J Multidisciplinary Scientific Journal

MDPI

Article

Filtering-Based Instance Selection Method for Overlapping Problem in Imbalanced Datasets

Marcio Rubbo and Leandro A. Silva *

Graduate Program in Electrical Engineering and Computing, Mackenzie Presbyterian University, Rua da Consolação, 896, Prédio 30, Consolação, São Paulo 01302-907, Brazil; mrubbo@gmail.com
* Correspondence: leandroaugusto.silva@mackenzie.br

Abstract: The overlapping problem occurs when a region of the dimensional data space is shared in a similar proportion by different classes. It has an impact on a classifier's performance due to the difficulty in correctly separating the classes. Further, an imbalanced dataset consists of a situation in which one class has more instances than another, and this is another aspect that impacts a classifier's performance. In general, these two problems are treated separately. On the other hand, Prototype Selection (PS) approaches are employed as strategies for selecting appropriate instances from a dataset by filtering redundant and noise data, which can cause misclassification performance. In this paper, we introduce Filtering-based Instance Selection (FIS), using as a base the Self-Organizing Maps Neural Network (SOM) and information entropy. In this sense, SOM is trained with a dataset, and, then, the instances of the training set are mapped to the nearest prototype (SOM neurons). An analysis with entropy is conducted in each prototype region. From a threshold, we propose three decision methods: filtering the majority class (H-FIS (High Filter IS)), the minority class (L-FIS (Low Filter IS)), and both classes (B-FIS). The experiments using artificial and real dataset showed that the methods proposed in combination with 1NN improved the accuracy, F-Score, and G-mean values when compared with the 1NN classifier without the filter methods. The FIS approach is also compatible with the approaches mentioned in the relevant literature.

Keywords: prototype selection; self-organizing maps; imbalanced datasets; overlapping problem

Citation: Rubbo, M.; Silva, L.A. Filtering-Based Instance Selection Method for Overlapping Problem in Imbalanced Datasets. *J* **2021**, *4*, 308–327. https://doi.org/10.3390/j4030024

Academic Editor: José Antonio Sáez

Received: 27 April 2021
Accepted: 11 June 2021
Published: 9 July 2021

1. Introduction

A situation that can cause problems in the performance of a classifier is class overlapping. It occurs when a region of the dimensional data space is shared in a similar proportion by different classes, resulting in misclassification.

In addition to the problems that impact a classifier's performance, an imbalanced dataset is a rather common problem that appears in different areas of knowledge such as medicine, psychology, industry, and in some other areas [1–4]. The imbalance problem comprises a situation in which one of the classes (i.e., the majority class) has more instances than the other class (i.e., the minority class).

Studies on this subject indicate that the overlapping of data can have a greater impact in terms of loss of accuracy than an imbalance of classes [5–7]. In the relevant literature, most of the works deal with these problems separately.

The classical algorithms to minimize class overlapping are defined as follows: Edited Nearest Neighbor (ENN), which removes the instance with the class that disagrees with neighborhood classes [8]; Decremental Reduction Optimization Procedure 3 (DROP3), a classification process the is conducted and the misclassification instances are removed [8]; Adaptive Threshold-based Instance Selection Algorithm 1 (ATISA1), which is similar to DROP3 with the difference that the instance classified as correct is selected [9]; and Ranking-based Instance Selection (RIS), which is a ranking-based approach defined by the relation of neighbor instance classes [10].

The class imbalance problems can be defined into data-level approaches and algorithmic-level approaches [11,12]. Data-level approaches consist of a sampling dataset realized, in a random way, with the objective of an undersampling or oversampling dataset. Algorithm-level approaches consist of an ensemble of classification algorithms trained with different dataset samples.

However, more recent work has started to address these two problems in a unique way. Vuttipittayamongkol and Elyan proposed an overlap-based undersampling method for maximizing the visibility of the minority class instances in the overlapping region [1]. Elyan et al. proposed a hybrid approach aimed at reducing the dominance of the majority class instances using class decomposition and by increasing the minority class instances using an oversampling method [2]. Yuan et al. proposed a Density-Based Adaptive k-Nearest Neighbors method (DBANN), which can handle imbalance- and overlapping-related problems simultaneously. To do so, a simple but effective distance adjustment strategy has been developed to adaptively find the most reliable query neighbors [4].

One classifier that is especially vulnerable to the overlapping problem and imbalanced dataset is the supervised method of Instance-Based Learning (IBL) [6]. The *k*NN classifier is a traditional IBL algorithm, which works using the nearest neighbor approach to classify an instance [13].

Several strategies have been proposed to solve these issues for *k*NN. One of these strategies is to use data reduction to select instances to enhance the classification process. PS, as this method is called, is similar to a pre-processing step before algorithm training that focuses on selection of instances that can contribute to improving the classification performance and reducing the training (or comparative) timing.

Different methods for this approach have been proposed. Garcia et al. organized all these works in a taxonomy and compared the different methods that try to improve the classification and performance with special reference to *k*NN [14].

There are different strategies to define the subset of prototypes selected in PS. From among the techniques available in the relevant literature, it is possible to highlight the random methods, distance methods, and clustering and evolutionary algorithms [15].

This work considers the clustering techniques as it evaluates the use of Kohonen's SOM [16]. SOM is an unsupervised method that clusters the instances according to their similarities. The standard version of this algorithm arranges the instances, according to their Euclidean distance, inside nodes (neurons) that are arranged in the form of a grid. This creates a map of nodes where the instances that are most similar to each other are inside the same node or in neighboring nodes.

The objective of this paper is to propose FIS uses of information entropy that are measured from the data clustered in the SOM's nodes by using a PS approach. The theory is that, with the removal of the chosen instances, the method will smoothen the borders of difficult datasets such as those suffering from overlap problems by minimizing the imbalanced data [17]. Some papers use SOM to preprocess a dataset [18–20]; however, most of them are focused on the generation of another dataset represented by prototypes, which, in the literature, is cited with a deform in the border region, causing the algorithm to reduce the generalization capacity.

In addition to the proposed method, this work has as a contribution the introduction of an overlapped measure to monitor the threshold of entropy, analyzed in each region of SOM nodes to filter the majority class in a region (H-FIS (High Filter IS)), the minority class (L-FIS (Low Filter IS)), and both the classes (B-FIS). These measures were created to identify different attributes that evaluate the data complexity before classification, thus revealing different information about the data, including overlap [21]. For this, a synthetic dataset was created to control the imbalanced dataset and find the best threshold parameters from complexity measures. Finally, this approach was validated using 12 real datasets and contrasting it with 1NN with and without the FIS approach to measure the gain in the data pre-processing.

The rest of the paper is organized as follows. Section 2 provides brief explanations of PS, SOM, and data complexity measures. The methods are proposed in Section 3. The methodology is detailed in Section 4. Experimental results and discussion are presented in Section 5. The last section (Section 6) contains the conclusions.

2. Theoretical Fundamentals

2.1. Prototype Selection

Prototype Selection (PS) consists of an approach to promote a transformation in a dataset by minimizing data complexity, reducing the requirements of storage of the raw dataset, and eliminating instances of noise [14]. The process starts with a raw dataset, and an algorithm of PS is used to find the most representative instance, forming a reduced dataset.

In this regard, suppose a raw dataset is to be used as a training dataset with m attributes and n instances, i.e., $\boldsymbol{X}_{Train} = [\boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \ldots, \boldsymbol{x}_{mn}]$, and each instance has a label where $\boldsymbol{Y}_{Train} = [y_1, y_2, \ldots, y_n]$. Normally, in the kNN classification process, $\boldsymbol{X}_{Train}$ is used as a model to classify test instances. In the prototype selection, a subset of $\boldsymbol{X}_{Train}$ is selected, i.e., $\boldsymbol{X}_{PS}$, where $\boldsymbol{X}_{PS} \subseteq \boldsymbol{X}_{Train}$. Then, $\boldsymbol{X}_{PS}$ is used to classify the test instances instead of the original $\boldsymbol{X}_{Train}$.

2.2. Self-Organizing Map

Self-Organizing Map (SOM) is a neural network that organizes a dataset in a grid of neurons located on a regular low-dimensional grid, usually a two-dimensional (2D) one [16]. It is conducted by an unsupervised learning algorithm that aims to associate similar instances in the same neurons or in the adjacent neurons of the grid.

The training set $\boldsymbol{X}_{Train}$ is used to train SOM. Additionally, each neuron j of the SOM grid has a weight vector $\boldsymbol{w}_j = [w_{j1}, w_{j2}, ..., w_{jn}]^T$, where $j = 1, 2, ..., l$; here, l is the total number of neurons of SOM.

The learning process starts with a random choice of the training dataset to be compared with the weight vector of the grid that is randomly initialized. The comparison between $\boldsymbol{x}_n$ and $\boldsymbol{w}_j$ is usually made through the Euclidean distance. The shortest distance indicates the closest neuron c, which will have its weight vector $\boldsymbol{w}_c$ updated to get close to the selected instance $\boldsymbol{x}_n$. Formally, neuron c is defined in Equation (1):

$$c = argmin \parallel \boldsymbol{x}_n - \boldsymbol{w}_j \parallel \tag{1}$$

The closest weights vector $\boldsymbol{w}_c$ and their neighbors are updated using the Kohonen algorithm [16]. However, the topological neighborhood is defined so that the farther away the neuron is from $\boldsymbol{w}_c$, the lower the intensity of the neighborhood to be updated. Please see the work of Kohonen [16] for a complete explanation of the training rule of the SOM.

2.3. Data Complexity Measures

Data complexity measures can indicate properties of data that increase or reduce the level of performance expected in a process of data classification. Such measures have bee studied by different authors [21–25].

The use of these complexity measures allows a better comprehension of both data distribution in the data space and how classes are separated. In this work too, these measures are used as a tool to gain a better comprehension of the performance of the pre-processing methods in the different datasets and identify the level of data overlap that exists in a dataset.

There are several measures suggested in the relevant literature. It was decided that the works of Cano [24] and Moran [25] would be used. In their different works, these authors identified that F1 and F3 are good measures to identify overlaps for different classifiers, while N2 is a good measure to evaluate the classification performance of kNN. As the works mentioned do not used the D3 measure, we decided to add it to our work.

These measures, described in the following sections, are determined for the two-class problem. For the multi-class problem, it is possible to extend such measures [25].

The Fischer's Discriminant Ratio (F1) measure [22] calculates the separability between two classes in a determined attribute. The measure for an attribute is given in Equation (2):

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{2}$$

where μ_1 and μ_2 are the class average measures and σ_1 and σ_2 are the respective standard deviations.

In a dataset with m attributes, the measure is calculated for each of these attributes. The value considered as F1 is the greatest value of f, i.e., the attribute with the greatest separation.

Low values of F1 indicate classes with closer data centers, which, in turn, indicate data overlap.

The Maximum (Individual) Feature Efficiency (F3) measure is calculated for each attribute based on their efficacy to separate classes [21]. This measure is calculated using the maximum and minimum values of the attributes in each of the classes. The value is taken as the fraction of instances that are outside the range of the opposite class.

The value of F3 is defined as the greatest value among the attributes. The value of this measure ranges from 0 to 1, where lower values indicate greater overlap.

The Ratio of Average Intraclass/Interclass NN Distance (N2) measure is calculated as the ratio of the intraclass distance to the interclass distance of the instances. This measure shows the distributions of the classes and how close the classes are to each other [21]. This measure is given by Equation (3):

$$N2 = \frac{\sum_{i=1}^{n} Intraclass(x_i)}{\sum_{i=1}^{n} Interclass(x_i)} \tag{3}$$

where x_i represents the instances and n represents the total number of instances. Low values of N2 indicate that the classes are more separable and, thus, easier to identify.

The class density in overlap region (D3) measure was proposed by Sanchez [23], and it indicates the number of instances there are in an overlapped area.

For the present purpose, this measure uses a kNN to evaluate whether an instance disagrees with its neighbors. In the case it does, that instance is marked as an overlapped instance. The result of the measure is the fraction of instances that are marked as being overlapped when compared with the total number of instances.

To identify these instances, the value of k must be chosen for the kNN classifier. For this work, the kNN used for the measures had $k = 5$ as the original work.

3. Filtering-Based Instance Selection Algorithm

The proposed method is developed on a trained SOM. As the instances with similar attributes are mapped in the same neurons, it is expected that the instances with different classes can co-exist inside the neuron, i.e., overlapped classes.

Thus, after SOM training, a second step is introduced in this paper. This step consists of post-processing through entropy calculation for each neuron of the grid [26]. The Shannon entropy for the two probabilities is defined in Equation (4):

$$H = -p * log_2(p) - q * log_2(q) \tag{4}$$

where p is the probability of class A inside the neuron and q of class B, where $p = 1 - q$.

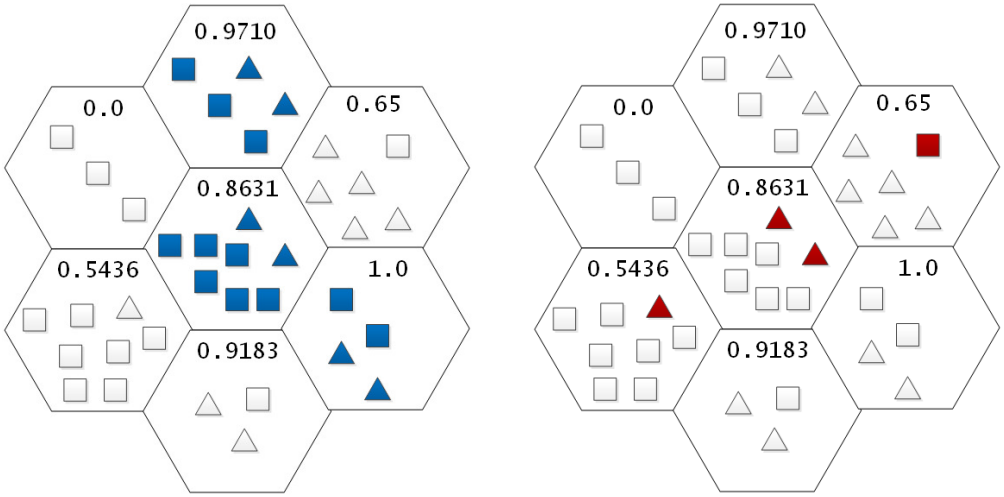
It is important to note that the entropy in the proposed method is calculated only according to the distribution inside each SOM's node. The overall distribution of the classes has no impact.

Based on the entropy value, there are three different filter methods that can be proposed:

- High Filter (H-FIS): If the neuron entropy is higher than or equal to a certain threshold (*ThresholdHigh*), the overlapped instances are removed. The premise here is that the overlapped instances do not have relevant information to delimited the borders between the classes.
- Low Filter (L-FIS): If the neuron entropy is lower than or equal to *ThresholdLow*, the instances from the class with the lower probability are removed. The premise here is that removing those instances from the class that do not agree with the majority class can smoothen the borders between the classes.
- Both (High and Low Filter) (B-FIS): H-FIS and L-FIS are combined through the comparison of *ThresholdHigh* higher than *ThresholdLow*. The premise here is that instances overlapped in the border can be removed.

After the instance filtering is done using any of the proposed approaches, the selected dataset is defined. This selected dataset ($\boldsymbol{X}_{PS}$) is used for the training of the classification algorithm.

An illustration of the FIS approach is represented in Figure 1. In this example, a hypothetical SOM trained has the instances mapped to the neurons grid. These instances belong to two classes (*A* and *B*), where class *A* is represented by squares and class *B* by triangles. The instances to be removed are highlighted in the figure.



(**a**) H-FIS (**b**) L-FIS

Figure 1. Example of the selection process to decide which instances should be removed in a hypothetical dataset after the use of SOM using H-FIS (**a**) and L-FIS (**b**). The value of the threshold hyperparameter was set at 0.8631 for both H-FIS and L-FIS. The instances selected for removal in this scenario are highlighted.

Figure 1a has the H-FIS approach parameterized with a *ThresholdHigh* of 0.8631. It is to be noted that the instances removed are in regions where the neuron entropy is equal or higher than the threshold value, and the instances from both the classes are removed. Figure 1b has the L-FIS approach parameterized with a *ThresholdLow* of 0.8631. Thus, the instances that belong to the minority (highlighted instances) class are removed.

4. Materials and Methods

Figure 2 represents the methodology followed during the experiments involving the FIS approach. In each dataset, an experimental step was considered, as explained in this section.

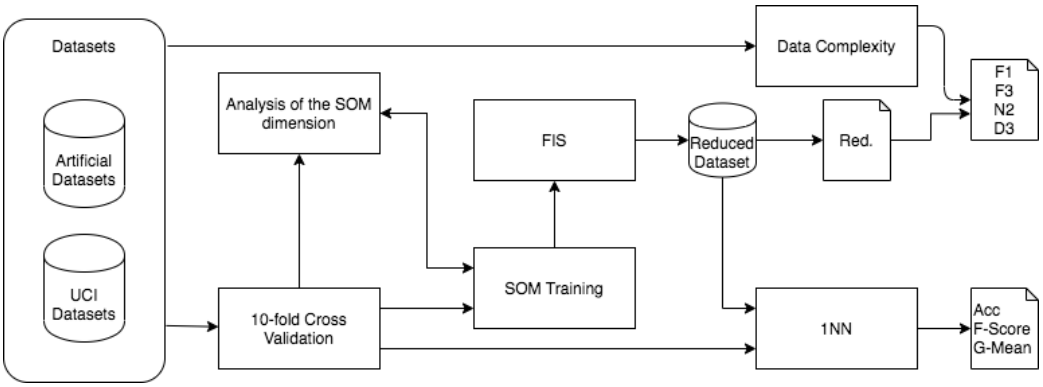


Figure 2. Schema of experimental methodology.

For the first step, a controlled environment was created to study the effectiveness of the methods in the overlapping problems and in imbalanced datasets.

This environment consists of 121 artificial datasets that represent different levels of overlapped and imbalanced classes. A sample of the dataset is presented in Figure 3.

These artificial datasets were created with 1000 instances, divided equally between two classes. The datasets have only two attributes to simplify the data visualization in 2D. The instances were generated using a random Gaussian distribution to fill the attributes' values.

To create the overlapping effect, one of the classes had the mean of the distribution fixed, while the second class had its mean changed in the several values of distance from the first class. This difference was set from 0 to 5 with steps of 0.125.

To create an imbalanced dataset, the approach taken was that of removing instances from the target class. The instance number of the positive class was chosen to generate different levels of the imbalance dataset, starting in scenarios with a low level of imbalance and moving to more severe levels. The number of instance chosen can be checked in Table 1, in which they were placed to identify the imbalance, the information of the proportion of the positive class, and the number of the negative class.

Table 1. Different levels of imbalance for the generation of artificial bases.

# Instance of Positive Class	Rate of Positive Class (%)	Imbalanced Rate
500	50%	1.00:1
409	45%	1.22:1
333	40%	1.50:1
269	35%	1.86:1
214	30%	2.34:1
167	25%	2.99:1
125	20%	4.00:1
88	15%	5.68:1
56	10%	8.93:1
26	5%	19.23:1
5	1%	100.00:1

An example of the distributions is displayed in Figure 3. In the figure, 15 of the artificial datasets created are displayed in a 2D graph. As can be noticed, low values of the difference between the classes average create a large class overlapping; as this value is increased, the data increase their separability until the classes are completely separated.

(**a**) Diff. 0.0; Class Pos. 10% (**b**) Diff. 0.0; Class Pos. 25% (**c**) Diff. 0.0; Class Pos. 50%

(**d**) Diff. 1.0; Class Pos. 10% (**e**) Diff. 1.0; Class Pos. 25% (**f**) Diff. 1.0; Class Pos. 50%

(**g**) Diff. 2.0; Class Pos. 10% (**h**) Diff. 2.0; Class Pos. 25% (**i**) Diff. 0.0; Class Pos. 50%

(**j**) Diff. 3.0; Class Pos. 10% (**k**) Diff. 3.0; Class Pos. 25% (**l**) Diff. 3.0; Class Pos. 50%

(**m**) Diff. 4.0; Class Pos. 10% (**n**) Diff. 4.0; Class Pos. 25% (**o**) Diff. 4.0; Class Pos. 50%

Figure 3. Distribution of the artificial datasets in different configurations of imbalanced rate and overlapping. The red x represents the negative class, while the blue $+$ represents the positive class.

To verify the behavior of the data outside a controlled environment, it was decided to test the methods on 12 datasets taken from the UCI repository [27]. These datasets are summarized in Table 2 and represent different data characteristics in relation to number of attributes, instances, and class balance.

Table 2. Summary of the UCI tables used in this experiment.

	Dataset	# Samples	Positive Class	% Positive Class	# Attributes
1	Ecoli	336	pp	15%	7
2	Glass	214	6	4%	9
3	Haberman	306	2	26%	3
4	Heart	303	1,2,3,4	4%	13
5	Hepatitis	155	1	21%	19
6	Iris	150	versicolor	33%	4
7	Libra	360	1,2,3	20%	90
8	Mamographic	961	Malign	46%	5
9	Pima	768	1	35%	8
10	SPECTF-Heart	268	0	21%	44
11	Wine	178	2	40%	13
12	Wisconsin	699	malign	34%	9

The methods require two groups of hyperparameters, the threshold values, and a defined SOM. It is necessary to test the methods in the selected datasets to define the different values for these hyperparameters.

For the values of the thresholds *ThresholdHigh* and *ThresholdLow*, the value of the entropy threshold was experimented from 0.0 to 1.0, with steps of 5% in the difference between the classes. This is represented in Table 3 that shows the ratio between the classes inside the node and the respective entropy value. These entropy values were used as the different hyperparameters for our tests.

Table 3. The different entropy values used in the experiment.

Ratio between the Classes Inside the Node		Calculated Entropy
Most Probable Class	Least Probable Class	
50%	50%	1.0000
55%	45%	0.9928
60%	40%	0.9710
65%	35%	0.9341
70%	30%	0.8813
75%	25%	0.8113
80%	20%	0.7219
85%	15%	0.6098
90%	10%	0.4690
95%	5%	0.2864
100%	0%	0.0000

For the SOM hyperparameters, a hexagon grid of equal sides was used. The different lengths of the grid size were calculated as defined in Equations (5) and (6), with C_{Map} having the values of $\{-2, -1, 0, 1, 2, 3, 4, 5\}$:

$$l_{SOM} = \frac{\sqrt{\#instances}}{2} + C_{Map} \tag{5}$$

$$SOMSize = (l_{SOM})^2 \tag{6}$$

The next step was to compare the classification of the different datasets with the classification of the 1NN without pre-processing to compare improvements. For the validation, the training and test datasets were separated using the 10-fold cross validation methodology.

To measure the performance, it was decided to use the accuracy, F-Score, and G-Mean to validate the impact of the methods. Both the F-Score and G-Mean are commonly used for the measurement of imbalanced data [28]. This imbalance is presented in some of the UCI datasets. The F-Score measures the effectiveness of the classifier focused on the positive class, while the G-Mean indicates that equal importance is given to both classes [15]. To calculate these measures, the confusion matrix was used to determine the classification performance in terms of the positive and negative classes.

The complexity measures F1, F3, N2, and D3 presented in Section 2.3 were used for all the UCI and artificial datasets.

5. Results

In Tables 4–7, the best values of accuracy, F-Score, and G-Mean obtained with the different approaches are summarized in terms of their means and standard deviations. These values represent the best value obtained with different values of threshold and the length of the grid size for each dataset. In Tables 4–6, some of the 41 artificial datasets were chosen to represent the data. The top classifier for a dataset is highlighted in bold.

Table 4. Results for artificial dataset with 50% of positive class.

Dataset	Method	Acc	F-Score	G-Mean
0.0/50%	1NN	0.5340 ± 0.0103	0.5296 ± 0.0124	0.5339 ± 0.0104
0.0/50%	H-FIS	0.5300 ± 0.0123	0.5297 ± 0.0062	0.5300 ± 0.0123
0.0/50%	L-FIS	**0.5344 ± 0.0105**	**0.5302 ± 0.0129**	**0.5343 ± 0.0106**
0.0/50%	B-FIS	0.5300 ± 0.0123	0.5297 ± 0.0062	0.5300 ± 0.0123
0.5/50%	1NN	0.5132 ± 0.0058	0.5104 ± 0.0064	0.5131 ± 0.0057
0.5/50%	H-FIS	**0.6026 ± 0.0171**	**0.6015 ± 0.0179**	**0.6016 ± 0.0178**
0.5/50%	L-FIS	0.5484 ± 0.0101	0.5494 ± 0.0117	0.5483 ± 0.0100
0.5/50%	B-FIS	**0.6026 ± 0.0171**	**0.6015 ± 0.0179**	**0.6016 ± 0.0178**
1.0/50%	1NN	0.6694 ± 0.0049	0.6665 ± 0.0044	0.6693 ± 0.0049
1.0/50%	H-FIS	0.7434 ± 0.0076	0.7391 ± 0.0143	0.7430 ± 0.0076
1.0/50%	L-FIS	0.7290 ± 0.0090	0.7277 ± 0.0097	0.7290 ± 0.0090
1.0/50%	B-FIS	**0.7472 ± 0.0083**	**0.7435 ± 0.0065**	**0.7468 ± 0.0082**
1.5/50%	1NN	0.7962 ± 0.0034	0.7965 ± 0.0048	0.7962 ± 0.0034
1.5/50%	H-FIS	**0.8420 ± 0.0056**	**0.8418 ± 0.0055**	**0.8420 ± 0.0056**
1.5/50%	L-FIS	0.8312 ± 0.0056	0.8329 ± 0.0061	0.8311 ± 0.0056
1.5/50%	B-FIS	**0.8420 ± 0.0056**	**0.8418 ± 0.0055**	**0.8420 ± 0.0056**
2.0/50%	1NN	0.8636 ± 0.0017	0.8637 ± 0.0017	0.8636 ± 0.0017
2.0/50%	H-FIS	0.9098 ± 0.0034	0.9091 ± 0.0038	0.9097 ± 0.0035
2.0/50%	L-FIS	0.8956 ± 0.0017	0.8943 ± 0.0026	0.8955 ± 0.0017
2.0/50%	B-FIS	**0.9104 ± 0.0029**	**0.9092 ± 0.0028**	**0.9103 ± 0.0029**
2.5/50%	1NN	0.9476 ± 0.0011	0.9476 ± 0.0012	0.9476 ± 0.0011
2.5/50%	H-FIS	0.9636 ± 0.0009	0.9636 ± 0.0009	0.9636 ± 0.0009
2.5/50%	L-FIS	0.9606 ± 0.0011	0.9607 ± 0.0012	0.9606 ± 0.0011
2.5/50%	B-FIS	**0.9638 ± 0.0022**	**0.9639 ± 0.0022**	**0.9638 ± 0.0022**
3.0/50%	1NN	0.9700 ± 0.0014	0.9700 ± 0.0014	0.9700 ± 0.0014
3.0/50%	H-FIS	0.9818 ± 0.0016	0.9818 ± 0.0017	0.9818 ± 0.0016
3.0/50%	L-FIS	0.9766 ± 0.0005	0.9766 ± 0.0006	0.9766 ± 0.0005
3.0/50%	B-FIS	**0.9820 ± 0.0016**	**0.9820 ± 0.0016**	**0.9820 ± 0.0016**
3.5/50%	1NN	0.9876 ± 0.0005	0.9876 ± 0.0005	0.9876 ± 0.0005
3.5/50%	H-FIS	0.9898 ± 0.0008	0.9898 ± 0.0008	0.9898 ± 0.0008
3.5/50%	L-FIS	0.9884 ± 0.0005	0.9884 ± 0.0006	0.9884 ± 0.0005
3.5/50%	B-FIS	**0.9900 ± 0.0014**	**0.9900 ± 0.0017**	**0.9900 ± 0.0017**
4.0/50%	1NN	0.9922 ± 0.0013	0.9922 ± 0.0013	0.9922 ± 0.0013
4.0/50%	H-FIS	**0.9954 ± 0.0017**	**0.9954 ± 0.0017**	**0.9954 ± 0.0017**
4.0/50%	L-FIS	0.9932 ± 0.0011	0.9932 ± 0.0011	0.9932 ± 0.0011
4.0/50%	B-FIS	**0.9954 ± 0.0017**	**0.9954 ± 0.0017**	**0.9954 ± 0.0017**
4.5/50%	1NN	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/50%	H-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/50%	L-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/50%	B-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/50%	1NN	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/50%	H-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/50%	L-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/50%	B-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**

Table 5. Results for artificial dataset with 25% of positive class.

Dataset	Method	Acc	F-Score	G-Mean
0.0/25%	1NN	0.6144 ± 0.0061	0.2252 ± 0.0102	0.4082 ± 0.0106
0.0/25%	H-FIS	0.7433 ± 0.0082	0.2131 ± 0.0215	0.3921 ± 0.0215
0.0/25%	L-FIS	0.6918 ± 0.0215	**0.2255 ± 0.0108**	**0.4085 ± 0.0111**
0.0/25%	B-FIS	**0.7439 ± 0.0016**	0.2131 ± 0.0215	0.3921 ± 0.0215
0.5/25%	1NN	0.6345 ± 0.0053	0.2764 ± 0.0159	0.4582 ± 0.0157
0.5/25%	H-FIS	**0.7514 ± 0.0065**	0.2838 ± 0.0093	0.4618 ± 0.0096
0.5/25%	L-FIS	0.7166 ± 0.0045	**0.2968 ± 0.0193**	0.4631 ± 0.0171
0.5/25%	B-FIS	0.7499 ± 0.0070	0.2928 ± 0.0324	**0.4631 ± 0.0101**
1.0/25%	1NN	0.7379 ± 0.0086	0.4859 ± 0.0115	0.6365 ± 0.0088
1.0/25%	H-FIS	0.8033 ± 0.0143	0.5524 ± 0.0212	0.6647 ± 0.0212
1.0/25%	L-FIS	0.7934 ± 0.0058	0.5435 ± 0.0209	0.6630 ± 0.0173
1.0/25%	B-FIS	**0.8075 ± 0.0090**	**0.5591 ± 0.0165**	**0.6707 ± 0.0167**
1.5/25%	1NN	0.8156 ± 0.0075	0.6192 ± 0.0128	0.7292 ± 0.0091
1.5/25%	H-FIS	**0.8645 ± 0.0057**	**0.7097 ± 0.0101**	0.7866 ± 0.0113
1.5/25%	L-FIS	0.8522 ± 0.0027	0.6862 ± 0.0069	0.7711 ± 0.0058
1.5/25%	B-FIS	0.8642 ± 0.0041	**0.7097 ± 0.0101**	**0.7933 ± 0.0077**
2.0/25%	1NN	0.8786 ± 0.0070	0.7565 ± 0.0121	0.8327 ± 0.0070
2.0/25%	H-FIS	**0.9241 ± 0.0035**	**0.8456 ± 0.0132**	**0.8905 ± 0.0097**
2.0/25%	L-FIS	0.9163 ± 0.0022	0.8266 ± 0.0042	0.8727 ± 0.0034
2.0/25%	B-FIS	**0.9241 ± 0.0035**	**0.8456 ± 0.0132**	**0.8905 ± 0.0097**
2.5/25%	1NN	0.9490 ± 0.0053	0.8984 ± 0.0106	0.9323 ± 0.0075
2.5/25%	H-FIS	0.9601 ± 0.0023	0.9204 ± 0.0051	0.9467 ± 0.0059
2.5/25%	L-FIS	0.9589 ± 0.0017	0.9175 ± 0.0040	0.9430 ± 0.0028
2.5/25%	B-FIS	**0.9613 ± 0.0027**	**0.9223 ± 0.0059**	**0.9467 ± 0.0026**
3.0/25%	1NN	0.9742 ± 0.0007	0.9486 ± 0.0012	0.9663 ± 0.0009
3.0/25%	H-FIS	0.9802 ± 0.0020	0.9603 ± 0.0040	0.9725 ± 0.0041
3.0/25%	L-FIS	0.9808 ± 0.0027	0.9617 ± 0.0054	**0.9744 ± 0.0039**
3.0/25%	B-FIS	**0.9811 ± 0.0023**	**0.9622 ± 0.0046**	0.9741 ± 0.0044
3.5/25%	1NN	0.9928 ± 0.0013	0.9856 ± 0.0025	0.9908 ± 0.0016
3.5/25%	H-FIS	0.9940 ± 0.0011	0.9880 ± 0.0021	0.9924 ± 0.0015
3.5/25%	L-FIS	0.9934 ± 0.0017	0.9868 ± 0.0034	0.9916 ± 0.0025
3.5/25%	B-FIS	**0.9943 ± 0.0007**	**0.9886 ± 0.0013**	**0.9930 ± 0.0014**
4.0/25%	1NN	0.9970 ± 0.0000	0.9940 ± 0.0000	0.9960 ± 0.0000
4.0/25%	H-FIS	**0.9979 ± 0.0008**	**0.9958 ± 0.0016**	**0.9978 ± 0.0016**
4.0/25%	L-FIS	0.9973 ± 0.0007	0.9946 ± 0.0013	0.9966 ± 0.0013
4.0/25%	B-FIS	**0.9979 ± 0.0008**	**0.9958 ± 0.0016**	**0.9978 ± 0.0016**
4.5/25%	1NN	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/25%	H-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/25%	L-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/25%	B-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/25%	1NN	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/25%	H-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/25%	L-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/25%	B-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**

Analyzing the results of artificial dataset, for the case of accuracy, which can be seen in Tables 4–6, the method proposed (FIS) resulted in significantly greater gains in the bases with high overlap, especially when there was a greater imbalance. A possible explanation for this good performance in the cases of high imbalance could be that given that the parameters were optimized for accuracy, a considerable portion of the positive class has been removed by this method, which favors the negative class, which, being in the majority, increases the accuracy value. This effect decreases as the base becomes more balanced and the gains become smaller. However, the effect does exist, including for the balanced bases.

Table 6. Results for artificial dataset with 10% of positive class.

Dataset	Method	Acc	F-Score	G-Mean
0.0/10%	1NN	0.8122 ± 0.0037	0.1029 ± 0.0203	0.3076 ± 0.0333
0.0/10%	H-FIS	**0.8982 ± 0.0016**	0.1234 ± 0.0295	0.3190 ± 0.0531
0.0/10%	L-FIS	0.8842 ± 0.0043	0.1070 ± 0.0209	0.3091 ± 0.0334
0.0/10%	B-FIS	0.8971 ± 0.0030	**0.1249 ± 0.0287**	**0.3190 ± 0.0530**
0.5/10%	1NN	0.8004 ± 0.0071	0.0947 ± 0.0247	0.2997 ± 0.0390
0.5/10%	H-FIS	**0.8978 ± 0.0023**	0.0944 ± 0.0168	0.2927 ± 0.0260
0.5/10%	L-FIS	0.8813 ± 0.0086	**0.1140 ± 0.0351**	**0.3042 ± 0.0397**
0.5/10%	B-FIS	0.8957 ± 0.0018	0.1048 ± 0.0285	0.2937 ± 0.0262
1.0/10%	1NN	0.8741 ± 0.0075	0.3967 ± 0.0304	0.6164 ± 0.0247
1.0/10%	H-FIS	0.9040 ± 0.0037	0.3960 ± 0.0398	0.6046 ± 0.0183
1.0/10%	L-FIS	0.8957 ± 0.0042	0.4211 ± 0.0271	**0.6166 ± 0.0247**
1.0/10%	B-FIS	**0.9061 ± 0.0039**	**0.4218 ± 0.0344**	0.6048 ± 0.0185
1.5/10%	1NN	0.8759 ± 0.0042	0.3937 ± 0.0193	0.6095 ± 0.0160
1.5/10%	H-FIS	0.9014 ± 0.0059	0.4290 ± 0.0562	0.6099 ± 0.0245
1.5/10%	L-FIS	0.9000 ± 0.0033	0.4326 ± 0.0176	**0.6187 ± 0.0231**
1.5/10%	B-FIS	**0.9065 ± 0.0087**	**0.4399 ± 0.0262**	0.6099 ± 0.0245
2.0/10%	1NN	0.9381 ± 0.0010	0.6982 ± 0.0063	0.8275 ± 0.0081
2.0/10%	H-FIS	0.9561 ± 0.0049	0.7557 ± 0.0301	0.8356 ± 0.0130
2.0/10%	L-FIS	0.9554 ± 0.0035	0.7633 ± 0.0190	0.8429 ± 0.0166
2.0/10%	B-FIS	**0.9583 ± 0.0063**	**0.7759 ± 0.0354**	**0.8483 ± 0.0151**
2.5/10%	1NN	0.9737 ± 0.0021	0.8727 ± 0.0087	0.9367 ± 0.0011
2.5/10%	H-FIS	0.9802 ± 0.0013	0.9006 ± 0.0058	0.9385 ± 0.0039
2.5/10%	L-FIS	**0.9817 ± 0.0015**	**0.9071 ± 0.0074**	0.9413 ± 0.0047
2.5/10%	B-FIS	0.9813 ± 0.0016	0.9055 ± 0.0077	**0.9415 ± 0.0040**
3.0/10%	1NN	0.9799 ± 0.0008	0.9007 ± 0.0044	0.9467 ± 0.0042
3.0/10%	H-FIS	**0.9881 ± 0.0010**	**0.9405 ± 0.0049**	**0.9628 ± 0.0040**
3.0/10%	L-FIS	0.9853 ± 0.0008	0.9254 ± 0.0070	0.9507 ± 0.0014
3.0/10%	B-FIS	**0.9881 ± 0.0010**	**0.9405 ± 0.0049**	**0.9628 ± 0.0040**
3.5/10%	1NN	0.9910 ± 0.0013	0.9549 ± 0.0067	0.9709 ± 0.0065
3.5/10%	H-FIS	**0.9942 ± 0.0023**	0.9714 ± 0.0120	0.9840 ± 0.0119
3.5/10%	L-FIS	0.9928 ± 0.0013	0.9643 ± 0.0065	0.9800 ± 0.0064
3.5/10%	B-FIS	**0.9942 ± 0.0015**	**0.9714 ± 0.0075**	**0.9840 ± 0.0051**
4.0/10%	1NN	0.9978 ± 0.0008	0.9893 ± 0.0040	0.9972 ± 0.0040
4.0/10%	H-FIS	**0.9982 ± 0.0000**	**0.9912 ± 0.0000**	**0.9990 ± 0.0000**
4.0/10%	L-FIS	0.9978 ± 0.0008	0.9893 ± 0.0040	0.9972 ± 0.0040
4.0/10%	B-FIS	**0.9982 ± 0.0000**	**0.9912 ± 0.0000**	**0.9990 ± 0.0000**
4.5/10%	1NN	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/10%	H-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/10%	L-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
4.5/10%	B-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/10%	1NN	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/10%	H-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/10%	L-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
5.0/10%	B-FIS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**

With regard to the overlap, the profit margins are higher in the bases that have more overlap, and they decrease as the level of overlap decreases to the point that we no longer have any gains with these methods. This is so because the classes are already sufficiently separated on these bases, creating conditions for the 1NN classifier without processing to have high performance.

To continue the analysis of the artificial bases, different graphs were generated with the best results of *G-Mean* after the variation of the parameters of *ThresholdHigh* and *ThresholdLow* and C_{Map}. These values were compared with the result of the 1NN classifier without pre-processing (baseline). To demonstrate the overlap, the value of the difference between the class means on the artificial base was placed on the *x*-axis, with values close to zero having high overlap and values more distant from zero having less overlap. The graphs were also divided according to their imbalance, choosing some fundamental values. In this way, it is possible to observe the impacts of overlap and imbalance at the same time

and to make comparisons between the three methods that were generated and the 1NN without any alteration. B-FIS had results similar to those of H-FIS as shown in Tables 4–6. This fact makes the gain curves of the two methods overlap. Thus, it was decided to hide the B-FIS curve in these graphics for better visualization.

From the results in Figure 4, it can be seen that the H-FIS and L-FIS methods showed the greatest gain in an intermediate range of the overlap level—between approximately 1.0 and 3.0 difference of mean. By analyzing this imbalance, it can be seen that the methods have greater benefits for an intermediate range of imbalance between 15% and 30%. However, in situations with more severe imbalance, such as 5% and 10%, these methods had lower gains compared to 1NN classifier without processing.

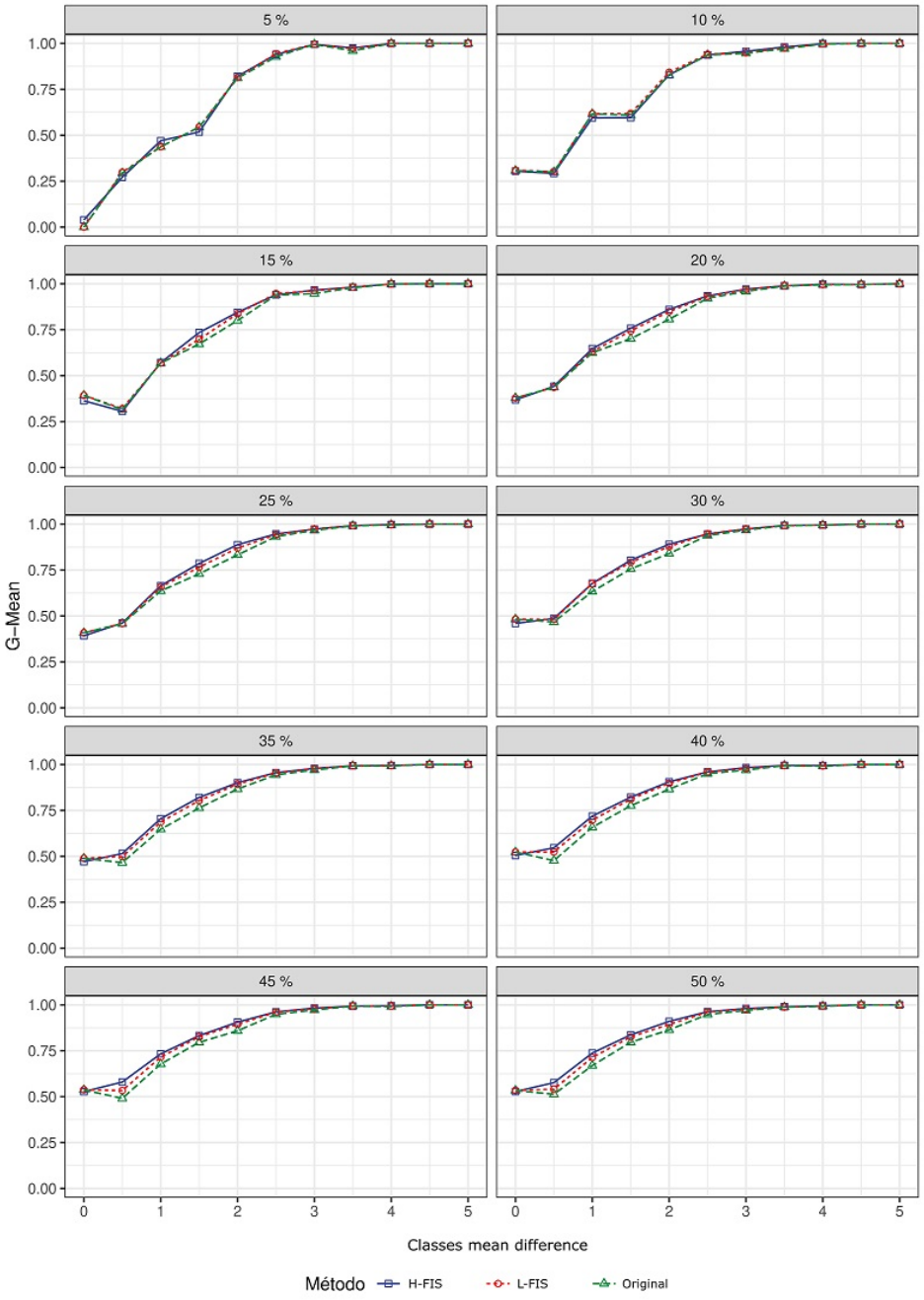


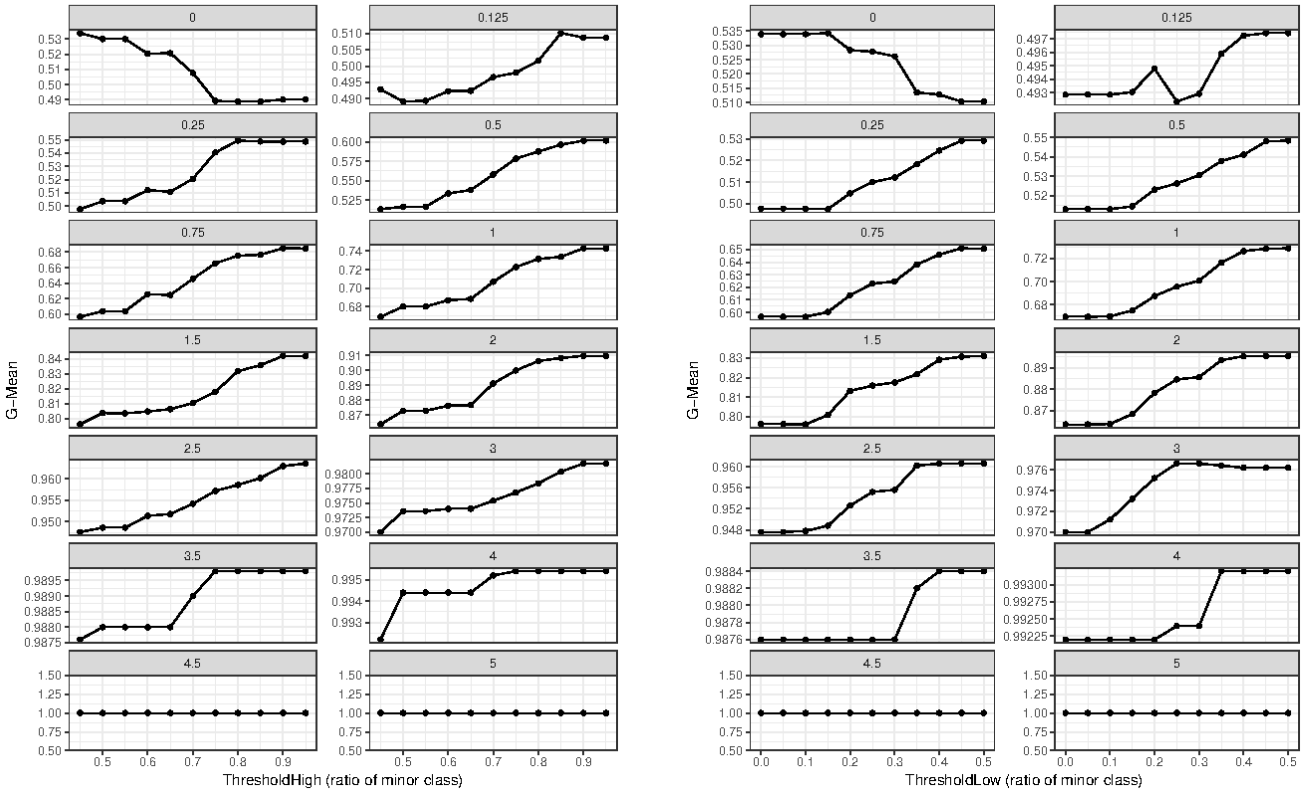
Figure 4. Comparison of *G-Mean* values on artificial bases for the methods developed in this work and 1NN without pre-processing (baseline). The grouping was done by proportion of the positive class.

From the experiments summarized in Tables 4, 6 and 7, all three methods showed an increase in accuracy, F-Score, and G-Mean when compared with the 1NN method on all datasets.

Table 7. Results for the UCI datasets.

Dataset	Method	Acc	F-Score	G-Mean
Ecoli	1NN	0.9393 ± 0.0045	0.8082 ± 0.0152	0.8908 ± 0.0127
	H-FIS	0.9613 ± 0.0030	0.8739 ± 0.0085	0.9204 ± 0.0017
	L-FIS	0.9643 ± 0.0042	0.8837 ± 0.0134	0.9271 ± 0.0092
	B-FIS	**0.9649 ± 0.0053**	**0.8855 ± 0.0168**	**0.9275 ± 0.0099**
Glass	1NN	0.9720 ± 0.0000	0.6667 ± 0.0000	0.8105 ± 0.0000
	H-FIS	0.9748 ± 0.0026	0.6824 ± 0.0215	0.7979 ± 0.0304
	L-FIS	**0.9804 ± 0.0051**	**0.7480 ± 0.0555**	**0.8268 ± 0.0297**
	B-FIS	0.9757 ± 0.0039	0.6824 ± 0.0215	0.7979 ± 0.0304
Haberman	1NN	0.6634 ± 0.0139	0.3324 ± 0.0320	0.4996 ± 0.0291
	H-FIS	0.7255 ± 0.0276	**0.3477 ± 0.0304**	**0.5078 ± 0.0264**
	L-FIS	0.7196 ± 0.0152	0.3359 ± 0.0315	0.5018 ± 0.0269
	B-FIS	**0.7366 ± 0.0132**	0.3458 ± 0.0287	0.5060 ± 0.0257
Heart	1NN	0.7617 ± 0.0108	0.7386 ± 0.0120	0.7591 ± 0.0109
	H-FIS	**0.8185 ± 0.0109**	**0.8018 ± 0.0088**	**0.8167 ± 0.0094**
	L-FIS	0.7947 ± 0.0180	0.7699 ± 0.0182	0.7900 ± 0.0172
	B-FIS	**0.8185 ± 0.0109**	**0.8018 ± 0.0088**	**0.8167 ± 0.0094**
Hepatitis	1NN	0.8013 ± 0.0161	0.5067 ± 0.0296	0.6595 ± 0.0205
	H-FIS	0.8413 ± 0.0149	0.5821 ± 0.0438	0.7022 ± 0.0384
	L-FIS	**0.8477 ± 0.0126**	**0.6096 ± 0.0369**	**0.7291 ± 0.0279**
	B-FIS	0.8465 ± 0.0029	0.5965 ± 0.0060	0.7173 ± 0.0263
Iris	1NN	0.9547 ± 0.0030	0.9659 ± 0.0023	0.9509 ± 0.0022
	H-FIS	**0.9587 ± 0.0056**	0.9688 ± 0.0042	**0.9570 ± 0.0067**
	L-FIS	0.9573 ± 0.0037	0.9678 ± 0.0027	0.9550 ± 0.0050
	B-FIS	**0.9587 ± 0.0119**	**0.9689 ± 0.0089**	**0.9570 ± 0.0067**
Libra	1NN	0.9911 ± 0.0012	0.9773 ± 0.0033	0.9775 ± 0.0032
	H-FIS	**0.9917 ± 0.0020**	**0.9787 ± 0.0051**	**0.9789 ± 0.0050**
	L-FIS	0.9911 ± 0.0012	0.9773 ± 0.0033	0.9775 ± 0.0032
	B-FIS	**0.9917 ± 0.0020**	**0.9787 ± 0.0051**	**0.9789 ± 0.0050**
Mamographic	1NN	0.7536 ± 0.0073	0.7307 ± 0.0069	0.7508 ± 0.0069
	H-FIS	0.7879 ± 0.0092	0.7766 ± 0.0105	0.7879 ± 0.0099
	L-FIS	0.7869 ± 0.0052	0.7762 ± 0.0050	0.7876 ± 0.0051
	B-FIS	**0.7983 ± 0.0060**	**0.7866 ± 0.0064**	**0.7986 ± 0.0060**
Pima	1NN	0.7109 ± 0.0080	0.5694 ± 0.0107	0.6613 ± 0.0085
	H-FIS	0.7362 ± 0.0117	0.5900 ± 0.0124	0.6761 ± 0.0102
	L-FIS	0.7411 ± 0.0053	0.6036 ± 0.0086	0.6870 ± 0.0069
	B-FIS	**0.7500 ± 0.0038**	**0.6129 ± 0.0080**	**0.6935 ± 0.0066**
SPECTF-Heart	1NN	0.6933 ± 0.0061	0.3507 ± 0.0098	0.5530 ± 0.0096
	H-FIS	**0.7925 ± 0.0057**	0.3767 ± 0.0430	0.5703 ± 0.0398
	L-FIS	0.7343 ± 0.0166	0.3753 ± 0.0315	0.5644 ± 0.0364
	B-FIS	**0.7925 ± 0.0057**	**0.3850 ± 0.0493**	**0.5777 ± 0.0459**
Wine	1NN	0.9506 ± 0.0025	0.9339 ± 0.0036	0.9360 ± 0.0034
	H-FIS	0.9551 ± 0.0000	0.9406 ± 0.0092	0.9437 ± 0.0121
	L-FIS	0.9528 ± 0.0031	0.9371 ± 0.0044	0.9390 ± 0.0041
	B-FIS	**0.9562 ± 0.0025**	**0.9420 ± 0.0034**	**0.9441 ± 0.0033**
Wisconsin	1NN	0.9577 ± 0.0033	0.9381 ± 0.0050	0.9509 ± 0.0045
	H-FIS	0.9694 ± 0.0030	0.9559 ± 0.0043	0.9678 ± 0.0038
	L-FIS	0.9677 ± 0.0048	0.9534 ± 0.0070	0.9657 ± 0.0056
	B-FIS	**0.9714 ± 0.0027**	**0.9589 ± 0.0039**	**0.9709 ± 0.0035**

The use of the *ThresholdHigh* and *ThresholdLow* have the disadvantage of having an additional hyperparameter for the classifier. To simplify this fact, we examined the behavior of the thresholds in H-FIS and L-FIS and the effect on the G-Mean in the artificial dataset. This can be seen in Figure 5.



(**a**) H-FIS (**b**) L-FIS

Figure 5. Graphs that represent the effect of the H-FIS (**a**) and L-FIS (**b**) on G-Mean in the artificial datasets as we adjust the process to be more aggressive in the removal of points. The point to the left of the graph represents the initial point without pre-processing.

From the artificial datasets, the analyses of the threshold graphs indicate that the best results are in areas where the threshold is more aggressive, especially in areas with high class overlapping. The exception happens in the dataset with the largest overlapping where the difference between the classes averages is zero. It is also possible to verify that, when the classes are far apart, the method brought no benefits as there is no class overlapping.

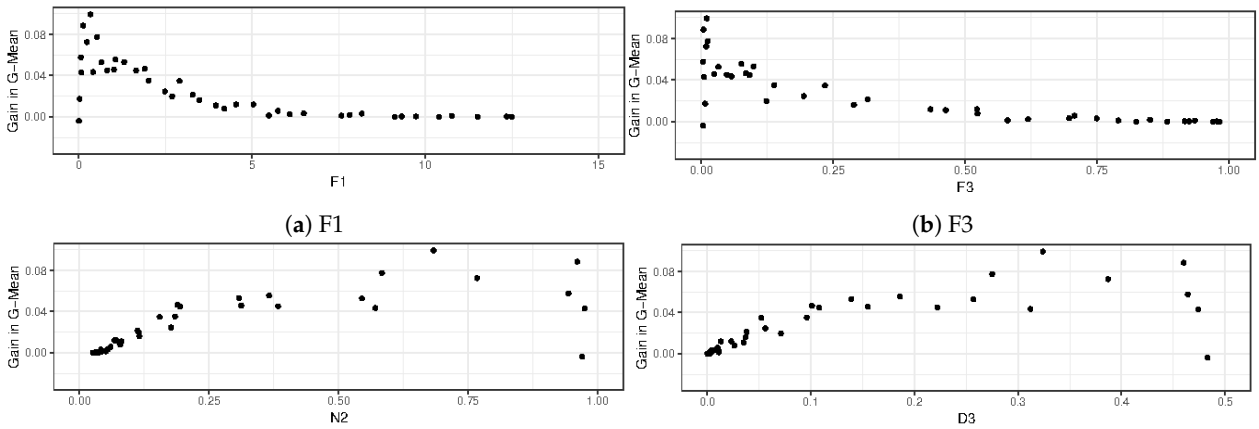
For the next step, we calculated the complexity measures for the artificial datasets. The results for some of the datasets are summarized in Table 8.

It is possible to use the data complexity values to evaluate the gain of the datasets compared to the complexity measures' values. This way, it is possible to verify the behavior of the methods according to the increase of data complexity. In Figure 6, the results are shown for B-FIS. The behavior is similar for H-FIS and L-FIS.

As can be noticed in the graphs in Figure 6, this method has a range of class overlapping with the highest improvements, thus showing that this method is efficient in the original proposal of datasets with overlapping classes. However, in datasets with severe overlapping, indicated by very low values of F1, this method had only slight improvements in 1NN. The gain of the method reduces as the overlapping decreases, until the method no longer has a gain for the original 1NN.

Table 8. Data complexity measures for the artificial datasets.

Difference in Class Average	F1	F3	N2	D3
0	0.0016	0.0030	0.9696	0.4830
0.25	0.0586	0.0030	0.9434	0.4640
0.5	0.1225	0.0040	0.9604	0.4600
0.75	0.3345	0.0100	0.6831	0.3240
1	0.5231	0.0120	0.5835	0.2750
1.5	1.0130	0.0240	0.3121	0.1550
2	1.8996	0.0840	0.1886	0.1010
2.5	3.4796	0.2890	0.1155	0.0370
3	4.5360	0.4340	0.0712	0.0230
3.5	6.0856	0.6190	0.0521	0.0110
4	8.1638	0.7490	0.0413	0.0040
4.5	10.3897	0.8830	0.0303	0.0010
5	12.4878	0.9700	0.0261	0.0000

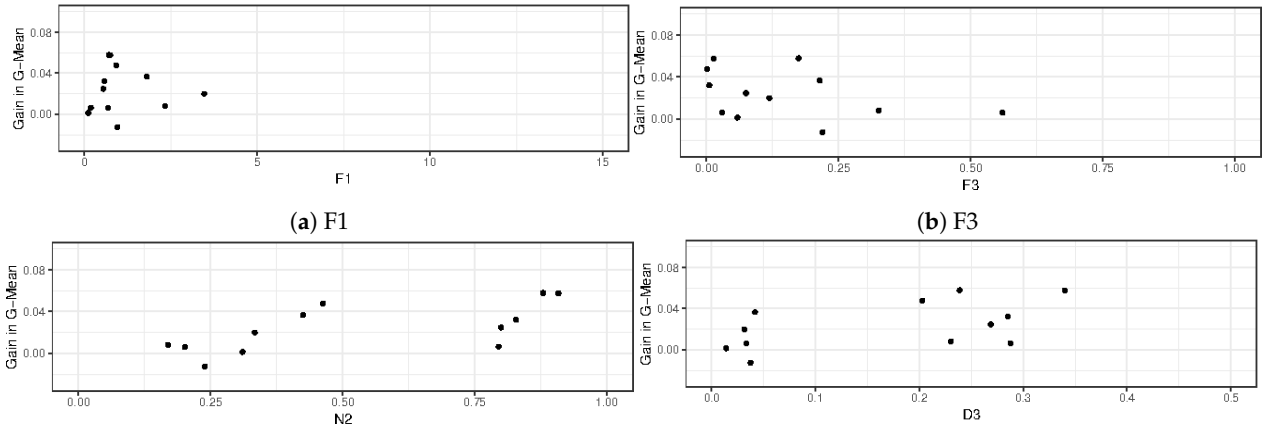


(**a**) F1 (**b**) F3

(**c**) N2 (**d**) D3

Figure 6. Graphs that represent the gain of G-Mean using the B-FIS in the artificial datasets by the different complexity measures. In the case of F1 and F3, low values represent high class overlapping, while, for N2 and D3, low values represent low class overlapping.

To check if these conclusions are the same for other distributions of data, these procedures were repeated and compared for the UCI datasets. The data complexity measures that were calculated for the datasets are presented in Table 9 and Figure 7.



(**a**) F1 (**b**) F3

(**c**) N2 (**d**) D3

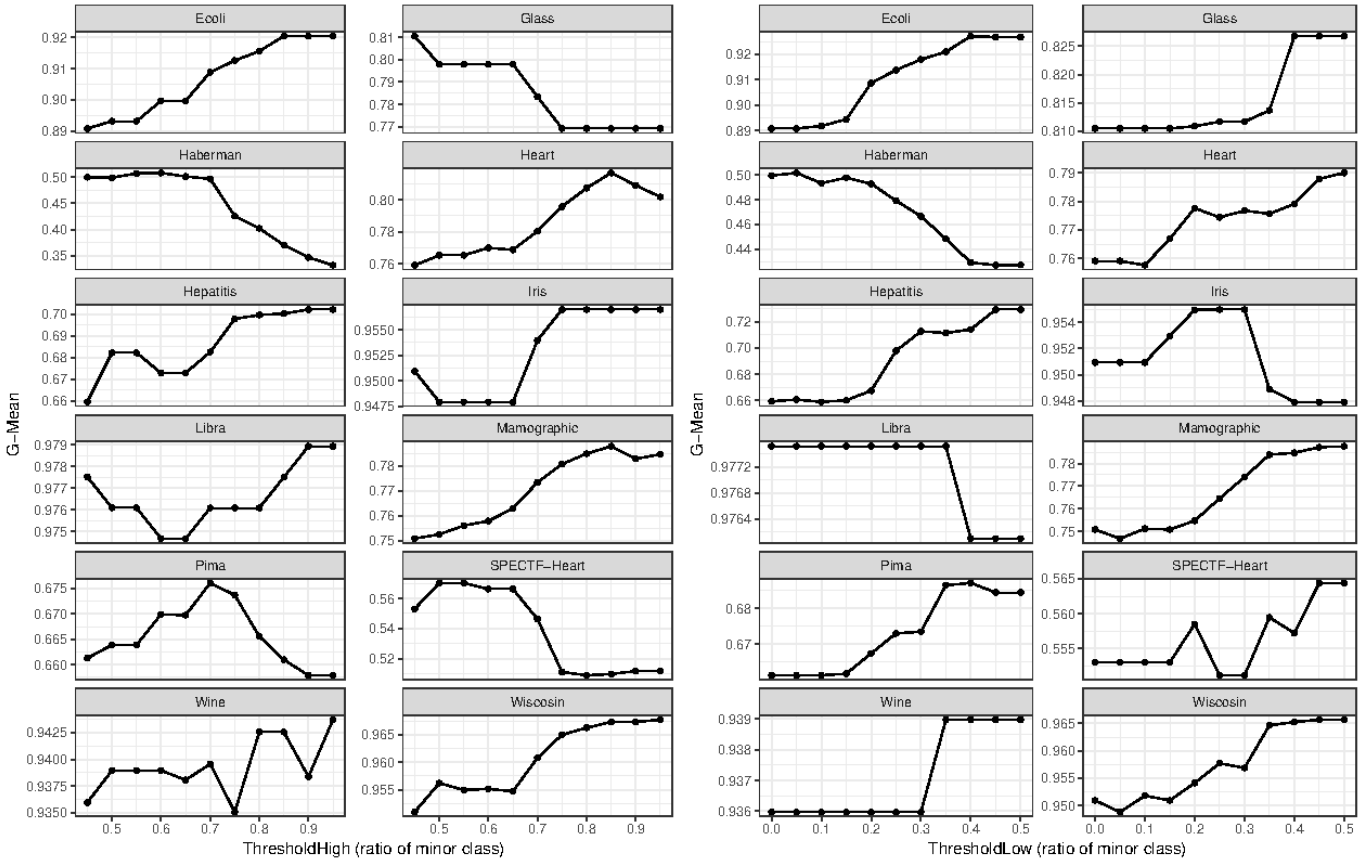
Figure 7. Graphs that represent the gain of G-Mean using the B-FIS in the UCI datasets by the different complexity measures. In the case of F1 and F3, low values represent high class overlapping, while, for N2 and D3, low values represent low class overlapping.

As can be evaluated from the results in Figure 7, the gains of the methods cannot be explained by only the complexity measures. This behavior indicates that other characteristics of the data such as data distribution and data imbalance may impact the gains of B-FIS.

The analyses of the impact of the threshold values on the UCI datasets is shown in Figure 8. For L-FIS, there is a similar behavior of the artificial datasets, where the change in behavior happens in cases where a dataset suffers from a more severe overlapping. This is indicated by the low values of F1 and F3 in the cases of the datasets *Haberman* and *Libra*. However, in the case of H-FIS for the UCI datasets, there was no clear pattern even when compared with the data complexity measures.

Table 9. Data complexity measures for the real datasets.

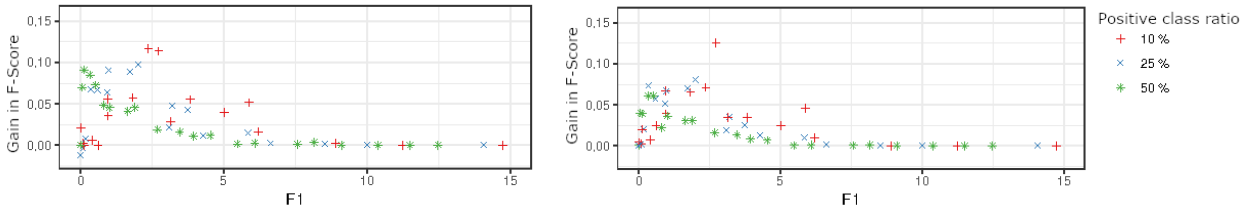
	Dataset	F1	F3	N2	D3
1	Ecoli	1.8042	0.2143	0.4247	0.0417
2	Glass	0.9531	0.2196	0.2392	0.0374
3	Haberman	0.1832	0.0294	0.7948	0.2876
4	Heart	0.7422	0.0132	0.9080	0.3399
5	Hepatitis	0.7075	0.1742	0.8789	0.2387
6	Iris	0.6802	0.5600	0.2010	0.0333
7	Libra	0.1102	0.0583	0.3102	0.0139
8	Mammographic	0.9175	0.0010	0.4624	0.2029
9	Pima	0.5743	0.0052	0.8277	0.2852
10	SPECTF-Heart	0.5443	0.0746	0.7994	0.2687
11	Wine	2.3331	0.3258	0.1692	0.2303
12	Wisconsin	3.4635	0.1187	0.3334	0.0315



(**a**) H-FIS (**b**) L-FIS

Figure 8. Graphs that represent the effect of H-FIS (**a**) and L-FIS (**b**) on G-Mean in the UCI datasets as we adjust the process to be more aggressive in the removal of the points. The point to the left of the graph represents the initial point without pre-processing.

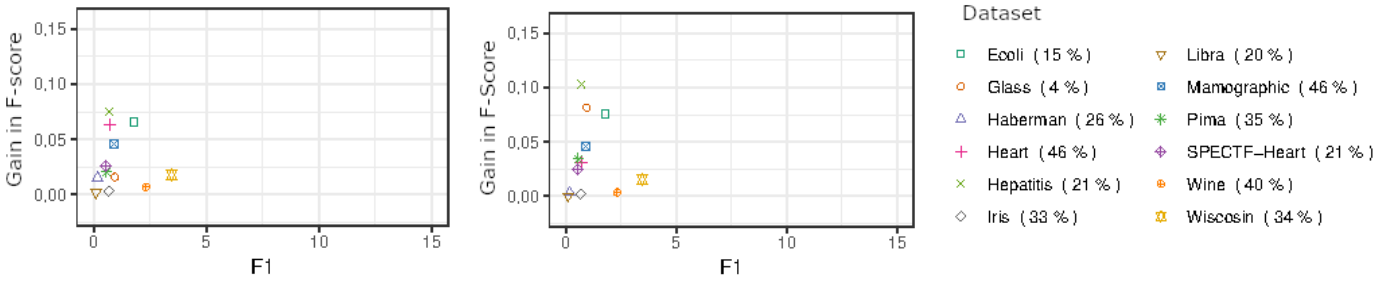
To better understand the effects of imbalance in the method, the effects of F1 were plotted against the gains of the F-Score dividing the classes' imbalance ratio. The data can be found in Figures 9 and 10.



(**a**) H-FIS (**b**) L-FIS

Figure 9. Graphs that represent the effect of the H-FIS (**a**) and L-FIS (**b**) on F-Score in the artificial datasets segregated by imbalance.

From the results in Figure 9, it is possible to identify that higher levels of data imbalance caused higher gains with a certain range of low F1 values where this effect was enhanced.



(**a**) H-FIS (**b**) L-FIS

Figure 10. Graphs that represent the effect of H-FIS (**a**) and L-FIS (**b**) on F-Score in the artificial datasets segregated by imbalance. The positive class ratio of that dataset is mentioned in parentheses.

Similarly, in the UCI datasets in Figure 10, the gains showed a similar pattern in intermediary ranges of imbalance and overlapping, especially in the case of L-FIS. The higher gains happened in datasets with more severe data imbalance.

To find the best size of SOM, the average maximum gain in F-Score was calculated for each value of C_{Map} with the help of Equation (5) for each dataset. The data were then ranked according to the F-Score value. An example of this process is summarized in Table 10 for the Ecoli dataset. The measure of the ranks in the datasets was calculated for each value of C_{Map}, and the average is summarized in Table 11. The best value is highlighted.

Table 10. Ranked values of *F-Score* for C_{Map} for the *Ecoli* dataset.

	Max *F-Score* Value		
C_{Mapa}	**H-FIS**	**L-FIS**	**B-FIS**
−2	0.8414 (8th)	0.8770 (1st)	0.8802 (1st)
−1	0.8618 (3rd)	0.8690 (2nd)	0.8740 (3rd)
0	0.8707 (1st)	0.8636 (3rd)	0.8786 (2nd)
1	0.8659 (2nd)	0.8589 (4th)	0.8721 (4th)
2	0.8562 (5th)	0.8360 (5th)	0.8622 (5th)
3	0.8584 (4th)	0.8328 (6th)	0.8597 (6th)
4	0.8536 (7th)	0.8201 (8th)	0.8590 (7th)
5	0.8551 (6th)	0.8230 (7th)	0.8563 (8th)

Table 11. Average rank of best *F-Score* for C_{Map} constant.

	Average Rank of Best *F-Score*		
C_{Mapa}	**H-FIS**	**L-FIS**	**B-FIS**
−2	4.3333	3.1667	**3.0000**
−1	4.1667	**3.0000**	3.5000
0	**3.5833**	**3.0000**	3.2500
1	4.8333	3.1667	4.3333
2	4.8333	4.0833	4.3333
3	4.2500	4.8333	5.0833
4	4.1667	5.9167	5.4167
5	5.7500	6.0833	7.0000

The best values, shown by the lowest average rank value, indicate that the best values for H-FIS and L-FIS occur in values close to the value zero; however, for B-FIS, the lowest values of C_{Map} are more effective.

It was decided to use the Wilcoxon signed-ranks test to validate whether the methods that were developed are statistically better than 1NN with no pre-processing. This test is appropriate for comparisons of classifiers over datasets [29].

The results for the UCI datasets are shown in Table 12. The results prove that the methods are statistically different with an $\alpha < 0.001$, proving statistically that the methods enhance the classification of 1NN for the three different measures.

Table 12. Calculated values of p using the Wilcoxon signed-ranks test for the UCI datasets.

Method Comparison	p **Acc**	p **F-Score**	p **G-Mean**
H-FIS-1NN	4.88×10^{-4}	4.88×10^{-3}	4.88×10^{-3}
L-FIS-1NN	3.86×10^{-3}	3.86×10^{-3}	3.86×10^{-3}
B-FIS-1NN	4.88×10^{-4}	4.88×10^{-3}	4.88×10^{-3}

Finally, a rough analysis carried out with the results available in the relevant literature is summarized in Table 13. As there is no structured dataset in the literature with separated training and test sets and a definition of the positive class in the dataset to investigate the algorithm performance in the overlap problem and imbalanced dataset, the comparison illustrated in this table is as shown in the papers cited. It may be noted that different algorithms show the best performance of G-Mean. The method proposed had good results which, in fact, were among the best results in the analyzed datasets.

Table 13. Comparative results of G-Mean for the UCI datasets used in the literature.

Dataset	**FIS**	**EVINCI** [30]	**DBANN** [4]	**NB-BASIC** [1]	**AdaOBU** [31]	**BoosOBU** [31]	**CDSMOTE** [2]	**SBagging** [32]
Ecoli	0.9275	0.7939	0.9798	0.9182	**1.000**	0.8944	0.93	–
Glass	0.8268	0.6621	0.7972	0.6776	0.7325	0.8105	**0.94**	–
Pima	0.6935	–	–	0.475	–	–	**0.728**	
Wine	**0.9562**	–	–	–	–	–	–	0.94
Wisconsin	0.9714	–	**0.9839**	0.9712	–	–	0.96	–

6. Discussion

This work introduces three prototype selection methods that use SOMs and entropy to act as a filter for the selection of prototypes. Experiments using the methods were done both in a controlled environment to simulate class overlapping and with UCI datasets. These experiments showed that the methods improved the accuracy, F-Score, and G-mean values when compared with the common 1NN classifier in the different datasets.

The use of data complexity measures allowed quantifying the class overlapping in artificial datasets. These measures indicated that the methods had increased gains in overlapping scenarios in the artificial datasets according to the original proposal of the methods.

The methods have the disadvantage of requiring the setting of new hyperparameters *ThresholdHigh* and *ThresholdLow* for the classification process. To simplify this fact, we examined the behavior of the thresholds in H-FIS and L-FIS and the impact on method improvements. This can be seen in Figures 5 and 8. L-FIS indicated a pattern of increased effectiveness with an increase in the value of *ThresholdLow* except for heavily overlapped datasets, which allows us to determine an ideal range based on an overlap measure, such as F1. H-FIS, however, did not show a clear pattern for *ThresholdHigh*.

The complexity measures allowed, by the analysis of F1, F3, and $D3_{Pos}$, indicating that these methods had good performances in databases with data overlapping. In addition, these methods showed increased gains in datasets with higher imbalance. However, the behavior of these methods could not be explained by only those measures, so other factors must affect the performance of the methods presented.

It would be interesting if further studies focus on trying some new situations, for instance, how these methods behave with noisy data and with different controlled environments involving imbalance and types of data distribution so that we can have a better comprehension of these methods. In addition, an extensive comparison with other algorithms can be done to have a better understanding of the benefits of the proposed methods.

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

Funding: This work was partially supported by CAPES (Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior) with process number 88887.199212/2018-00 and Mackenzie (Universidade Presbiteriana Mackenzie).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The public datasets used in this work can be found in the UCI Machine Learning Repository https://archive.ics.uci.edu/ml/index.php (accessed on 9 June 2021).

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

References

1. Vuttipittayamongkol, P.; Elyan, E. Improved overlap-based undersampling for imbalanced dataset classification with application to epilepsy and parkinson's disease. *Int. J. Neural Syst.* **2020**, *30*, 2050043. [CrossRef]
2. Elyan, E.; Moreno-Garcia, C.F.; Jayne, C. CDSMOTE: Class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. *Neural Comput. Appl.* **2021**, *33*, 2839–2851. [CrossRef]
3. Le, T.; Baik, S.W. A robust framework for self-care problem identification for children with disability. *Symmetry* **2019**, *11*, 89. [CrossRef]
4. Yuan, B.W.; Luo, X.G.; Zhang, Z.L.; Yu, Y.; Huo, H.W.; Johannes, T.; Zou, X.D. A novel density-based adaptive k nearest neighbor method for dealing with overlapping problem in imbalanced datasets. *Neural Comput. Appl.* **2020**, *33* , 1–25. [CrossRef]
5. Prati, R.C.; Batista, G.E.A.P.A.; Monard, M.C. Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior. In Proceedings of the MICAI 2004: Advances in Artificial Intelligence, Mexico City, Mexico, 26–30 April 2004; Volume 2972, pp. 312–321. [CrossRef]
6. Garcia, V.; Sanchez, J.; Mollineda, R. An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. *Prog. Pattern Recognit. Image Anal. Appl. Proc.* **2007**, *4756*, 397–406. [CrossRef]
7. Denil, M.; Trappenberg, T. Overlap versus imbalance. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2010 ; Volume 6085 LNAI, pp. 220–231. [CrossRef]
8. Wilson, D.R.; Martinez, T.R. Reduction Techniques for Instance-Based Learning Algorithms. *Mach. Learn.* **2000**. *38*, 257–286. [CrossRef]
9. Cavalcanti, G.D.; Ren, T.I.; Pereira, C.L. ATISA: Adaptive Threshold-based Instance Selection Algorithm. *Expert Syst. Appl.* **2013**, *40*, 6894–6900. [CrossRef]
10. Cavalcanti, G.D.; Soares, R.J. Ranking-based instance selection for pattern classification. *Expert Syst. Appl.* **2020**, *150*, 113269. [CrossRef]

11. Rout, N.; Mishra, D.; Mallick, M.K. Handling imbalanced data: a survey. In *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 431–443.
12. Le, T.; Lee, M.Y.; Park, J.R.; Baik, S.W. Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset. *Symmetry* **2018**, *10*, 79. [CrossRef]
13. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
14. García, S.; Derrac, J.; Cano, J.R.; Herrera, F. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 417–435. [CrossRef]
15. Branco, P.; Torgo, L.; Ribeiro, R.P. A Survey of Predictive Modeling on Imbalanced Domains. *ACM Comput. Surv.* **2016**, *49*, 1–50. [CrossRef]
16. Kohonen, T. Essentials of the self-organizing map. *Neural Netw.* **2013**, *37*, 52–65. [CrossRef]
17. Rubbo, M.; Silva, L.A. Prototype Selection Using Self-Organizing-Maps and Entropy for Overlapped Classes and Imbalanced Data. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; Volume 1, pp. 1–8. [CrossRef]
18. Arabmakki, E.; Kantardzic, M. SOM-based partial labeling of imbalanced data stream. *Neurocomputing* **2017**, *262*, 120–133. [CrossRef]
19. Douzas, G.; Bacao, F. Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. *Expert Syst. Appl.* **2017**. [CrossRef]
20. Moreira, L.J.; Silva, L.A. Prototype Generation Using Self-Organizing Maps for Informativeness-Based Classifier. *Comput. Intell. Neurosci.* **2017**, *2017*, 1–15. [CrossRef]
21. Basu, M.; Ho, T.K. *Data Complexity in Pattern Recognition*; Springer: London, UK, 2006; p. 297. [CrossRef]
22. Ho, T.K.; Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**. [CrossRef]
23. Sánchez, J.S.; Mollineda, R.A.; Sotoca, J.M. An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Anal. Appl.* **2007**, *10*, 189–201. [CrossRef]
24. Cano, J.R. Analysis of data complexity measures for classification. *Expert Syst. Appl.* **2013**, *40*, 4820–4831. [CrossRef]
25. Morán-Fernández, L.; Bolón-Canedo, V.; Alonso-Betanzos, A. Can classification performance be predicted by complexity measures? A study using microarray data. *Knowl. Inf. Syst.* **2017**, *51*, 1067–1090. [CrossRef]
26. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *5*, 3. [CrossRef]
27. Dheeru, D.; Karra Taniskidou, E. UCI Machine Learning Repository. 2017. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 9 June 2021).
28. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
29. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
30. Fernandes, E.R.; de Carvalho, A.C. Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning. *Inf. Sci.* **2019**, *494*, 141–154. [CrossRef]
31. Vuttipittayamongkol, P.; Elyan, E. Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. *Inf. Sci.* **2020**, *509*, 47–70. [CrossRef]
32. Yongqing, Z.; Min, Z.; Danling, Z.; Gang, M.; Daichuan, M. Improved SMOTEBagging and its application in imbalanced data classification. In *IEEE Conference Anthology*; IEEE: Piscataway, NJ, USA, 2013; pp. 1–5.