methods. The data-level method is a preprocessing method that uses techniques to balance (under-sampling, over-sampling, or hybrid) the training set before using traditional classification algorithms. The algorithm-level method is a technique of modifying a particular algorithm to adapt it to the imbalanced dataset. As for the cost-sensitive methods, they are the link between the data-level methods and the algorithm-level methods. The cost of misclassification is modified according to the learning algorithm for cost-sensitive. The ensemble learning methods are a combination of traditional classification algorithms.

This study uses a hybrid method combining a preprocessing technique (sub-sampling) and ensemble learning. The undersampling technique is MLPUS (Multilayer perceptron Under-sampling), which comprises three key steps: clustering, SM (stochastic measurement) evaluation, and training MLP on the evaluated samples. Clustering is the grouping of samples from the majority class to select the essential samples. The stochastic measurement evaluation is used for sample selection, and the last step is training the MLP on the samples selected by SM. The ensemble learning method used is bagging, which takes as training data the datasets balanced by MLPUS. To evaluate the performance of our method, we use forty-four other datasets and six other methods in this paper.

The contributions made in this paper are as follows:

(1) The collection of climatic and rice production data from 1990 to 2020 for the Niger officearea and their fusion to make the Niger_Rice dataset.
(2) The MLPUS and Bagging methods are combined to make a hybrid method of solving imbalanced dataset problems.
(3) We combine MLPUS and Boosting methods to make a hybrid method of solving imbalanced dataset problems.

The remainder body of this paper is organized as follows: Section 2 gives the related works. The method used is detailed in Section 3. Section 4 provides the elements of our experiment, and finally, we conclude with Section 5.

2. Related Works

The details of two categories (data-level and ensemble learning) are presented in this section because the outcome of this paper is based on data-level and ensemble learning. However, the cost-sensitive algorithms are detailed in [7] with AdaCost and in [8] with AdaC1, AdaC2, and AdaC3. As for the algorithm level category, its details are reported in [6]. The algorithm level category is detailed in [9,10].

2.1. Sampling Methods

Sampling or data-level methods are used to have a certain balance of classes in the training set. These methods can be under-sampling, oversampling, or hybrid methods (combining the previous two ways).

The under-sampling methods remove instances of the majority class by following a particular technique. For example, the removing observations of the majority class randomly [11], the under-sampling based on ENN [12], and that based on the Tomek Link distance method [13].

Re-sampling methods based on over-sampling the minority class are the most common in the data-level category. In this category, we have the over-sampling proposed by [14], the over-sampling based on the cluster [15]. The SMOTE (Synthetic Minority Oversampling Technique) is the most widely used literature [10]. SMOTE creates the synthetic instances in the minority class based on the nearest neighbors of that class's given sample [16]. Several techniques have been proposed to improve SMOTE, such as Safe-level SMOTE [17], or Borderline-SMOTE [18], or even ADASYN [19]. The paper [20] reports 85 variants of SMOTE.

Hybrid re-sampling methods combine over-sampling and under-sampling techniques [21]. In Paper [22], a mixed re-sampling method has been proposed using SMOTE and an under-Sampling algorithm to solve the noise problem. Another hybrid re-sampling approach combines spatiotemporal over-sampling and selective un-der-sampling to align foreground and background pixels in a video [23]. Two separate and parallel particle swarm optimization processes used a mixed re-sampling method [24].

2.2. Ensemble Methods

According to the paper [25], sampling methods and ensemble methods have effectively resolved class imbalance in recent years. The ensemble methods, often also called ensemble solutions, combine several basic classifiers to integrate their results and generate a single classifier to improve performance. Ensemble solutions generally give better performance compared to the individual classifier [26]. In the literature, they are Bagging and boosting come up most often. This category can also merge the previous categories to be effective in the problem of class imbalance. With a random sampling of the training data, bagging [27] obtains a basic classifier. The ensemble methods are often not adapted to the problem of class imbalance [28]. Combining these methods with the other methods (data-level, algorithm-level, and cost-sensitive) is used to adapt them to the specific problem of class imbalance. It is in this context that bagging with under-sampling methods has been proposed in [11]. However, often ensemble methods are only implemented to solve the class imbalance problem such as: SMOTEBagging [29], SMOTEBoost [30], RAMOBoost [31], RUSBoost [32], EUSBoost [33], EasyEnsemble [34], Random balance-boost [35].

The SMOTEBagging [29] is a mix of the Bagging and SMOTE techniques. SMOTE generates artificial instances of the positive class to build a dataset with balanced categories. SMOTEBoost [30] is a combination of SMOTE and AdaBoostM2. Every turn after boosting, SMOTE is used to generate new synthetic cases from the minority class. These synthetic data have the same weights as the original data, but the original data's weights have been changed. However, the information that has significant weights are those that are difficult for the previous classifiers. The RAMOBoost [31] is the combination of ADASYN [19] and AdaBoostM2. The only difference between RAMOBoost [31] and SMOTEBoost [30] is the algorithm used to create synthetic instances.

In contrast, SMOTEBoost [30] uses SMOTE for these instances, ROMOBoost uses ADASYN [19]. These artificial data are created based on underlying data distribution. RUSBoost is the combination of random under-Sampling and Ada-BoostM2. After each round of boost, this time, it is the random under-Sampling that is applied. EUSBoost [33] is the union of AdaBoostM2 and an evolutionary algorithm. In this technique, the under-Sampling uses an evolutionary algorithm to remove instances of the majority class. EUSBoost uses different subsets of majority class instances to promote diversity on each iteration to train each classifier. EasyEnsemble [34] subdivides the majority class into several subsets, then introduces an AdaBoost set by taking each subsets' and mixing the classifiers' outputs. Random balance-boost is a combination of SMOTEBoost [30] and RUSBoost [32]. In other words, after each boost, a hybrid re-sampling is performed. The SMOTE technique does the oversampling for the minority class, and the under-sampling is done randomly for the majority class. The paper [36] categorizes these approaches into four principal families UnderBagging [11], OverBagging [29], (hybrid) UnderOverBagging [29] and IIVotes [37].

3. Research Materials Proposed Hybrid MLPUS with Bagging Methods

As we have already mentioned in the introduction section, our proposed hybrid method uses re-sampling and ensemble learning methods. We use the hybrid combination of MLP under-sampling and bagging. The MLPUS is an under-sampling method [38] that brings together three key concepts: clustering (grouping) majority class samples, using stochastic measurement (SM) evaluation to select large samples, and training the MLP the examples set from SM evaluation. Algorithm 1 shows the pseudo-code of the existing method MLPUS [38]. The clustering method used by MLPUS is K-means, the number of clusters n is determined by $\sqrt{N_p}$ for each class. N_p denotes the number of samples of the minority class. The selection closest to the cluster's center of gravity is estimated and then added to the training set. Since the number of collected samples is equal, in the initial training of MLP, we have an equal number of samples for each class. The value p is a constant for each iteration. The majority of class samples are grouped into N_p clusters so that only the most significant representatives participate in the sub-sampling and the distribution of the data is then preserved to get the same number N_p of samples for each class and perform SM on each class. The samples close to the centers are chosen among the N_p clusters, and their SM is calculated. After this calculation, only the n examples are selected, those with a high SM. The same procedure is calculated for the samples of the minority class. The MLP obtains a balanced dataset of these $2n$ samples for training. We get $2in$ of samples where i is the number of iterations; on the other hand, i cannot be greater than n. The samples will be iteratively removed from the original imbalanced dataset until the minority samples are more significant than n.

The MLP (Multi-Layer Perceptron) has a standard neural network architecture using backpropagation to train its model. It has at least three layers: the output layer, the input layer, and one or more intermediate (hidden) layers. In this architecture, the initial weights of the connections are random, and a learning rate is chosen as a function of a constraint affecting the MLP. If this rate is lower, MLP learning is slow, and if it is high, MLP learning will not go well. We define the inputs as $x_1, x_2, \ldots, x_n$ and the corresponding weights $w_1, w_2, \ldots, w_n$, then the outputs of each neuron are calculated as follows $x_1w_1 + x_2w_2 + \ldots + x_nw_n$. For each layer unit, the outcome is propagated, and its error is calculated as error = predicted_output − actual_output. The function which defines the MLP is as follows:

$$f(x) = \sum_{i=1}^{m} w_{ki} f(\sum_{j=1}^{n} w_{ij} x_j) \tag{1}$$

where w_{ki} is the connection of input neuron k and the hidden layer neuron i, w_{ij} is the connection of hidden layer neuron i and output neuron j, m is the number of hidden neurons [38], and $f(x)$ is the activation function. Activation functions are functions used to calculate the weighted sum of inputs and biases in neural networks. AFs are used to decide whether a neuron can be triggered or not. AFs can be the main functions or their variants. Here are some main activation functions:

- The sigmoid AF, also called logistic function [39], is defined as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

- The AF tanh is the smoother hyperbolic tangent function centered on zero with a range between -1 and 1 [40] and given by:

$$f(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right) \tag{3}$$

- The rectified linear unit (ReLU) AF [41] determines the threshold operation on each input element and sets negative values to zero. The formula of ReLU is defined by:

$$f(x) = max(0, x) \tag{4}$$

- The Swish AF [42] is defined by

$$f(x) = \frac{x}{1 + e^{-x}} \tag{5}$$

- The exponential linear unit (ELU) AF [43] is given by

$$f(x) = \begin{pmatrix} x, & if\ x > 0 \\ \alpha e^x - 1, & if\ x \le 0 \end{pmatrix} \tag{6}$$

- The Exponential linear Squashing (ELiSH) AF [44] is given by:

$$f(x) = \begin{cases} \left(\frac{x}{1+e^{-x}} \right), & x \ge 0 \\ \left(\frac{e^x - 1}{1+e^{-x}} \right), & x < 0 \end{cases} \tag{7}$$

In the paper [38], the calculation of SM for the MLP is the main criterion for under-sampling. The SM is the square of the difference between the output of the future sample and the original dataset. Thus, the greatest value is assigned to the hard-to-learn samples and added iteratively to the training set. The MLP will not misclassify these samples. The following formula gives the calculation of x samples by the SM:

$$I(x) = \frac{1}{H} \sum_{h=1}^{H} (g(x + \Delta x_h) - g(x))^2 \tag{8}$$

where is the Halton point and is defined as:

$$g(x) = \sum_{i=1}^{m} w_{ki} f(\sum_{j=1}^{n} w_{ij} x_j) \tag{9}$$

where m is the number of hidden neurons, et $f(x)$ is the sigmoid function.

We use the bootstrap aggregation method, also called bagging. The bootstrap aggregation method was introduced by Breiman [27] to construct bagging sets for the first time. This method relies on the idea of training different classifiers with random replicates (the size of the original training dataset is kept) of the original training dataset. Different subsets of data are used to achieve diversity with re-sampling. The deduction of a class for an unknown opinion of each singular classifier is obtained by majority or weighted vote. Its simplicity and its good generalizability have enabled bagging methods to deal with data imbalance problems with many approaches. Algorithm 2 shows how the pseudo-code of our method will process. The framework of our proposed method is shown in Figure 1.

Algorithm 1 MultiLayer Perceptron UnderSampling (MLPUS)

```
input Imbalanced Training Set $D$
output Balanced Training set $D$
1.  Step 1: The initial MLP training
2.  $D_{maj}$ is majority class sample
3.  $D_{min}$ is minority class sample
4.  $n = \sqrt{N}_p\ (N_p)$ is the number of clusters for both $D_{min}$ and $D_{maj}$
5.  $G_0$ centroid of $D_{min}$
6.  $H_0$ centroid of $D_{maj}$
7.  $D_{min} = D_{min} - G_0$
8.  $D_{maj} = D_{maj} - H_0$
9.  Step 2 Train MLP using $D$
10. While $N_p > n$ do
11. Step 3: most essential samples from $D_{maj}$
12. $N_p$ become number of the cluster for $D_{maj}$
13. Initialization of C
14. $G_d$ new centroid for $D_{min}$
15. $H_d$ new centroid for $D_{maj}$
16. d = d + 1
17. for i = 1 to $N_p$ do
18. C = $D_{maj}$ + C
19. End for
20. Step 4 compute the value of the Stochastic measure for each sample of C and $D_{min}$ as
21. $f(x) = \frac{1}{H}\sum_{h=1}^{H}(g(x+\Delta x_h) - g(x))^2$
22. Step 5 add sample from C get largest SM to set $G_d$ and $H_d$ respectively
23. Step 6:
24. $D_{min} = D_{min} - G_d$
25. $D_{maj} = D_{maj} - H_d$
26. $D = D \cup G_d \cup H_d$
27. Step 7 Train MLP using $D$
```

Algorithm 2 Proposed method

```
input imbalanced dataset set S = $\{x_i y_i\} = 1, \ldots,$ N; and $y_i\ y_i \in [-1, 1]$; n: Bootstrap size, T: number of iterations, I: Weak Learner
1:  for t = 1 to T do
2:  $S_t \leftarrow$ MLPUS (n, S)
3:  $h_t \leftarrow I(S_t)$
4:  end for
Output Bagged classifier: $H(x) = \sin\left(\sum_{t=1}^{T} h_t(x)\right)$
```

Since in the paper [36], bagging associated with re-sampling methods has provided approaches that can address imbalanced data. Our proposed method consists of combining a re-sampling method (under-sampling) with bagging to deal with data imbalance. Our proposed method is from the UnderBagging family. It consists of using MLP to under-sample the initially imbalanced training dataset to balance and then use bagging. Our approach consists of three main steps, as shown in Figure 1. The first step is to initialize and then train the MLP on the imbalanced training set to find the most representative samples. The second step is to assess the SM of the most representative samples to provide a training set balance. The third and last step is to bootstrap the dataset provided after the training set.

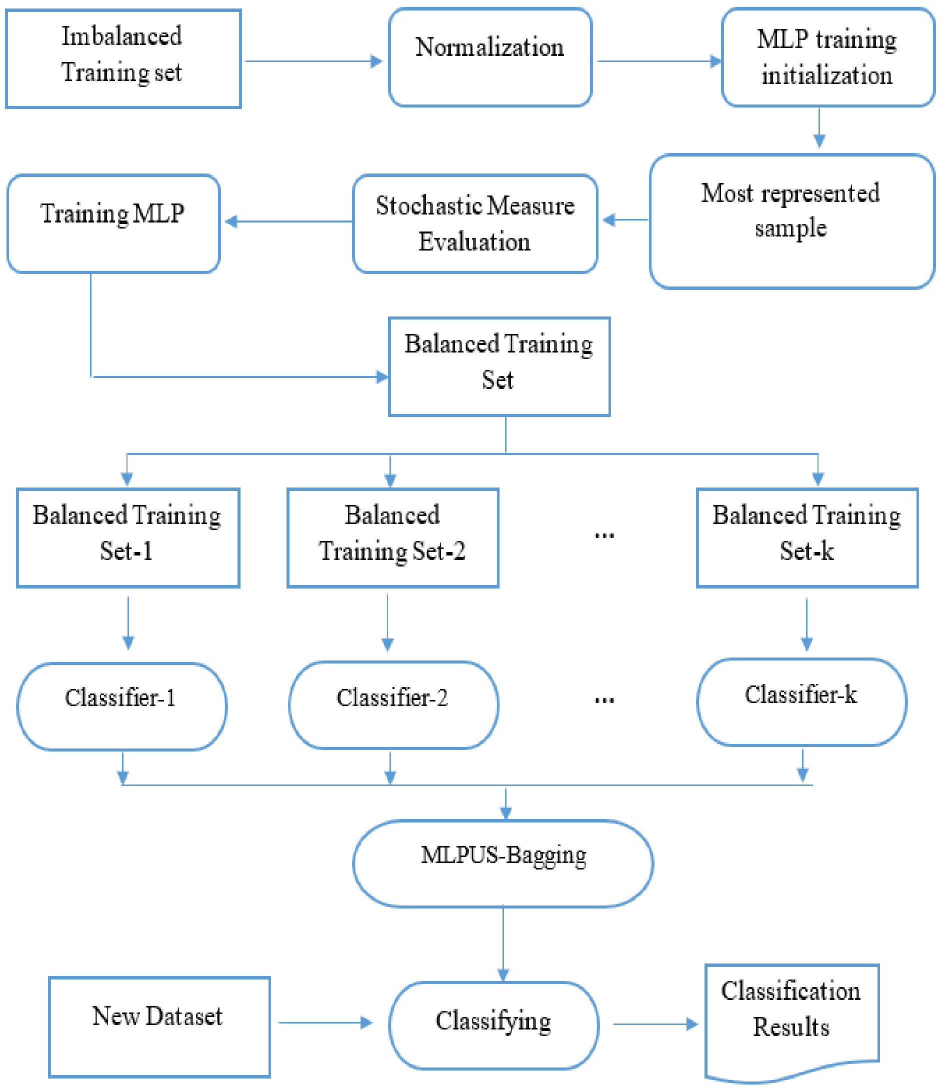

Figure 1. Proposed method Framework.

The balanced training set was obtained after applying MLPUS postulates for bootstrapping. As the EL method is bagging, it results in the different classifiers in sequential order, as shown in Figure 1. Each primary bagging classifier replaces the original training set with bootstrapping. That ensures that all the base classifiers are not affected by the imbalance. In the end, each base classifier provides a Bagged classifier: $H(x)$.

4. Experiments

4.1. Datasets

This paper uses forty-five imbalanced datasets, of which one is a real-life Niger_Rice dataset, and the other forty-four datasets come from the KEEL dataset repository [45]. The real-life dataset (Niger_Rice) is a rice production dataset available in the URL (https://github.com/moussdiall/Imbalanced_dataset_Niger_Rice, accessed on 21 July 2021). This dataset has as attributes the total precipitation, the average of maximum, average and minimum temperature of six months (from June to November) of a regular season, according to the Niger Office in Mali. The Niger Office is a Malian parastatal company that manages one of the largest and oldest irrigated areas in West Africa. Table 1 describes the features of the Niger_Rice dataset.

Table 1. Niger_Rice dataset headers description.

Header	Description
P	Determines the total amount of precipitation recorded from June to November
Max	Represents the average of the maximum temperature recorded from June to November
Min	Returns the value of the average of the minimum temperature recorded from June to November
Average	Returns the value of the average temperature recorded from June to November
Yield	yes or no class (which qualifies the result as good or bad depending on the threshold)

The datasets coming from the KEEL repository are not wholly independent. Several of them are only variants of the original datasets. For example, we have twenty variants of the yeast dataset, fourteen variants of the glass dataset, eight variants of the ecoli dataset, four variants of the vehicle dataset, two variants of the new-thyroid, shuttle, Abalone, page-blocks dataset, and the other datasets are present with only one variant. In Table 2, we give the details of each dataset. The header Att designates the number of attributes of the dataset. The header NI indicates the number of instances contained in the dataset. P and N represent respectively the number of positive and negative class samples (minority and majority). IR. (Imbalanced Ratio) designates the quotient between the majority class (negative class) and the minority class (positive class). The IR of the Niger_Rice dataset is 3.43, while the IR of other datasets ranges from 1.82 to 129.44.

Table 2. Datasets Description.

No	Dataset's Name	Att	NI	P	N	IR
1	glass1	9	214	76	138	1.82
2	ecoli-0_vs_1	7	220	77	143	1.86
3	wisconsin	9	683	239	444	1.86
4	pima	8	768	268	500	1.87
5	iris0	4	150	50	100	2
6	glass0	9	214	70	144	2.06
7	yeast1	8	1484	429	1055	2.46
8	haberman	3	306	81	225	2.78
9	vehicle2	18	846	218	628	2.88
10	vehicle1	18	846	217	629	2.9
11	vehicle3	18	846	212	634	2.99
12	glass-0-1-2-3_vs_4-5-6	9	214	51	163	3.2
13	vehicle0	18	846	199	647	3.25
14	ecoli1	7	336	77	259	3.36
15	new-thyroid1	5	215	35	180	5.14
16	new-thyroid2	5	215	35	180	5.14
17	ecoli2	7	336	52	284	5.46
18	segment0	19	2308	329	1979	6.02
19	glass6	9	214	29	185	6.38
20	yeast3	8	1484	163	1321	8.1
21	ecoli3	7	336	35	301	8.6
22	page-blocks0	10	5472	559	4913	8.79
23	yeast-2_vs_4	8	514	51	463	9.08
24	yeast-0-5-6-7-9_vs_4	8	528	51	477	9.35
25	vowel0	13	988	90	898	9.98
26	glass-0-1-6_vs_2	9	192	17	175	10.29
27	glass2	9	214	17	197	11.59
28	shuttle-c0-vs-c4	9	1829	123	1706	13.87
29	yeast-1_vs_7	7	459	30	429	14.3
30	glass4	9	214	13	201	15.47

Table 2. *Cont.*

No	Dataset's Name	Att	NI	P	N	IR
31	ecoli4	7	336	20	316	15.8
32	page-blocks-1-3_vs_4	10	472	28	444	15.86
33	abalone9-18	8	731	42	689	16.4
34	glass-0-1-6_vs_5	9	184	9	175	19.44
35	shuttle-c2-vs-c4	9	129	6	123	20.5
36	yeast-1-4-5-8_vs_7	8	693	30	663	22.1
37	glass5	9	214	9	205	22.78
38	yeast-2_vs_8	8	482	20	462	23.1
39	yeast4	8	1484	51	1433	28.1
40	yeast-1-2-8-9_vs_7	8	947	9	938	30.57
41	yeast5	8	1484	44	1440	32.73
42	ecoli-0-1-3-7_vs_2-6	7	281	7	274	39.14
43	yeast6	8	1484	35	1449	41.4
44	abalone19	8	4174	32	4142	129.44
45	Niger_Rice	4	62	14	48	3.43

The following subsections provide details of the design of the Niger_Rice dataset.

4.1.1. Niger_Rice Dataset Study Area

The research area is in Mali, in the Niger River's inner delta. Mali, like the other Sahel nations, has a diverse environment, according to [2]. With over 100,000 ha, the Niger Office comprises seven irrigated areas: Kolongo, Niono, N'Debougou, M'Bewani, Macina, Molodo, and Kouroumari.

4.1.2. Niger_Rice Dataset Data Collection

Climate data and agricultural yields from the records of Mali's National Meteorological Agency (MALI METEO) and the Niger Office company make up the Niger_Rice dataset. Rice data for the two Areas (named Casier and Hors-Casier) were collected from the Niger Office's Planning and Statistics Department from 1990 to 2020. In addition, the Mali Météo provided climate data that could affect agricultural production over the same period from 1990 to 2020. Precipitation, minimum, average, and maximum temperatures are among the climate records.

- Precipitation: the cumulative average monthly rainfall (measured in millimeters) in the Niger Office region during the agricultural season (June to November).
- Minimum temperature: the average minimum temperature (in degrees Celsius) in the Niger Office region for the monthly agricultural season (June to November).
- Maximum temperature: the average monthly maximum temperature (in degrees Celsius) in the Niger Office region during the agricultural season (June to November).
- Average temperature: the average monthly average temperature (in degrees Celsius) of the Niger Office region during the agricultural season (June to November).

The quantity of agricultural output per cultivation area is used to measure agricultural yield. Agricultural production is measured in tones, and cultivation area is measured in hectares. The ratio of agricultural output to the cultivated area (tones/hectare) is known as yield.

4.1.3. Niger_Rice Dataset Preprocessing

To make this data usable by machine learning technologies, we preprocess the data with the following steps:

Step 1: Collect monthly climate data (precipitation, maximum, minimum, and average temperatures) for the regular agricultural season (June to November) recorded at the Niger office area with the Mali Météo from 1990 to 2020.

Step 2: Calculate the total precipitation, the average maximum, minimum, and average temperature for the season (June to November) in the two zones (Locker and non-locker) of the Niger Office from 1990 to 2020

Step 3: Collect data on the area under cultivation, production, and agricultural yield from 1990 to 2020, with the planning and statistics service of the Niger Office.

Step 4: We have gathered this raw data on a Microsoft Excel sheet composed of the headings of the following columns: No, year, name of the zone, precipitation, maximum temperature, average temperature, minimum temperature, crop area, production of the site, and the yield.

Step 5: For proper data preparation to apply data mining technologies, non-demanding columns have been removed. These columns are: No, year, name of the zone.

Step 6: As the yield is calculated according to the crop area and the production, these two columns have been deleted. The yield determines the output quality (good or bad); we define the yield as the class label.

Step 7: The dataset is sorted by performance to rank "good" or "bad" records. The bad yield is less than 6.2 tons/hectare, and the good yield is over 6.2 tons/hectare. The bad class has 48 records, while the good class has 14 records.

Step 8: The final File of this dataset is saved in CSV format to apply machine learning techniques. The final dataset file has five columns: Rainfall, maximum temperature, average temperature, minimum temperature, and crop yield.

4.2. Evaluation Metrics and Experimental Setting

4.2.1. Baseline

The principal exisiting solutions used in this paper as benchmark methods are six including MLP classifier [46], MLPUS [38], Under_Bagging [11], SMOTE_Bagging [29], RUS_Boost [32], and SMOTE_Boost [30].

4.2.2. Performance Evaluation

A popular performance concept for classification is the Confusion Matrix, a table that shows the model's predictions against actual labels (see Table 3). The rows of this confusion matrix define the instances of a current class and the columns the instances of the predicted label.

Table 3. Confusion Matrix.

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

With TP (True Positives), when the prediction and the actual value are positive. TN (True Negative) when the prediction and the actual value are negative. FP (False Positive): when the real value is negative while the prediction is positive The FN (False Negative) when the real value is positive while the prediction is negative.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{recall} = \frac{\text{FP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{12}$$

$$\text{F} - \text{score} = \frac{2{*}\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{13}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{14}$$

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} \tag{15}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{16}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{17}$$

$$\mathrm{G} - \mathrm{mean} = \sqrt{\mathrm{TPR} \times \mathrm{TNR}} \tag{18}$$

TPR, FNR, FPR, and TNR are True Positive Rate, False Negative Rate, False Positive Rate, and True Negative Rate. The ROC (Receiver Operating Characteristic) curve plots the evolution of TPR as a function of FPR, varying a threshold on the confidence (probability).

4.2.3. Experimental Setting

The parameters set are n, k, and m. The n is the number of clusters for each class at the level of the MLPUS method. The k is the number of subdivisions of the training set applying for bagging after balancing the training set, and m is the number of neurons hidden for feedback-propagation to train its MLP model. These parameters are defined by the user and must be known for the execution of the experiment. To determine the significance of the results, we use an alpha = 0.05, which is the confidence interval between the classifier's results. In implementing these experiments, we used a laptop computer with an Intel Core i7-4720HQ (2.59) microprocessor, with 8.00 GB RAM and a 64-bit file system for the operating system. We use MATLAB for preprocessing methods and Weka machine learning tools to create the models. The two-fold-five-iterations split cross-validation is used to train the model.

4.3. Experimental Results and Discussion

The different experimental results of our method and six other methods of the 45 imbalanced datasets of Table 2 are summarized in this section. The results in this experiment are obtained by two-fold cross-validation with five iterations, with the standard deviation (std) between the parentheses. The other six methods mentioned earlier as baseline are MLP classifier [46] on the original imbalanced dataset, MLPUS (the MLP Under-Sampling preprocessing method) [38], SMOTEBagging [29], SMOTEBoosting [30], Under-Bagging [11], and RUS_Boost [32]. Beyond our proposed method, MLPUS with Bagging, we also performed MLPUS with boosting in the experiment. In total, in this experiment, we compare eight methods for a better analysis. Table 4 shows the accuracy (std) of the different techniques with the 23 datasets in Table 2. The accuracy is not a powerful metric in an imbalanced dataset. We have considered three other metrics: F-Measure, G-Mean, and the ROC curve (Receiver Operating Characteristic). Table 5 shows the F-Measure (std) metric of the different methods with the 45 datasets of Table 2. The G-Mean (std) results of each technique on each of the datasets of Table 2 are shown in Table 6. Finally, Table 7 shows the different outcomes of each method's ROC (std) curve with the 45 datasets in Table 2. The results are significantly better with the letter "v", or significantly weak with the symbol "*", or not significant with a confidence interval of alpha 0.05, i.e., 5% risk of error.

Table 4. Accuracy results of all methods for all datasets.

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
ecoli-0_vs_1	97.91 (2.48)	99.09 (1.24)	98.04 (1.79)	97.92 (1.63)	97.78 (1.67)	97.28 (3.32)	97.14 (2.69)	97.91 (2.48)
ecoli1	92.73 (4.60)	90.17 (2.52)	92.22 (3.66)	90.40 (2.83)	90.16 (2.66)	93.37 (4.62)	91.04 (5.06)	91.82 (6.66)
ecoli2	97.87 (3.45)	94.34 (2.88)	90.33 (4.95) *	94.54 (2.36)	94.33 (2.49)	90.98 (5.62) *	90.57 (6.44) *	96.74 (6.26)
ecoli3	93.43 (6.81)	93.16 (4.15)	90.00 (3.91)	92.50 (2.74)	93.85 (2.47)	92.00 (5.18)	90.00 (5.83)	92.57 (6.00)
glass-0-1-2-3_vs_4-5-6	94.73 (6.07)	90.18 (2.01) *	95.05 (6.12)	94.87 (3.11)	94.64 (3.16)	94.32 (3.33)	95.66 (4.16)	94.52 (5.12)
glass0	92.00 (5.58)	80.86 (4.38) *	71.43 (7.58) *	86.75 (5.16) *	88.59 (4.23) *	77.14 (6.84) *	76.71 (8.88) *	92.43 (3.90)
glass1	77.36 (7.95)	68.22 (5.40) *	65.05 (8.34) *	82.76 (5.20)v	86.21 (4.53)v	75.26 (7.42)	78.82 (5.30)	74.37 (7.45)
glass6	88.67 (9.18)	97.67 (4.03)v	91.67 (10.21)	95.72 (2.22)v	95.79 (2.11)v	89.33 (8.74)	88.70 (8.06)	90.39 (6.99)
haberman	56.87 (7.47)	74.17 (4.34)v	62.92 (8.03)v	71.42 (4.55)v	69.46 (4.59)v	62.23 (7.65)v	64.70 (7.58)v	59.87 (5.89)
iris0	98.60 (3.07)	100.00 (0.00)	100.00 (0.00)	99.30 (1.35)	99.50 (1.02)	98.60 (2.29)	99.00 (2.04)	98.60 (3.07)
new-thyroid1	94.86 (4.84)	98.14 (1.95)	95.71 (3.91)	97.68 (1.97)	97.52 (2.18)	91.71 (6.74)	93.43 (7.96)	95.14 (5.35)
newthyroid2	94.86 (4.84)	98.14 (1.04)	100.00 (0.00)v	98.08 (1.58)	97.76 (2.03)	92.57 (6.99)	94.86 (6.02)	95.14 (5.35)
page-blocks0	98.00 (0.85)	96.69 (0.52)	93.56 (2.64) *	97.11 (0.38)	97.17 (0.42)	95.31 (1.29)	94.85 (1.23)	98.60 (0.96)
pima	77.46 (3.70)	74.09 (2.75)	76.49 (3.24)	78.86 (2.04)	77.70 (2.23)	73.76 (4.36)	71.60 (4.85) *	73.54 (3.72)
segment0	97.66 (1.68)	99.70 (0.33)	99.08 (1.25)	99.48 (0.40)	99.75 (0.28)	98.24 (0.93)	98.97 (0.90)	99.15 (1.01)
vehicle0	94.57 (2.95)	96.93 (0.50)	95.73 (2.11)	96.40 (1.24)	97.45 (0.89)	92.86 (3.07)	94.57 (2.66)	95.48 (2.96)
vehicle1	82.35 (3.13)	83.21 (2.39)	77.41 (3.83) *	81.62 (2.90)	82.94 (2.27)	74.83 (5.18) *	72.39 (4.48) *	82.49 (3.15)
vehicle2	75.37 (2.84)	97.87 (0.89)v	96.33 (1.88)v	97.22 (1.07)v	98.52 (0.83)v	95.14 (2.80)v	96.92 (2.07)v	75.56 (4.71)
vehicle3	80.43 (4.22)	82.51 (2.20)	78.99 (8.92)	82.16 (2.76)	82.63 (2.42)	74.48 (3.70) *	73.54 (2.65) *	82.32 (4.23)
wisconsin	99.29 (0.84)	95.90 (0.85)	95.39 (2.84) *	97.16 (0.95)	97.61 (0.95)	96.95 (1.38)	95.86 (1.61)	99.41 (0.96)
yeast1	83.91 (1.89)	77.63 (2.33) *	69.12 (4.39) *	78.42 (1.59) *	76.84 (1.98) *	71.66 (2.61) *	69.72 (3.20) *	81.42 (2.49)
yeast3	92.21 (3.57)	94.54 (1.24)	89.58 (2.69)	95.06 (1.05)	94.52 (1.19)	92.27 (2.23)	90.49 (2.75)	91.54 (4.47)
abalone19	99.23 (0.07)	73.13 (11.96) *	86.37 (0.89) *	98.48 (0.05)	98.48 (0.05)	68.85 (7.25) *	62.69 (9.38) *	82.56 (11.18) *
abalone9-18	95.08 (1.01)	81.87 (10.28) *	87.50 (2.24) *	91.72 (1.45)	90.56 (1.67) *	72.72 (15.74) *	65.66 (14.93) *	84.50 (9.15) *
ecoli-0-1-3-7_vs_2-6	98.93 (0.97)	84.67 (22.53) *	93.58 (1.85) *	97.57 (0.93)	98.61 (0.77)	73.33 (14.91) *	80.00 (29.81) *	82.00 (22.53) *
ecoli4	98.51 (1.49)	90.50 (9.74) *	95.92 (1.48)	97.19 (2.00)	97.19 (2.23)	85.00 (16.30) *	90.00 (10.46) *	91.00 (8.48) *
glass-0-1-6_vs_2	89.07 (2.14)	73.90 (19.18) *	84.97 (3.20)	88.52 (3.07)	83.74 (0.91)	59.52 (20.48) *	82.38 (11.86) *	81.05 (16.34) *
glass-0-1-6_vs_5	96.73 (1.28)	84.67 (22.40) *	97.37 (1.68)	96.36 (2.99)	98.46 (2.29)	93.33 (14.91)	93.33 (14.91)	85.67 (17.27) *
glass2	89.27 (2.59)	75.33 (21.60) *	88.02 (2.60)	87.88 (2.46)	85.28 (0.95)	61.90 (12.14) *	67.62 (11.86) *	85.90 (14.61)
glass4	96.72 (2.67)	93.20 (9.30)	89.25 (3.51) *	96.90 (5.80)	94.26 (3.72)	80.67 (14.22) *	88.67 (10.43) *	92.40 (9.55)
glass5	97.19 (3.05)	87.00 (20.14) *	97.66 (1.53)	98.18 (2.49)	98.21 (1.86)	100.00 (0.00)	100.00 (0.00)	92.00 (13.28)
page-blocks-1-3_vs_4	99.79 (0.47)	94.33 (9.19) *	98.31 (0.96)	99.60 (0.89)	99.60 (0.89)	87.73 (9.59) *	94.70 (4.85)	98.27 (5.46)
shuttle-c0-vs-c4	99.95 (0.12)	99.02 (1.67)	99.98 (0.05)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	99.10 (1.57)

Table 4. *Cont*.

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
shuttle-c2-vs-c4	99.23 (1.72)	68.00 (30.02) *	98.21 (1.47)	98.52 (2.03)	99.26 (1.66)	90.00 (22.36) *	90.00 (22.36) *	92.00 (22.11) *
vowel0	99.70 (0.28)	92.22 (3.76) *	94.19 (1.12) *	98.14 (0.80)	98.05 (0.51)	98.89 (1.52)	97.78 (2.32)	94.00 (4.29)
yeast-0-5-6-7-9_vs_4	91.09 (3.21)	82.14 (7.22) *	84.23 (2.09) *	89.64 (2.45)	89.12 (1.88)	85.19 (8.80) *	81.24 (9.81) *	86.67 (6.59) *
yeast-1-2-8-9_vs_7	96.83 (0.37)	93.33 (8.67)	97.15 (0.43)	94.88 (1.02)	94.17 (0.85)	70.00 (17.28) *	61.67 (9.50) *	96.00 (6.85)
yeast-1-4-5-8_vs_7	95.67 (0.73)	94.67 (6.75)	82.52 (2.03) *	92.11 (0.80)	91.70 (0.03)	53.33 (4.56) *	61.67 (17.28) *	97.33 (5.23)
yeast-1_vs_7	92.59 (2.48)	90.33 (11.46)	79.86 (3.18) *	90.59 (1.97)	89.16 (1.17)	73.33 (9.13) *	75.00 (5.89) *	98.33 (3.40)v
yeast-2_vs_4	95.14 (2.05)	84.65 (9.19) *	92.55 (1.58)	94.34 (1.94)	94.16 (2.39)	90.10 (10.06)	89.19 (6.47) *	89.00 (6.51) *
yeast-2_vs_8	97.93 (0.74)	84.50 (9.74) *	89.91 (2.14) *	96.21 (1.80)	96.01 (1.43)	67.50 (14.25) *	72.50 (13.69) *	97.50 (5.10)
yeast4	97.37 (0.55)	80.92 (8.44) *	90.53 (0.90) *	95.11 (1.28)	93.62 (2.45)	73.67 (8.30) *	69.67 (4.80) *	85.29 (6.21) *
yeast5	97.64 (0.89)	93.67 (6.11)	99.22 (0.35)	98.17 (0.75)	97.97 (0.58)	97.71 (3.13)	96.60 (5.01)	96.59 (5.38)
yeast6	97.78 (0.77)	89.71 (10.93) *	95.78 (0.82)	97.43 (0.85)	96.97 (1.34)	91.43 (7.82) *	88.57 (8.14) *	95.43 (6.14)
Niger_Rice	75.60 (16.85)	72.44 (12.11)	61.33 (17.26) *	76.49 (9.45)	76.21 (8.89)	60.00 (13.33) *	58.93 (16.60) *	72.80 (14.42)
	(v/-/- *)	**(3/22/20)**	**(3/24/18)**	**(4/39/2)**	**(4/38/3)**	**(2/21/22)**	**(2/20/23)**	**(1/34/10)**

* show that the result is significantly weak compared to the result of the first column.

Table 5. F-Measure results of all methods for all datasets.

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
ecoli-0_vs_1	0.98 (0.02)	0.99 (0.01)	0.98 (0.02)	0.98 (0.02)	0.98 (0.02)	0.97 (0.04)	0.97 (0.03)	0.98 (0.02)
ecoli1	0.93 (0.04)	0.77 (0.07) *	0.92 (0.04)	0.87 (0.04) *	0.87 (0.04) *	0.94 (0.04)	0.91 (0.05)	0.92 (0.06)
ecoli2	0.98 (0.03)	0.82 (0.08) *	0.90 (0.05) *	0.89 (0.05) *	0.89 (0.05) *	0.91 (0.06) *	0.90 (0.06) *	0.96 (0.08)
ecoli3	0.94 (0.06)	0.68 (0.19) *	0.90 (0.04)	0.80 (0.07) *	0.84 (0.07) *	0.92 (0.05)	0.90 (0.06)	0.93 (0.06)
glass-0-1-2-3_vs_4-5-6	0.95 (0.06)	0.78 (0.03) *	0.95 (0.06)	0.93 (0.04)	0.93 (0.04)	0.94 (0.03)	0.96 (0.04)	0.95 (0.05)
glass0	0.92 (0.05)	0.72 (0.07) *	0.71 (0.07) *	0.87 (0.05) *	0.89 (0.04)	0.77 (0.07) *	0.77 (0.10) *	0.93 (0.04)
glass1	0.78 (0.09)	0.49 (0.09) *	0.62 (0.12) *	0.84 (0.05)v	0.87 (0.04)v	0.75 (0.07)	0.79 (0.06)	0.74 (0.09)
glass6	0.88 (0.10)	0.90 (0.17)	0.91 (0.11)	0.91 (0.04)	0.91 (0.05)	0.90 (0.08)	0.89 (0.08)	0.90 (0.07)
haberman	0.51 (0.11)	0.39 (0.08) *	0.59 (0.10)v	0.65 (0.05)v	0.62 (0.09)v	0.60 (0.10)v	0.65 (0.09)v	0.47 (0.10)
iris0	0.98 (0.03)	1.00 (0.00)	1.00 (0.00)	0.99 (0.01)	0.99 (0.01)	0.99 (0.02)	0.99 (0.02)	0.98 (0.03)
new-thyroid1	0.95 (0.05)	0.94 (0.06)	0.96 (0.04)	0.96 (0.04)	0.95 (0.04)	0.92 (0.06)	0.94 (0.07)	0.95 (0.05)
newthyroid2	0.95 (0.05)	0.94 (0.03)	1.00 (0.00)v	0.97 (0.03)	0.96 (0.04)	0.93 (0.07)	0.95 (0.06)	0.95 (0.05)
page-blocks0	0.98 (0.01)	0.83 (0.03)	0.94 (0.03)	0.92 (0.01)	0.92 (0.01)	0.95 (0.01)	0.95 (0.01)	0.99 (0.01)
pima	0.77 (0.04)	0.62 (0.03) *	0.77 (0.04)	0.80 (0.02)	0.79 (0.02)	0.74 (0.05)	0.72 (0.06)	0.73 (0.04)

Table 5. *Cont.*

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
segment0	0.98 (0.02)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.98 (0.01)	0.99 (0.01)	0.99 (0.01)
vehicle0	0.95 (0.03)	0.94 (0.01)	0.96 (0.02)	0.95 (0.02)	0.97 (0.01)	0.93 (0.03)	0.95 (0.03)	0.95 (0.03)
vehicle1	0.83 (0.03)	0.67 (0.03) *	0.78 (0.04) *	0.78 (0.04) *	0.79 (0.03)	0.76 (0.05) *	0.73 (0.05) *	0.83 (0.03)
vehicle2	0.75 (0.03)	0.96 (0.02)v	0.96 (0.02)v	0.97 (0.01)v	0.98 (0.01)v	0.95 (0.03)v	0.97 (0.02)v	0.75 (0.05)
vehicle3	0.81 (0.04)	0.64 (0.05) *	0.80 (0.08)	0.78 (0.04)	0.79 (0.03)	0.75 (0.04) *	0.74 (0.03) *	0.83 (0.04)
wisconsin	0.99 (0.01)	0.94 (0.01)	0.95 (0.03)	0.97 (0.01)	0.98 (0.01)	0.97 (0.01)	0.96 (0.02)	0.99 (0.01)
yeast1	0.85 (0.02)	0.56 (0.03) *	0.70 (0.04) *	0.76 (0.02) *	0.74 (0.02) *	0.72 (0.03) *	0.69 (0.04) *	0.82 (0.03)
yeast3	0.92 (0.04)	0.75 (0.04) *	0.90 (0.03)	0.88 (0.03) *	0.86 (0.03) *	0.92 (0.02)	0.91 (0.03)	0.92 (0.05)
abalone19	0.73 (0.14)	0.00 (0.00) *	0.87 (0.01)v	0.00 (0.00) *	0.00 (0.00) *	0.71 (0.07)	0.59 (0.16) *	0.83 (0.11)v
abalone9-18	0.82 (0.10)	0.52 (0.10) *	0.87 (0.02)	0.48 (0.10) *	0.30 (0.26) *	0.72 (0.16) *	0.67 (0.12) *	0.84 (0.09)
ecoli-0-1-3-7_vs_2-6	0.88 (0.17)	0.69 (0.41) *	0.94 (0.02)v	0.73 (0.09) *	0.86 (0.08)	0.60 (0.37) *	0.73 (0.43) *	0.86 (0.17)
ecoli4	0.90 (0.10)	0.87 (0.13)	0.96 (0.01)	0.87 (0.08)	0.87 (0.10)	0.88 (0.12)	0.91 (0.09)	0.91 (0.09)
glass-0-1-6_vs_2	0.78 (0.15)	0.07 (0.15) *	0.86 (0.03)v	0.52 (0.10) *	0.00 (0.00) *	0.54 (0.25) *	0.81 (0.14)	0.84 (0.14)v
glass-0-1-6_vs_5	0.80 (0.33)	0.59 (0.33) *	0.97 (0.02)v	0.78 (0.23)	0.93 (0.11)v	0.93 (0.15)v	0.93 (0.15)v	0.85 (0.22)
glass2	0.81 (0.16)	0.08 (0.18) *	0.89 (0.02)v	0.34 (0.22) *	0.00 (0.00) *	0.66 (0.12) *	0.65 (0.17) *	0.88 (0.12)v
glass4	0.93 (0.10)	0.75 (0.21) *	0.89 (0.04)	0.88 (0.22)	0.68 (0.21) *	0.82 (0.12) *	0.89 (0.10)	0.92 (0.11)
glass5	0.84 (0.29)	0.65 (0.41) *	0.98 (0.02)v	0.87 (0.18)	0.88 (0.14)	1.00 (0.00)v	1.00 (0.00)v	0.91 (0.15)v
page-blocks-1-3_vs_4	0.94 (0.10)	0.98 (0.03)	0.98 (0.01)	0.98 (0.04)	0.98 (0.04)	0.89 (0.09)	0.95 (0.05)	0.98 (0.06)
shuttle-c0-vs-c4	0.99 (0.02)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.02)
shuttle-c2-vs-c4	0.63 (0.40)	0.80 (0.45)v	0.98 (0.01)v	0.87 (0.18)v	0.93 (0.15)v	0.80 (0.45)v	0.80 (0.45)v	0.90 (0.29)v
vowel0	0.92 (0.04)	0.98 (0.02)v	0.94 (0.01)	0.95 (0.02)	0.94 (0.02)	0.99 (0.02)v	0.98 (0.02)v	0.94 (0.04)
yeast-0-5-6-7-9_vs_4	0.82 (0.09)	0.46 (0.20) *	0.85 (0.02)	0.67 (0.09) *	0.66 (0.08) *	0.82 (0.14)	0.77 (0.18) *	0.86 (0.07)
yeast-1-2-8-9_vs_7	0.94 (0.07)	0.16 (0.14) *	0.19 (0.18) *	0.34 (0.19) *	0.26 (0.07) *	0.65 (0.26) *	0.58 (0.19) *	0.97 (0.06)
yeast-1-4-5-8_vs_7	0.95 (0.06)	0.19 (0.17) *	0.84 (0.02) *	0.12 (0.12) *	0.03 (0.06) *	0.57 (0.09) *	0.57 (0.19) *	0.98 (0.05)
yeast-1_vs_7	0.92 (0.09)	0.30 (0.19) *	0.80 (0.03) *	0.45 (0.16) *	0.35 (0.11) *	0.68 (0.22) *	0.74 (0.07) *	0.98 (0.03)
yeast-2_vs_4	0.85 (0.09)	0.73 (0.13) *	0.93 (0.02)v	0.84 (0.06)	0.83 (0.08)	0.89 (0.13)	0.89 (0.07)	0.89 (0.07)
yeast-2_vs_8	0.82 (0.13)	0.69 (0.10) *	0.89 (0.03)v	0.68 (0.17) *	0.67 (0.14) *	0.65 (0.18) *	0.71 (0.11) *	0.98 (0.05)
yeast4	0.80 (0.10)	0.48 (0.14) *	0.91 (0.01)v	0.52 (0.11) *	0.51 (0.11) *	0.74 (0.10) *	0.67 (0.09) *	0.85 (0.08)
yeast5	0.94 (0.06)	0.63 (0.11) *	0.99 (0.00)	0.85 (0.06) *	0.83 (0.04) *	0.98 (0.03)	0.97 (0.05)	0.97 (0.06)
yeast6	0.88 (0.14)	0.47 (0.22) *	0.96 (0.01)v	0.68 (0.11) *	0.64 (0.17) *	0.91 (0.08)v	0.88 (0.09)	0.96 (0.06)v
Niger_Rice	0.73 (0.19)	0.82 (0.08)v	0.57 (0.20) *	0.76 (0.08)	0.76 (0.08)	0.55 (0.21) *	0.54 (0.23) *	0. 69 (0.17) *
	(v/-/- *)	**(4/13/28)**	**(14/22/9)**	**(4/21/20)**	**(5/22/18)**	**(7/22/16)**	**(6/23/16)**	**(6/38/1)**

* show that the result is significantly weak compared to the result of the first column.

Table 6. G-Mean Results of all methods for all datasets.

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
ecoli-0_vs_1	0.98 (0.04)	0.98 (0.00)	0.98 (0.03)	0.97 (0.02)	0.98 (0.02)	0.97 (0.05)	0.97 (0.04)	0.97 (0.04)
ecoli1	0.93 (0.06)	0.84 (0.04) *	0.92 (0.07)	0.90 (0.04)	0.90 (0.04)	0.93 (0.06)	0.91 (0.07)	0.92 (0.08)
ecoli2	0.98 (0.05)	0.90 (0.06) *	0.90 (0.07) *	0.92 (0.04) *	0.92 (0.04) *	0.91 (0.09) *	0.90 (0.09) *	0.96 (0.07)
ecoli3	0.93 (0.08)	0.83 (0.10) *	0.90 (0.07)	0.87 (0.05) *	0.90 (0.04)	0.92 (0.08)	0.90 (0.10)	0.92 (0.10)
glass-0-1-2-3_vs_4-5-6	0.94 (0.08)	0.83 (0.04) *	0.95 (0.07)	0.94 (0.05)	0.94 (0.05)	0.94 (0.06)	0.96 (0.06)	0.94 (0.07)
glass0	0.92 (0.08)	0.79 (0.05) *	0.71 (0.13) *	0.87 (0.07) *	0.88 (0.06)	0.77 (0.10) *	0.76 (0.14) *	0.92 (0.07)
glass1	0.77 (0.12)	0.60 (0.11) *	0.65 (0.09) *	0.82 (0.07)	0.86 (0.06)v	0.75 (0.09)	0.79 (0.10)	0.74 (0.12)
glass6	0.88 (0.14)	0.93 (0.05)v	0.92 (0.10)v	0.95 (0.05)v	0.94 (0.04)v	0.89 (0.11)	0.89 (0.10)	0.90 (0.13)
haberman	0.56 (0.14)	0.53 (0.06)	0.62 (0.09)v	0.70 (0.07)v	0.68 (0.09)v	0.62 (0.11)v	0.65 (0.11)v	0.56 (0.19)
iris0	0.98 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.00)	0.99 (0.00)	0.98 (0.00)	0.99 (0.00)	0.98 (0.00)
new-thyroid1	0.95 (0.07)	0.96 (0.04)	0.95 (0.07)	0.96 (0.03)	0.97 (0.04)	0.91 (0.10)	0.93 (0.10)	0.95 (0.07)
newthyroid2	0.95 (0.07)	0.96 (0.04)	1.00 (0.00)v	0.98 (0.03)	0.97 (0.04)	0.92 (0.09)	0.94 (0.07)	0.95 (0.07)
page-blocks0	0.98 (0.01)	0.88 (0.00) *	0.93 (0.03) *	0.95 (0.00)	0.95 (0.00)	0.95 (0.02)	0.95 (0.02)	0.99 (0.01)
pima	0.77 (0.05)	0.70 (0.07)	0.76 (0.05)	0.79 (0.03)	0.78 (0.03)	0.73 (0.06)	0.71 (0.07)	0.73 (0.06)
segment0	0.98 (0.02)	0.99 (0.00)	0.99 (0.01)	0.99 (0.00)	0.99 (0.00)	0.98 (0.01)	0.99 (0.01)	0.99 (0.01)
vehicle0	0.94 (0.04)	0.96 (0.01)	0.95 (0.03)	0.96 (0.02)	0.97 (0.01)	0.93 (0.04)	0.94 (0.03)	0.95 (0.04)
vehicle1	0.82 (0.05)	0.77 (0.04) *	0.77 (0.05) *	0.81 (0.05)	0.82 (0.04)	0.75 (0.07) *	0.72 (0.06) *	0.82 (0.05)
vehicle2	0.75 (0.06)	0.97 (0.02)v	0.96 (0.03)v	0.97 (0.01)v	0.98 (0.01)v	0.95 (0.03)v	0.96 (0.03)v	0.75 (0.07)
vehicle3	0.80 (0.07)	0.74 (0.05) *	0.79 (0.08)	0.81 (0.04)	0.82 (0.04)	0.74 (0.06) *	0.73 (0.06) *	0.82 (0.06)
wisconsin	0.99 (0.01)	0.96 (0.02)	0.95 (0.04)	0.97 (0.01)	0.97 (0.01)	0.97 (0.02)	0.95 (0.03)	0.99 (0.01)
yeast1	0.83 (0.03)	0.66 (0.02) *	0.69 (0.05) *	0.78 (0.03) *	0.76 (0.03) *	0.71 (0.04) *	0.69 (0.06) *	0.81 (0.03)
yeast3	0.92 (0.06)	0.85 (0.04) *	0.89 (0.03) *	0.92 (0.02)	0.91 (0.02)	0.92 (0.04)	0.90 (0.05)	0.91 (0.07)
abalone19	0.73 (0.19)	0.00 (0.00) *	0.86 (0.01)v	0.00 (0.00) *	0.00 (0.00) *	0.68 (0.19) *	0.63 (0.20) *	0.83 (0.16)v
abalone9-18	0.82 (0.15)	0.68 (0.03) *	0.87 (0.02)	0.60 (0.03) *	0.49 (0.07) *	0.72 (0.16) *	0.65 (0.20) *	0.84 (0.14)
ecoli-0-1-3-7_vs_2-6	0.83 (0.21)	0.84 (0.07)	0.93 (0.03)v	0.83 (0.04)	0.93 (0.04)v	0.73 (0.30) *	0.85 (0.31)	0.80 (0.25)
ecoli4	0.90 (0.15)	0.92 (0.04)	0.96 (0.02)v	0.92 (0.03)	0.92 (0.04)	0.84 (0.20) *	0.90 (0.16)	0.91 (0.14)
glass-0-1-6_vs_2	0.73 (0.22)	0.22 (0.05) *	0.85 (0.05)v	0.61 (0.05) *	0.00 (0.00) *	0.60 (0.30) *	0.83 (0.19)v	0.80 (0.19)v
glass-0-1-6_vs_5	0.85 (0.28)	0.77 (0.09) *	0.97 (0.02)v	0.90 (0.05)	0.97 (0.05)v	0.95 (0.00)v	0.95 (0.00)v	0.86 (0.25)
glass2	0.73 (0.18)	0.26 (0.04) *	0.88 (0.05)v	0.49 (0.04) *	0.00 (0.00) *	0.60 (0.22) *	0.67 (0.17) *	0.85 (0.16)v
glass4	0.94 (0.13)	0.90 (0.08) *	0.89 (0.05) *	0.94 (0.08)	0.77 (0.07) *	0.81 (0.16) *	0.90 (0.16)	0.93 (0.14)
glass5	0.88 (0.25)	0.83 (0.12) *	0.97 (0.03)v	0.93 (0.04)	0.94 (0.05)v	1.00 (0.00)v	1.00 (0.00)v	0.93 (0.18)
page-blocks-1-3_vs_4	0.94 (0.11)	1.00 (0.00)v	0.98 (0.01)	1.00 (0.00)v	0.99 (0.02)	0.88 (0.10)	0.94 (0.00)	0.98 (0.06)

Table 6. *Cont.*

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
shuttle-c0-vs-c4	0.99 (0.02)	0.99 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.02)
shuttle-c2-vs-c4	0.71 (0.44)	0.89 (0.00)v	0.98 (0.02)v	0.89 (0.00)v	0.95 (0.00)v	0.89 (0.00)v	0.89 (0.00)v	0.94 (0.24)v
vowel0	0.92 (0.05)	0.99 (0.00)v	0.94 (0.02)	0.97 (0.02)	0.96 (0.03)	0.99 (0.02)v	0.98 (0.03)v	0.94 (0.05)
yeast-0-5-6-7-9_vs_4	0.82 (0.13)	0.63 (0.06) *	0.84 (0.03)	0.77 (0.03)	0.77 (0.05)	0.85 (0.14)	0.80 (0.19)	0.86 (0.1)
yeast-1-2-8-9_vs_7	0.93 (0.07)	0.32 (0.00) *	0.33 (0.00) *	0.48 (0.04) *	0.41 (0.02) *	0.70 (0.25) *	0.61 (0.27) *	0.96 (0.00)
yeast-1-4-5-8_vs_7	0.94 (0.06)	0.36 (0.04) *	0.82 (0.03) *	0.26 (0.00) *	0.14 (0.00) *	0.52 (0.18) *	0.61 (0.22) *	0.97 (0.00)
yeast-1_vs_7	0.90 (0.00)	0.51 (0.08) *	0.80 (0.04) *	0.57 (0.04) *	0.49 (0.03) *	0.73 (0.2) *	0.75 (0.15) *	0.98 (0.00)v
yeast-2_vs_4	0.85 (0.12)	0.82 (0.04)	0.92 (0.02)v	0.90 (0.05)	0.89 (0.05)	0.90 (0.17)	0.89 (0.10)	0.89 (0.09)
yeast-2_vs_8	0.84 (0.15)	0.74 (0.00) *	0.90 (0.03)	0.74 (0.00) *	0.73 (0.00) *	0.67 (0.31) *	0.72 (0.26) *	0.97 (0.00)v
yeast4	0.81 (0.14)	0.61 (0.00) *	0.90 (0.01)v	0.63 (0.03) *	0.68 (0.04) *	0.74 (0.15) *	0.70 (0.16) *	0.85 (0.12)
yeast5	0.93 (0.09)	0.81 (0.04) *	0.99 (0.00)v	0.93 (0.03)	0.91 (0.03)	0.97 (0.00)	0.96 (0.05)	0.96 (0.06)
yeast6	0.90 (0.14)	0.67 (0.05) *	0.96 (0.01)	0.78 (0.00) *	0.76 (0.04) *	0.91 (0.10)	0.88 (0.10)	0.95 (0.09)
Niger_Rice	0.76 (0.24)	0.59 (0.18) *	0.60 (0.26)	0.77 (0.14)	0.76 (0.15)	0.60 (0.29) *	0.60 (0.34) *	0.73 (0.22)
	(v/-/-*)	**(5/13/27)**	**(14/20/11)**	**(5/26/14)**	**(8/24/13)**	**(6/21/18)**	**(7/24/14)**	**(6/39/0)**

* show that the result is significantly weak compared to the result of the first column.

Table 7. Area under ROC results of all methods for all datasets.

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
ecoli-0_vs_1	0.99 (0.02)	1.00 (0.01)	1.00 (0.00)	0.99 (0.01)	0.99 (0.01)	0.99 (0.02)	0.99 (0.02)	0.99 (0.02)
ecoli1	0.98 (0.03)	0.96 (0.02)	0.97 (0.02)	0.96 (0.02)	0.96 (0.02)	0.95 (0.04)	0.95 (0.04)	0.95 (0.07)
ecoli2	0.99 (0.03)	0.96 (0.03)	0.95 (0.06)	0.97 (0.02)	0.98 (0.02)	0.94 (0.05) *	0.95 (0.06) *	0.97 (0.05)
ecoli3	0.99 (0.01)	0.90 (0.09) *	0.93 (0.07) *	0.96 (0.02)	0.97 (0.02)	0.92 (0.07) *	0.92 (0.07) *	0.94 (0.05) *
glass-0-1-2-3_vs_4-5-6	0.98 (0.02)	0.95 (0.04)	0.96 (0.08)	0.98 (0.02)	0.99 (0.01)	0.98 (0.03)	0.97 (0.04)	0.95 (0.05)
glass0	0.96 (0.04)	0.84 (0.05) *	0.79 (0.05) *	0.94 (0.03)	0.95 (0.02)	0.86 (0.07) *	0.86 (0.07) *	0.97 (0.03)
glass1	0.85 (0.08)	0.71 (0.03) *	0.68 (0.04) *	0.90 (0.04)v	0.93 (0.03)v	0.84 (0.06)	0.86 (0.06)	0.83 (0.08)
glass6	0.97 (0.05)	0.95 (0.07)	0.91 (0.16) *	0.96 (0.04)	0.98 (0.02)	0.94 (0.06)	0.91 (0.08) *	0.96 (0.06)
haberman	0.64 (0.10)	0.68 (0.08)	0.63 (0.11)	0.78 (0.04)v	0.70 (0.04)	0.63 (0.08)	0.65 (0.08)	0.61 (0.05)
iris0	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.01)	0.99 (0.01)	0.99 (0.02)	0.99 (0.02)	0.99 (0.03)
new-thyroid1	0.99 (0.03)	1.00 (0.00)	0.99 (0.02)	1.00 (0.01)	0.98 (0.03)	0.98 (0.03)	0.96 (0.07)	0.98 (0.04)
newthyroid2	0.99 (0.03)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.02)	0.99 (0.02)	0.97 (0.05)	0.98 (0.04)
page-blocks0	1.00 (0.00)	0.97 (0.02)	0.98 (0.02)	0.99 (0.00)	0.99 (0.00)	0.99 (0.01)	0.98 (0.01)	1.00 (0.00)

Table 7. *Cont.*

Dataset	MLPUS_Bagging	MLP Classifier	MLP Under-Sampling	SMOTE_Bagging	SMOTE_Boost	Under-Bagging	RUS_Boost	MLPUS_Boost
pima	0.84 (0.03)	0.82 (0.04)	0.84 (0.03)	0.86 (0.02)	0.85 (0.02)	0.80 (0.05)	0.78 (0.04) *	0.81 (0.03)
segment0	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.01)	1.00 (0.01)
vehicle0	0.99 (0.02)	0.99 (0.01)	0.99 (0.01)	0.99 (0.00)	1.00 (0.00)	0.98 (0.02)	0.99 (0.01)	0.99 (0.03)
vehicle1	0.91 (0.03)	0.90 (0.02)	0.85 (0.05) *	0.90 (0.02)	0.91 (0.02)	0.82 (0.04) *	0.79 (0.05) *	0.90 (0.03)
vehicle2	0.84 (0.03)	0.99 (0.02)v	0.98 (0.02)v	0.99 (0.00)v	1.00 (0.00)v	0.98 (0.02)v	0.99 (0.01)v	0.84 (0.04)
vehicle3	0.89 (0.04)	0.87 (0.03)	0.86 (0.07)	0.91 (0.02)	0.91 (0.02)	0.83 (0.03) *	0.83 (0.04) *	0.90 (0.03)
wisconsin	1.00 (0.00)	0.99 (0.00)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	1.00 (0.01)
yeast1	0.91 (0.02)	0.79 (0.03) *	0.79 (0.04) *	0.86 (0.02) *	0.85 (0.02) *	0.79 (0.02) *	0.76 (0.03) *	0.90 (0.02)
yeast3	0.97 (0.02)	0.97 (0.01)	0.96 (0.01)	0.98 (0.01)	0.97 (0.01)	0.97 (0.02)	0.96 (0.02)	0.98 (0.02)
abalone19	0.82 (0.16)	0.83 (0.03)	0.94 (0.01)v	0.86 (0.05)	0.77 (0.05) *	0.77 (0.10) *	0.70 (0.11) *	0.90 (0.13)v
abalone9-18	0.90 (0.09)	0.92 (0.04)	0.94 (0.01)	0.88 (0.03)	0.83 (0.05) *	0.79 (0.17) *	0.74 (0.19) *	0.94 (0.07)
ecoli-0-1-3-7_vs_2-6	0.89 (0.24)	0.94 (0.12)	0.99 (0.01)v	0.93 (0.09)	0.98 (0.04)v	1.00 (0.00)v	0.85 (0.22)	0.88 (0.23)
ecoli4	0.98 (0.05)	0.99 (0.01)	0.99 (0.01)	0.98 (0.03)	0.99 (0.01)	0.95 (0.11)	1.00 (0.00)	0.98 (0.06)
glass-0-1-6_vs_2	0.89 (0.15)	0.81 (0.14) *	0.88 (0.04)	0.87 (0.10)	0.80 (0.08) *	0.78 (0.25) *	0.87 (0.16)	0.90 (0.15)
glass-0-1-6_vs_5	0.94 (0.15)	0.95 (0.06)	1.00 (0.01)	0.98 (0.01)	1.00 (0.01)	0.95 (0.11)	0.95 (0.11)	0.87 (0.18) *
glass2	0.90 (0.12)	0.74 (0.13) *	0.90 (0.03)	0.91 (0.05)	0.83 (0.04) *	0.74 (0.26) *	0.75 (0.17) *	0.91 (0.13)
glass4	0.99 (0.03)	0.98 (0.02)	0.97 (0.02)	0.95 (0.11)	0.91 (0.19) *	0.88 (0.11) *	0.84 (0.15) *	0.93 (0.09)
glass5	0.95 (0.12)	0.89 (0.21) *	1.00 (0.01)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.93 (0.11)
page-blocks-1-3_vs_4	0.99 (0.05)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.96 (0.04)	1.00 (0.01)	0.99 (0.03)
shuttle-c0-vs-c4	1.00 (0.01)	0.99 (0.02)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.02)
shuttle-c2-vs-c4	0.89 (0.24)	1.00 (0.00)v	1.00 (0.01)v	0.95 (0.11)v	0.95 (0.11)	0.90 (0.22)	0.90 (0.22)	0.94 (0.17)
vowel0	0.96 (0.03)	1.00 (0.00)	0.98 (0.00)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.98 (0.03)
yeast-0-5-6-7-9_vs_4	0.91 (0.04)	0.82 (0.10) *	0.93 (0.02)	0.91 (0.05)	0.90 (0.05)	0.89 (0.07)	0.92 (0.03)	0.95 (0.04)
yeast-1-2-8-9_vs_7	0.99 (0.03)	0.70 (0.10) *	0.80 (0.07) *	0.86 (0.06) *	0.82 (0.02) *	0.72 (0.19) *	0.70 (0.12) *	0.98 (0.06)
yeast-1-4-5-8_vs_7	0.99 (0.03)	0.70 (0.14) *	0.88 (0.02) *	0.83 (0.04) *	0.78 (0.07) *	0.66 (0.05) *	0.63 (0.13) *	0.97 (0.05)
yeast-1_vs_7	0.98 (0.03)	0.81 (0.07) *	0.87 (0.02) *	0.91 (0.05) *	0.86 (0.05) *	0.84 (0.07) *	0.84 (0.09) *	0.98 (0.04)
yeast-2_vs_4	0.95 (0.04)	0.94 (0.06)	0.98 (0.01)	0.98 (0.02)	0.98 (0.01)	0.96 (0.06)	0.97 (0.03)	0.97 (0.02)
yeast-2_vs_8	0.92 (0.12)	0.85 (0.14) *	0.93 (0.02)	0.92 (0.07)	0.91 (0.06)	0.76 (0.17) *	0.73 (0.19) *	0.99 (0.03)
yeast4	0.91 (0.07)	0.88 (0.05)	0.97 (0.00)v	0.96 (0.02)	0.93 (0.02)	0.86 (0.10)	0.81 (0.05) *	0.95 (0.03)
yeast5	0.99 (0.02)	0.98 (0.03)	1.00 (0.00)	0.99 (0.01)	0.99 (0.00)	0.99 (0.03)	0.97 (0.05)	0.99 (0.02)
yeast6	0.97 (0.05)	0.95 (0.04)	0.99 (0.00)	0.92 (0.08)	0.95 (0.02)	0.90 (0.07)	0.90 (0.12)	1.00 (0.01)
Niger_Rice	0.86 (0.17)	0.76 (0.21) *	0.64 (0.26) *	0.87 (0.08)	0.84 (0.08)	0.72 (0.19) *	0.75 (0.21) *	0.80 (0.18) *
	(v/-/- *)	**(2/30/13)**	**(5/30/10)**	**(4/37/4)**	**(3/33/9)**	**(2/27/16)**	**(1/26/18)**	**(1/41/3)**

* show that the result is significantly weak compared to the result of the first column.

In Table 4, the accuracy of our method revealed on the 45 datasets is twenty times significantly better than the MLP classifier in the original imbalanced dataset. At the same time, it is significantly weak only three times. Also, the proposed method wins respectively eighteen, two, three, twenty-two, twenty-three, and ten times against MLPUS, SMOTEBagging, SMOTEBoosting, Under-Bagging, RUSBoost, and MLPUS_Boost methods. It loses three, four, four, two, and two times against MLP Under-Sampling, SMOTEBagging, SMOTEBoosting, Under-Bagging, and RUSBoost methods. In contrast, it loses one against the MLPUS_Boost technique.

The different F-measure results on the 45 datasets show in Table 5 that our method respectively wins twenty-eight, nine, twenty, eighteen, sixteen, and sixteen times against MLP classifier on the original imbalanced dataset, MLPUS, SMOTEBagging, SMOTEBoosting, Under-Bagging, and RUSBoost methods. However, It wins only one time against the MLPUS_Boost method. At the same time, Our proposed method respectively loses four, fourteen, four, five, seven, six, and six times against MLP, MLPUS, SMOTEBagging, SMOTEBoosting, Under-Bagging, RUSBoost, and MLPUS_Boost methods on the F-measure metric.

In Table 6, the results of G-mean on 45 imbalanced datasets reveal that our method wins respectively twenty-seven, eleven, fourteen, thirteen, eighteen, and fourteen times against MLP classifier in the original imbalanced dataset, MLPUS, SMOTEBagging, SMOTEBoosting, Under-Bagging, and RUSBoost methods. However, the MLPUSBagging did not win against the MLPUS_Boost technique. At the same time, the proposed method loses five, fourteen, five, eight, six, seven, and six times, respectively, against MLP, MLPUS, SMOTEBagging, SMOTEBoosting, Under-Bagging, RUSBoost, and MLPUS_Boost methods in terms of G-mean on 45 imbalanced datasets.

The different Area Under ROC results in Table 7 shows that for the 45 imbalanced datasets, our proposed method wins thirteen, ten, four, nine, sixteen, eighteen, and three times respectively against MLP, MLPUS, SMOTEBagging, SMOTEBoosting, Under-Bagging, RUSBoost, and MLPUS_Boost methods. The MLPUS_Bagging loses two, five, four, three, and two times, respectively, against MLP, MLPUS, SMOTEBagging, SMOTEBoosting, and Under-Bagging methods. While the MLPUSBagging loses only ones against the RUSBoost and MLPUS_Boost methods in Area Under ROC results on the 45 imbalanced datasets.

Apart from accuracy, which is not an adequate metric for dealing with imbalanced datasets, our proposed method has shown significant results for the Niger_Rice dataset with other metrics such as F-Measure, G-mean, and the area under the ROC. These results of the F-measure from the Niger_Rice dataset show that only one method is significantly better considering the p-value of 0.05. On the other hand, these same results indicate that our proposed method is considerably better than the five methods with the p-value (see Table 5). Thus, the results of F-Measure for the Niger_Rice dataset are 0.73, 0.82, 0.57, 0.76, 0.76, 0.55, 0.54, and 0.69, respectively, for our proposed method, the classifier MLP on the original imbalanced dataset, MLPUS, SMOTE_Bagging, SMOTE_Boost, Under-Bagging, RUS_Boost MLPUS_Boost methods.

Beyond F-measure, the results of our method for the Niger_Rice dataset are much better for the G-Mean metric, with three methods significantly weak and no method better than our proposed method with the p-value 0.05 (see Table 6). This result of the Niger_Rice dataset means that the true positive and negative rates are distributed well with our proposed method. Thus, the G-mean results for the Niger_Rice dataset are 0.76, 0.59, 0.60, 0.77, 0.76, 0.60, 0.60, and 0.22, respectively, for our proposed method, the classifier MLP on the original imbalanced dataset, MLPUS, SMOTE_Bagging, SMOTE_Boost, Under-Bagging, RUS_Boost MLPUS_Boost methods.

These results are even better for the area under the ROC metric on the Niger_Rice dataset. Our approach provides better results than others methods (see Figure 2). The areas under the ROC curve results show us that whatever the threshold, the true positive rate against the false positive rate is significantly better with our method than others except for the SMOTE_Bagging method. The Roc curve of our proposed method is far superior to the other curves; only the Roc curve of SMOTEBagging and SMOTEBoost are competitive with it.

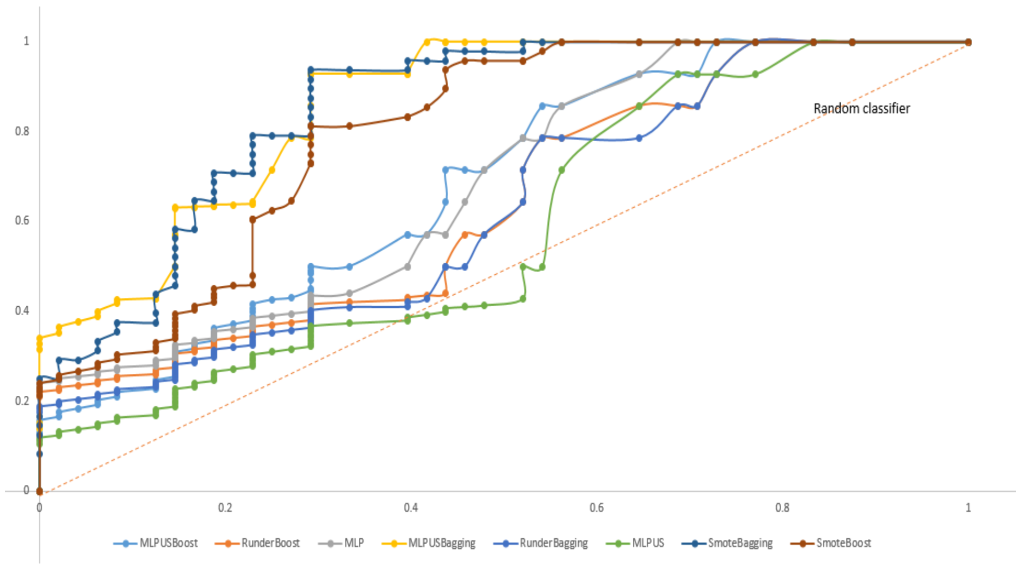

Figure 2. ROC curves result on Niger_Rice dataset for different methods.

Besides these promising results of our method, it presents some limitations. The first limit is a reduced number of samples from the minority class. Because when this sample is reduced and selected, the square root of this reduced number can lead to a loss of essential samples in both classes. Another drawback is choosing the best activation function to train the model. In this experiment, the sigmoid function was used. However, the other activation functions are being explored in future experiments. In this experiment, the results show that the IR is not a factor that influences the model's training when the samples of the minority class are not sufficiently reduced.

5. Conclusions

This paper has proposed a hybrid method composed of the under-sampling and ensemble methods to deal with a class imbalance problem for a real-life Niger_Rice dataset. The under-sampling method consisted of taking the samples by evaluating them with the stochastic measure by training the Multilayer perceptron. The ensemble methods used in this research are bagging and boosting. The proposed method, MLPUS_Bagging, consists of aggregating the different training sets provided after the under-sampling of the original training set. To measure and quantify our method, we compare it with six other hybrid methods combining the preprocessing methods and the ensemble methods and the combination of our MLPUS with Boosting. Beyond our real-life Niger_Rice dataset, forty-four other datasets were used in this study to understand the impact of our method on the well-known imbalanced datasets. The results clearly show that on the 45 imbalanced datasets, our method is better than the other methods concerning metrics such as F-measure, G-mean, and ROC curve with a p-value of 0.05. The results of our method for the Niger_Rice real-life dataset are 75.6, 0.73, 0.76, and 0.86, respectively, for accuracy, F-measure, G-mean, and ROC.

In comparison, the MLP classifier on the original imbalance Niger_Rice dataset gives results 72.44, 0.82, 0.59, and 0.76 respectively for accuracy, F-measure, G-mean, and ROC. Our hybrid method combining the under-sampling and ensemble methods gave convincing results. However, in future work, we will try to study another oversampling method using the evaluation of the stochastic measure by training the multilayer perceptron. We will also explore the hybrid method combining oversampling using the evaluation of the stochastic measure by training the multilayer perceptron and ensemble methods. We will also explore how our proposed method can deal with the multi-class imbalance problem in future work.

Author Contributions: Conceptualization, M.D. and S.X.; methodology, M.A.E.; software, E.D.E.; validation, A.F., M.D. and S.X.; formal analysis, M.D.; investigation, A.O.A.; resources, M.D.; data curation, S.X.; writing—original draft preparation, M.D.; writing—review and editing, S.X.; visualization, M.A.E.; supervision, S.X.; project administration, E.D.E.; funding acquisition, M.D., A.F., A.O.A. and S.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the KEEL dataset repository at https://sci2s.ugr.es/keel/imbalanced.php, accessed on 22 July 2021 [45]; and Niger_Rice dataset can be found here: https://github.com/moussdiall/Imbalanced_dataset_Niger_Rice, accessed on 22 July 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, L.; Traore, S.; Ge, J.; Li, Y.; Wang, S.; Zhu, G. Using boosted tree regression and artificial neural networks to forecast upland rice yield under climate change in Sahel. *Comput. Electron. Agric.* **2019**, *166*, 105031. [CrossRef]
2. Zwarts, L.; Beukering, P.; Van Koné, B.; Wymenga, E.; Taylor, D. The Economic and Ecological Effects of Water Management Choices in the Upper Niger River: Development of Decision Support Methods. *Int. J. Water Resour. Dev.* **2006**, *22*, 135–156. [CrossRef]
3. McCoy, J.T.; Auret, L. Machine learning applications in minerals processing: A review. *Miner. Eng.* **2019**, *132*, 95–109. [CrossRef]
4. Prati, R.C.; Batista, G.E.A.P.A.; Silva, D.F. Class imbalance revisited: A new experimental setup to assess the performance of treatment methods. *Knowl. Inf. Syst.* **2015**, *45*, 247–270. [CrossRef]
5. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Editorial. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 1–6. [CrossRef]
6. Kuncheva, L.I.; Arnaiz-González, Á.; Díez-Pastor, J.-F.; Gunn, I.A.D. Instance selection improves geometric mean accuracy: A study on imbalanced data classification. *Prog. Artif. Intell.* **2019**, *8*, 215–228. [CrossRef]
7. Fan, W.; Stolfo, S.J.; Chan, P.K. AdaCost: Misclassification Cost-sensitive Boosting. *Icml* **1999**, *99*, 97–105.
8. Sun, Y.; Kamel, M.S.; Wong, A.K.C.; Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [CrossRef]
9. Díez-Pastor, J.F.; Rodríguez, J.J.; García-Osorio, C.I.; Kuncheva, L.I. Diversity techniques improve the performance of the best imbalance learning ensembles. *Inf. Sci.* **2015**, *325*, 98–117. [CrossRef]
10. Kaur, H.; Pannu, H.S.; Malhi, A.K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.* **2019**, *52*, 136. [CrossRef]
11. Barandela, R.; Sánchez, J.S.; Valdovinos, R.M. New Applications of Ensembles of Classifiers. *Pattern Anal. Appl.* **2003**, *6*, 245–256. [CrossRef]
12. Wilson, D.L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Trans. Syst. Man Cybern.* **1972**, *2*, 408–421. [CrossRef]
13. Tomek, I. Two modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 769–772.
14. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
15. Jo, T.; Japkowicz, N. Class imbalances versus small disjuncts. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 40–49. [CrossRef]
16. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
17. Bunkhumpornpat, C.; Sinapiromsaran, K.; Lursinsap, C. Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 475–482.
18. Han, H.; Wang, W.-Y.; Mao, B.-H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *International Conference on Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887.
19. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328.
20. Kovács, G. Smote-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing* **2019**, *366*, 352–354. [CrossRef]
21. Xu, Z.; Shen, D.; Nie, T.; Kou, Y. A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. *J. Biomed. Inform.* **2020**, *107*, 103465. [CrossRef]

22. Sáez, J.A.; Luengo, J.; Stefanowski, J.; Herrera, F. SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inf. Sci.* **2015**, *291*, 184–203. [CrossRef]
23. Zhang, X.; Zhu, C.; Wu, H.; Liu, Z.; Xu, Y. An Imbalance Compensation Framework for Background Subtraction. *IEEE Trans. Multimed.* **2017**, *19*, 2425–2438. [CrossRef]
24. Li, J.; Fong, S.; Wong, R.K.; Chu, V.W. Adaptive multi-objective swarm fusion for imbalanced data classification. *Inf. Fusion* **2018**, *39*, 1–24. [CrossRef]
25. Bailey, J.; Khan, L.; Washio, T.; Dobbie, G.; Huang, J.Z.; Wang, R. Advances in knowledge discovery and data mining: 20th pacific-asia conference, PAKDD 2016 Auckland, New Zealand, April 19–22, 2016 proceedings, part I. *Lect. Notes Comput. Sci.* **2016**, *9651*, 14–26.
26. Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
27. Breiman, L. Bagging predictions. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
28. Freund, Y.; Schapire, R.E.; Hill, M. Experiments with a New Boosting Algorithm. *Icml* **1996**, *96*, 148–156.
29. Wang, S.; Yao, X. Diversity analysis on imbalanced data sets by using ensemble models. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*; IEEE: Piscataway, NJ, USA, 2009; pp. 324–331.
30. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In *European Conference on Principles of Data Mining and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 107–119.
31. Chen, S.; He, H.; Garcia, E.A. RAMOBoost: Ranked minority oversampling in boosting. *IEEE Trans. Neural Netw.* **2010**, *21*, 1624–1642. [CrossRef] [PubMed]
32. Seiffert, C.; Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2010**, *40*, 185–197. [CrossRef]
33. Galar, M.; Fernández, A.; Barrenechea, E.; Herrera, F. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognit.* **2013**, *46*, 3460–3471. [CrossRef]
34. Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* **2009**, *39*, 539–550.
35. Díez-Pastor, J.F.; Rodríguez, J.J.; García-Osorio, C.; Kuncheva, L.I. Random Balance: Ensembles of variable priors classifiers for imbalanced data. *Knowl. Based Syst.* **2015**, *85*, 96–111. [CrossRef]
36. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 463–484. [CrossRef]
37. Błaszczyński, J.; Deckert, M.; Stefanowski, J.; Wilk, S. Integrating selective pre-processing of imbalanced data with Ivotes ensemble. *Lect. Notes Comput. Sci.* **2010**, *6086*, 148–157.
38. Babar, V.; Ade, R. A Novel Approach for Handling Imbalanced Data in Medical Diagnosis using Undersampling Technique. *Commun. Appl. Electron.* **2016**, *5*, 36–42. [CrossRef]
39. Li, H.; Jiang, X.; Huo, G.; Su, C.; Wang, B. A novel feed rate scheduling method based on Sigmoid function with chord error and kinematics constraints. *arXiv* **2021**, arXiv:2105.05434.
40. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
41. Brown, M.J.; Hutchinson, L.A.; Rainbow, M.J.; Deluzio, K.J.; De Asha, A.R. A comparison of self-selected walking speeds and walking speed variability when data are collected during repeated discrete trials and during continuous walking. *J. Appl. Biomech.* **2017**, *33*, 384–387. [CrossRef] [PubMed]
42. Ramachandran, P.; Zoph, B.; Le Google Brain, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
43. Clevert, D.-A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
44. Basirat, M.; Roth, P.M. The Quest for the Golden Activation Function. *arXiv* **2018**, arXiv:1808.00783.
45. Ernández, A.F.; Uengo, J.L.; Errac, J.D. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *J. Mult. Valued Log. Soft Comput.* **2011**, *17*, 255–287.
46. Kohonen, T. An introduction to neural computing. *Neural Netw.* **1988**, *1*, 3–16. [CrossRef]