

Article

Application of the Gravitational Search Algorithm for Constructing Fuzzy Classifiers of Imbalanced Data

Marina Bardamova ^{*}, Ilya Hodashinsky , Anton Konev  and Alexander Shelupanov

Faculty of Security, Tomsk State University of Control Systems and Radioelectronics, 40 Lenina Prospect, 634050 Tomsk, Russia; hia@fb.tusur.ru (I.H.); kaa1@keva.tusur.ru (A.K.); saa@fb.tusur.ru (A.S.)

^{*} Correspondence: bmb@fb.tusur.ru; Tel.: +7-(3822)-70-15-29

Received: 28 September 2019; Accepted: 24 November 2019; Published: 28 November 2019



Abstract: The presence of imbalance in data significantly complicates the classification task, including fuzzy systems. Due to a large number of instances of bigger classes, instances of smaller classes are not recognized correctly. Therefore, additional tools for improving the quality of classification are required. The most common methods for handling imbalanced data have several disadvantages. For example, methods for generating additional instances of minority classes can worsen classification if there is a strong overlap of instances from different classes. Methods that directly modify the fuzzy classification algorithm lead to a decline in the interpretability of the model. In this paper, we study the efficiency of the gravitational search algorithm in the tasks of selecting the features and tuning the term parameters for fuzzy classifiers of imbalanced data. We consider only data with two classes and apply the algorithm based on extreme values of classes to construct models with a minimum number of rules. In addition, we propose a new quality metric based on the sum of the overall accuracy and the geometric mean with the presence of a priority coefficient between them.

Keywords: fuzzy classifiers; gravitational search algorithm; imbalanced data; geometric mean; feature selection

1. Introduction

The classification task is to divide objects in the feature space into classes or categories based on retrospective observations with the given class label values. Real data are characterized by an imbalanced distribution of classes when the number of instances in some classes exceeds the number of instances in other classes. This situation is mainly explained by the limited occurrence of minority class instances [1]. For example, the normal web browsing traffic is dominant when classifying traffic on the Internet. However, detection of rare malicious connections is very important for training [1]. Similar examples can be given from the field of medical diagnosis, detection of bank fraud, and diagnosis of equipment malfunctions.

The search for regularities in imbalanced data is a difficult task for specialists in data mining, machine learning, pattern recognition, and statistics [2]. The main problem of constructing classifiers of imbalanced data is poor adaption of standard training algorithms, which leads to a significant reduction in the effectiveness of classification. Due to the imbalance between classes, the standard classifier usually defines instances of minority classes incorrectly, since the model is retrained on instances of bigger classes [1].

It is not enough to evaluate the constructed classifier of imbalanced data using the overall accuracy [3]. Positive classes (with the smallest number of instances) are usually more important than negative classes (with the biggest number of instances). Reducing misclassification of minority class instances is crucial in many real-world challenges [4,5]. However, improving the classification quality of positive classes often leads to poor recognition of instances of negative classes, as instances

of different classes often intersect. Thus, in each data classification task, the developer of the data analysis system needs to prioritize; either to focus on improving the overall accuracy or try to correctly identify positive instances with some worsening in the definition of negative ones, or to look for some compromise. Finally, it all depends on the purpose of creating the model and the requirements for it.

There is a large list of classification methods, for example, naive Bayes classifiers, support vector machines, artificial neural networks, and others. Unlike other methods, fuzzy classification does not imply the existence of rigid boundaries between neighboring classes. A classifying object may belong to several classes with various degrees of confidence. The advantage of a fuzzy classifier is understandability and interpretability of the rules, which makes fuzzy classifiers a practically useful data analysis tool.

In many real-world applications, an accurate, but also a computationally simple system, is required. Therefore, we propose to use two procedures for constructing a fuzzy classifier. The first is to shrink the input feature space to reduce complexity. The second is to tune the fuzzy classifier parameters, which increases the definition quality of the output class label. Since these two procedures are formulated as optimization problems, a single optimization algorithm is applied to solve both of them. We use the gravitational search algorithm (GSA), which has previously proven itself well when working with a fuzzy classifier [6].

Since the goal of our work is to improve the efficiency of the fuzzy classifier of imbalanced data, it is necessary to choose an appropriate metric to use as a fitness function for the GSA. We explore the possibilities of applying the following metrics: the overall accuracy, the geometric mean, and a new function that combines the two previous estimates to find a compromise version of the classifier.

The main contributions of this paper are as follows.

1. We propose a new metric based on the sum of the overall accuracy and the geometric mean of each class accuracy. The presence of the coefficient controls the priority of the estimates used.
2. We demonstrated the use of the feature selection method based on the binary gravitational search algorithm in order to reduce the effect of imbalance on classification. The application of the new metric as the fitness function assisted to find subsets of relevant features for both classes.
3. We presented the combination of binary and continuous algorithms for constructing fuzzy classifiers of imbalanced data. The continuous gravitational search algorithm helped to increase the quality of classification on selected features.

This article is organized as follows. Section 2 discusses the levels of problems when working with imbalanced data and provides basic methods for solving them. The procedure for constructing a fuzzy classifier and objective functions under consideration are described in Section 3. Section 3 gives a short description of the gravitational search algorithm. Sections 4 and 5 present the experimental results and their analysis, respectively. Finally, we present the conclusions of our work in Section 6.

2. Related Works

Here, we represent the main approaches to improving the quality of imbalanced data classification. There are three levels of training problems on such data which include: (1) Problems associated with the definition of classification performance indexes, (2) problems related to the learning algorithm, and (3) problems related to the training data [7].

The first level is determined by the lack of an objective method for evaluating (quantitative measures) existing knowledge to select the optimal classifier. The understanding that the overall accuracy is an insufficient measure for classifying imbalanced data has led to the application of new metrics such as the AUC (the area under the ROC curve) [8], the geometric mean, the balanced accuracy, the $F\beta$ -measure, and others [9]. To assess the effectiveness of the classifiers, the authors in [9] have proposed 18 indicators, which are classified into the following three categories:

1. Threshold metrics geared towards minimizing the number of errors, i.e., the overall accuracy, the averaged accuracy (arithmetic and geometric), the $F\beta$ -measure, and the Kappa-statistics;

2. Metrics based on the probabilistic understanding of an error and used to assess the reliability of classifiers, such as the mean absolute error, the mean square error, and the cross-entropy;
3. Metrics based on estimating instance separability, for example, the AUC, which is equivalent to Mann–Whitney–Wilcoxon statistics [9] for two classes.

After analyzing the above 18 indicators, the authors of [7] conclude that the choice of metrics for imbalanced data is of paramount importance. Fernandez et al. [10] have described the use of a multiobjective evolutionary algorithm with a pair of metrics, which are the overall accuracy and the F1 measure. They concluded that the algorithm with simultaneous optimization of this pair of metrics can lead to a balanced accuracy for both classes.

Classification algorithms make some changes to their construction and training processes in order to reduce the influence of imbalance in classes on the classification quality [7].

Cost-sensitive learning methods are based on modifying the classification algorithm so that the costs of misclassifying the instances of minority classes are greater as compared with the instances of majority classes. A typical solution, here, is to use a weight matrix that takes into account the costs of each incorrectly classified instance [11]. This solution is not suitable for a fuzzy classifier since it does not estimate the probability of assigning an object to a particular class.

There is a small list of methods for creating fuzzy classifiers in the presence of imbalance. Weights were added to fuzzy rules in [12–14]. Adding a weight function allows setting the priority of some rules over others when determining the output of the classifier. The weight values are most often configured by optimization algorithms. Another method of changing the fuzzy classification tool is to introduce a bipolar model using the principle of labeling the class, called maximum rule. The adjusted degree of belonging to each class is calculated based on the positive and negative degrees of membership in the bipolar fuzzy classifier [15]. The disadvantage of this model is the need to additionally introduce and adjust the matrix of dissimilarity coefficients and the difficulty to apply this method with another principle of assigning labels. Furthermore, the addition of supplementary modifications to fuzzy systems complicates the interpretation of the resulting models. Consequently, methods for improving the quality of the classifier without interfering directly with the classification algorithm are relevant.

Data play an integral role in machine learning and data mining research. A number of data preprocessing methods have been developed in order to correct the imbalance in the data. Over-sampling methods based on increasing instances of a positive class try to produce a balanced dataset by creating additional instances of the minority class, while undersampling methods reduce the number of majority class instances to achieve a quantitative balance. The most famous representative of oversampling is the SMOTE and its modifications [5,16–18], in which the generation of new instances from a positive class depends on the measure of proximity to existing instances. Among undersampling methods, random undersampling (RUS) is often used. This non-heuristic method aims to eliminate class imbalance by randomly excluding instances of the negative class. Obviously, the disadvantage of RUS is the loss of information about data of a negative class [7,16,19].

Hybrid methods that combine two previous strategies of adding and removing data instances are described in [20,21]. In order to preserve useful information about majority classes, clustering methods have recently been applied [7,22,23].

Preprocessing methods are universal and easy to apply but have low efficiency and cannot be used as the only tool for solving the imbalance problem in classes. In addition, creating new instances of data is not acceptable for some classification tasks. For example, the artificial creation of patient's records can lead to errors in diagnosing diseases.

Another way to change data to improving the quality of recognizing minority classes is by carry out a procedure for selecting informative features. Feature selection consists of selecting, from the input feature space, such a subset that would have fewer attributes but provide comparable classification accuracy relative to the full set. The formed subset should be sufficient to adequately represent all classes in the training samples. Selection methods are usually divided into four types, namely, integrated methods, filters, wrappers, and hybrid methods.

A peculiarity of the integrated (built-in) methods is the principle of feature selection, which is part of the general mechanism of training a model on specific data [24]. An example of applying such methods is the selection of features during training a decision tree [25]. However, not every classification algorithm embeds the selection process into the learning process.

Filtering methods, on the contrary, are universal, as they are used independently of the classifier at the stage of data preparation. Four groups of filters are distinguished in [26]. The methods which make up the first group are based on the distance. They select features that provide the greatest distance between classes. The second group of filters uses the calculation of the amount of information. Such methods select features which, when attached to an existing set, reduce its entropy [27]. The third group determines the relationship between features and classes using the correlation coefficient or mutual information [28]. The fourth group is represented by filters that minimize the number of inconsistent features. A case of inconsistency is the presence of two instances belonging to different classes but having the same values of the same features. Filter algorithms are easy to use but have low efficiency.

Wrappers are methods that evaluate each subset of features based on the effectiveness of the constructed classifier. As a search algorithm, they usually use metaheuristic algorithms. Since such algorithms are iterative, the classifier needs to be reconstructed after each iteration. Wrapper methods can require considerable time and resources for large datasets [24]. The advantage of wrappers is the ability to choose a set of features that will be optimal for a particular classification algorithm.

The method of applying a genetic algorithm for feature selection in the wrapper mode based on the SVM classifier is described in [29]. The fitness function of this algorithm is a measure consisting of a compromise between the geometric mean and the share of selected features. The results showed that the proposed method selects features that improve the recognition of minority classes.

Hybrid feature selection methods consist of a combination of filters and wrappers. First, a filter is used for preliminary selection, then a classifier is built on the resulting subset and a wrapper algorithm is launched [30]. This approach is described in [31], which uses symmetric uncertainty for filtering in order to weigh features relative to their dependence on class labels, and the harmonic search as the wrapper algorithm. Hybrid selection methods can be a good solution for data with a large number of features.

3. Materials and Methods

3.1. The Fuzzy Classifier

3.1.1. The Fuzzy Classifier Structure

Classification algorithms determine the most suitable class from the set of all classes $\mathbf{C} = \{c_1, c_2, \dots, c_l\}$ to each object $\mathbf{x}_p = \{x_{p1}, x_{p2}, \dots, x_{pm}\}$ from the set of n objects ($p \in [1, n]$), where x_{pk} is the value of the k th feature of the p th object, $k \in [1, m]$, m is the number of features. The fuzzy classifier is constructed on the basis of production rules, each of which has its own set of fuzzy terms. A fuzzy term is a structure on the feature definition domain, reflecting the degree of object membership to a rule. The terms can be described by membership functions of various kinds such as triangles, trapezoids, bells, or Gaussian-type functions. In this work, we used the membership functions of the Gaussian type, which differ from others by the property of symmetry. Figure 1 shows an example of partitioning some attribute x_1 by Gaussian terms.

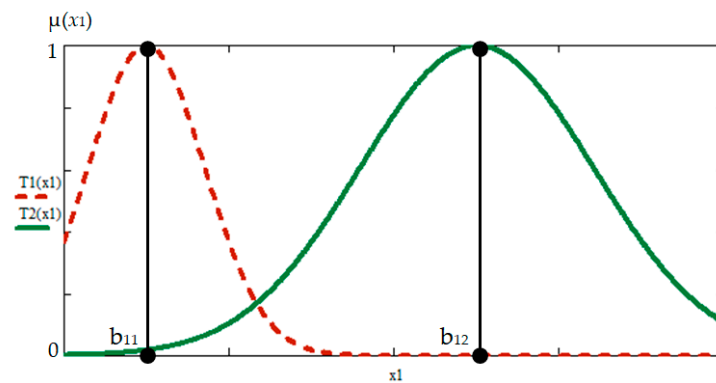


Figure 1. Example of a fuzzy partition of feature x_1 by two symmetric fuzzy terms.

A Gaussian fuzzy term characterizing the k th feature in the i th rule is given by the following expression:

$$T_{ik}(\mathbf{x}) = e^{-\left(\frac{x_k - b_{ki}}{c_{ki}}\right)^2},$$

where i is the rule number to which the term ($i \in [1, r]$) belongs, r is the number of fuzzy rules, b is the coordinate of the term vertex, and c is the function dispersion. The term parameters listed sequentially for each feature compose the antecedent vector $\theta = (b_{11}, c_{11}, b_{12}, c_{12}, b_{13}, c_{13}, b_{21}, c_{21}, \dots, b_{mr}, c_{mr})$.

The standard fuzzy rule consists of the antecedent part, which lists the variables and their terms, and the consequent part, which specifies the output class label as:

$$R_i: \text{If } x_1 \text{ is } T_{i1} \text{ and } x_2 \text{ is } T_{i2} \text{ and } \dots \text{ and } x_m \text{ is } T_{im} \text{ then class is } c_j,$$

where c_j is the label of the j th class from the set of classes \mathbf{C} , $class$ is an output variable.

To use the possibility of feature selection in the wrapper mode, the binary feature vector $\mathbf{S} = (s_1, s_2, \dots, s_m)$ must be introduced into the antecedent part. If $s_k = 1$, then the k th feature is taken into account in the classification; otherwise the feature is ignored. Given the vector \mathbf{S} , the fuzzy rule will change as follows:

$$R_i: \text{If } (s_1 \wedge x_1) \text{ is } T_{i1} \text{ and } (s_2 \wedge x_2) \text{ is } T_{i2} \text{ and } \dots \text{ and } (s_m \wedge x_m) \text{ is } T_{im} \text{ then class is } c_j,$$

where the record $(s_p \wedge x_p)$ indicates the use ($s_p = 1$) or ignorance ($s_p = 0$) of the feature and its terms in the classifier. The binary vector $\mathbf{S} = (s_1, s_2, \dots, s_m)$ is formed by the feature selection algorithm.

3.1.2. Generation of the Fuzzy Rule Base

There are various methods for generating fuzzy terms and forming a fuzzy classifier rule base such as uniform partitioning, random generation, clustering [32], and others. In this paper, we apply an algorithm based on the extreme values of classes of the training data. This algorithm constructs compact classifiers by using the minimum possible number of rules. In this case, the number of rules is equal to the number of classes, that is, there is one rule for each class.

The algorithm based on extreme values of classes is presented in [6]. The first step is to determine the minimum and maximum values of the features for each class. In the second step, the terms are generated in such a way that the entire definition area is covered in the interval between the two extremes, and the top of the term is located in the middle of this segment. In the third step, the rule base is formed. Each feature is represented in the rule by only one term. The terms belonging to each separate class are combined in the antecedent part of the rule by the conjunction operation. The consequent part of the rule contains the label of this class.

The presented algorithm is very simple, but its efficiency is not high. Therefore, it is necessary to use parameter tuning as an additional training step. The description of the procedure for term parameter tuning with the gravitational search algorithm is given in Section 3.2.

3.1.3. Output of Fuzzy Classifier

The output of the classifier for the input string \mathbf{x}_p is formed by sequentially performing three steps. In the first step, the value of the membership function of the object to each term is calculated:

$$\mu_{ik}(x_{pk}) = T_{ik}(x_{pk}).$$

The degree of the object membership to each rule is evaluated in the second step:

$$\beta_i(\mathbf{x}_p) = \prod_{k=1}^m \mu_{ik}(x_{pk}).$$

The third step is to define the output class by the maximum rule. The output of the classifier will be the class that corresponds to the rule with the highest degree of membership:

$$\text{class}(\mathbf{x}_p) = c_{j^*}, j^* = \arg \max_{1 \leq i \leq m} \beta_i(\mathbf{x}_p),$$

After the procedure of forming the output has finished, the constructed model can be evaluated using various performance indexes.

3.1.4. Classification Quality Evaluation

The most common classification quality criterion is the overall accuracy, which is the percentage of correct classification. In the observation table $\{(\mathbf{x}_p; c_p), p \in [1, z]\}$, where z is the number of instances, the measure of accuracy can be given as follows:

$$Acc(\boldsymbol{\theta}, \mathbf{S}) = \frac{\sum_{p=1}^z \begin{cases} 1, & \text{if } c_p = \arg \max_{1 \leq j \leq m} f_j(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S}) \\ 0, & \text{otherwise} \end{cases}}{z},$$

where $f(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S})$ is the output of the fuzzy classifier with the parameter vector $\boldsymbol{\theta}$ and the binary feature vector \mathbf{S} at the point \mathbf{x}_p . As noted earlier, the overall accuracy is not an objective assessment of classification quality when there is an imbalance in the class distribution.

The geometric mean is a sensitive estimate for the accuracy of each class:

$$GM(\boldsymbol{\theta}, \mathbf{S}) = \left(\prod_{i=1}^l Acc_i(\boldsymbol{\theta}, \mathbf{S}) \right)^{\frac{1}{l}},$$

where $Acc_i(\boldsymbol{\theta}, \mathbf{S})$ is the classification accuracy of i th class:

$$Acc_i(\boldsymbol{\theta}, \mathbf{S}) = \frac{\sum_{p=1}^{z_i} \begin{cases} 1, & \text{if } c_p = \arg \max_{1 \leq j \leq m} f_j(\mathbf{x}_p; \boldsymbol{\theta}, \mathbf{S}) \\ 0, & \text{otherwise} \end{cases}}{z_i},$$

where z_i is the number of instances with the i th class label. Thus, the fewer instances represent a class, the geometric mean increases more significantly with an increment in the number of correctly classified instances of that class. In the case when one of the classes is classified absolutely incorrectly, the geometric mean is zero.

While using general accuracy as the objective function, on the one hand, the classifier prefers to focus on recognizing negative classes. The geometric mean, on the other hand, can lead to a large loss in the quality of classification of negative classes, even if the accuracy of positive classes is low. We propose to use a compromise option that combines both of these metrics and allows varying their importance degree using the coefficient $\gamma \in [0; 1]$:

$$Fit(\boldsymbol{\theta}, \mathbf{S}) = \gamma \cdot GM(\boldsymbol{\theta}, \mathbf{S}) + (1 - \gamma) \cdot Acc(\boldsymbol{\theta}, \mathbf{S}).$$

The problem of constructing a fuzzy classifier reduces to searching for the maximum of the selected function.

3.2. Training a Classifier with the Gravitational Search Algorithm

For selecting feature and tuning term parameters we suggest using the gravitational search algorithm in two versions, i.e., binary for optimizing the binary feature vector \mathbf{S} and continuous for optimizing the continuous vector of term parameters $\boldsymbol{\theta}$. The GSA was first proposed by Rashedi, Nezamabadi-pour, and Saryazdi in 2009 [33], and in the same year, its binary version was described [34]. This algorithm is widely used to solve various problems. For example, the GSA was applied to optimize parameters in a geothermal power generation system in the study of Özkara and Keçebaş [35], to determine the location of a microseismic source in order to warn about explosions in tunnels in [36]. Mahanipour and Nezamabadi-pour described the use of GSA for the automatic creation of computer programs in [37] and the feature construction in [38].

The application of the binary and the continuous versions of the GSA for the fuzzy classifier has been described in detail earlier in [6]. In the binary GSA, a population of particles corresponding to binary feature vectors \mathbf{S} is generated randomly. At each iteration, the algorithm calculates particle masses, gravity, acceleration, and velocity. Transformation functions are applied to transform the obtained speed value into a binary equivalent in order to update the feature vector. In this paper, we use the V-type transformation function:

$$\text{IF } (rand(0;1) < \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2} v_i^d(t+1)\right) \right|), \text{ then } s_i^d(t+1) = \overline{s_i^d(t)}, \text{ else } s_i^d(t+1) = s_i^d(t),$$

where $rand(0;1)$ is a random number in the range from 0 to 1, v_i^d is the speed of the d th element of the i th particle, s_i^d is the value of the d th element of the i th feature vector, and t is the iteration number.

The continuous GSA optimizes the numerical vector $\boldsymbol{\theta}$, consisting of the term parameters. In this version of the algorithm, the population is formed as follows: The first vector is input to the algorithm after the stage of creating the classifier structure and the remaining vectors are generated based on the first one with some deviation. Unlike the binary version, in GSA_c the vector value is updated by the simple addition of the current value and the calculated speed:

$$\theta_i^d(t+1) = \theta_i^d(t) + V_i^d(t+1),$$

where θ_i^d is the value of the d th element of the i th vector.

Five parameters are used in both versions of the GSA: the number of iterations t , the number of particles P , the value of the gravitational constant G_0 , the coefficient of the gravitational constant decrease α , and the variable for calculating the attractive force ϵ . The computational complexity of the GSA with n agents is $O(n \times d)$ where d is the search space dimension [39]. We did not modify the original GSA, therefore, both algorithms have the complexity $O(P \times d)$, where P is the number of particles and d is the size of the dataset.

The classifier training procedure is as follows: After the algorithm based on extremes values of classes has created the initial vector $\boldsymbol{\theta}$, the binary GSA searches for the optimal vector \mathbf{S} ; then, the classifier is rebuilt on the obtained set of features \mathbf{S}_{best} and the algorithm for optimizing the

term parameters is launched; the continuous GSA runs for a given number of iterations and provides the best parameter vector θ_{best} ; and the resulting S_{best} and θ_{best} are used to construct and validate the classifier on test data.

4. Experimental Results

The experiment was performed on imbalanced binary datasets from the KEEL repository [40]. The sets are described in Table 1. Here, F_{all} is the number of features in a dataset, Str_{all} is the number of lines, Str_{+} is the number of rows of the smallest class, Str_{-} is the number of rows of the largest class, and IR is the imbalance ratio. The imbalance ratio is the ratio of the number of rows of a negative class to the number of rows of a positive class.

Table 1. Description of the datasets used in the experiment.

No	Data Set	F_{all}	Str_{all}	Str_{+}	Str_{-}	IR
1	vehicle0	18	846	199	647	3.25
2	newthyroid2	5	215	35	180	5.14
3	segment0	19	2308	329	1979	6.02
4	page-blocks0	10	5472	559	4913	8.79
5	vowel0	13	988	90	898	9.98
6	cleveland-0vs4	13	177	13	164	12.62
7	ecoli4	7	336	20	316	15.8
8	yeast4	8	1484	51	1433	28.1

Five-fold cross-validation was applied in all stages of the experiment. The data were divided into five pairs of training and test samples. The structure of the fuzzy classifier was formed by the algorithm based on the extreme values of classes with symmetric Gaussian terms. Since only two classes are represented in all data, the number of rules in all cases was two.

In the first stage of the experiment, the efficiency of the continuous gravitational algorithm was tested when the priority coefficient γ in the fitness function was changed. The tuning of the fuzzy classifier parameters was carried out on full sets of features. The following parameters were set for the GSA_c : 750 iterations, 15 particles, $G_0 = 10$, $\alpha = 10$, and $\varepsilon = 0.01$. The particle population was cleared after each 150th iteration, except for the best particle on the basis of which the population was generated anew. The parameters were chosen empirically as the most universal for the selected datasets.

Table 2 contains the results of the first experimental stage, used to assess the quality of the constructed model based on the following: the classification accuracy, the geometric mean, as well as the percentage of correctly classified instances of the positive class relative to the total number of instances of the positive class (true positive rate) and the percentage of correctly classified instances of the negative class relative to the total number of instances of the negative class (true negative rate). The table shows the results obtained on the test data as an average of three runs (Avr.), and the best one (Best).

The purpose of the second experimental stage consisted of verifying the effectiveness of GSA on the task of selecting features in the wrapper mode for the fuzzy classifier of imbalanced data. The binary gravitational algorithm with the same coefficient γ was run three times on each sample. Due to the stochasticity of the algorithm, one to three different feature sets could be obtained on the same sample. Next, a set of features with the highest fitness function value was selected. A classifier was built on this set; the parameters of the created model were tuned by the continuous algorithm. The obtained values of quality indicators were averaged over three independent runs of the GSA_c .

Table 2. Classification results obtained while using the continuous gravitational search algorithm for tuning fuzzy classifier parameters.

γ	0		1		0.25		0.5		0.75	
	Avr.	Best	Avr.	Best	Avr.	Best	Avr.	Best	Avr.	Best
vehicle0										
Acc.	81.64	82.50	81.52	82.03	84.75	85.11	82.62	86.28	82.82	84.28
GM	57.47	59.56	69.28	70.36	74.85	81.45	74.28	81.63	74.12	80.40
TP _{rate}	34.84	37.69	54.10	55.78	62.56	77.23	63.65	74.87	62.93	74.82
TN _{rate}	96.03	96.29	89.95	90.11	91.55	87.48	88.46	89.80	88.91	87.16
newthyroid2										
Acc.	98.76	99.07	98.76	99.07	98.60	99.07	98.91	99.53	98.45	99.07
GM	98.05	98.24	98.85	99.44	98.35	99.44	98.54	99.72	97.46	98.24
TP _{rate}	97.14	97.14	99.05	100.00	98.10	100.00	98.10	100.00	96.19	97.14
TN _{rate}	99.07	99.44	98.70	98.89	98.70	98.89	99.07	99.44	98.89	99.44
segment0										
Acc.	91.29	91.42	90.83	90.90	91.41	91.46	91.13	91.16	90.87	90.90
GM	92.36	92.82	94.15	94.31	93.53	93.57	93.85	93.99	94.05	94.07
TP _{rate}	93.92	94.83	99.09	99.39	96.65	96.65	97.87	98.18	98.78	98.78
TN _{rate}	90.85	90.85	89.46	89.49	90.53	90.60	90.01	89.99	89.56	89.59
page-blocks0										
Acc.	93.24	93.93	88.96	91.03	93.37	94.19	92.85	93.59	90.99	91.01
GM	65.30	72.39	76.79	80.59	74.64	77.05	74.01	79.42	74.18	78.28
TP _{rate}	44.07	53.31	64.52	69.59	57.36	60.64	56.95	65.30	58.64	65.86
TN _{rate}	98.83	98.55	91.74	93.47	97.47	98.01	96.93	96.80	94.67	93.87
vowel0										
Acc.	92.11	92.71	88.59	89.67	96.86	97.67	96.19	96.86	95.75	96.46
GM	47.99	54.09	90.22	92.15	93.94	96.69	95.01	96.75	94.75	97.04
TP _{rate}	36.67	46.67	92.59	95.56	90.74	95.56	93.70	97.78	93.70	97.78
TN _{rate}	97.66	95.88	88.20	89.09	97.48	97.89	96.44	94.77	95.96	96.33
cleveland-0vs4										
Acc.	92.86	95.51	87.03	90.37	91.76	95.49	90.79	93.25	88.92	90.41
GM	54.43	73.17	74.50	80.36	71.47	73.00	66.10	72.20	70.73	76.32
TP _{rate}	38.46	53.85	61.54	69.23	56.67	56.67	46.15	53.85	57.78	66.67
TN _{rate}	97.15	98.78	89.02	92.07	94.73	98.77	94.31	96.34	91.67	92.69
ecoli4										
Acc.	96.91	97.32	94.25	97.02	96.92	97.62	95.78	95.84	94.84	95.53
GM	76.17	79.06	91.06	95.90	78.71	81.74	78.25	85.83	82.07	85.73
TP _{rate}	61.00	65.00	88.33	95.00	65.00	70.00	66.00	80.00	73.33	80.00
TN _{rate}	99.18	99.37	94.62	97.15	98.94	99.37	97.66	96.84	96.20	96.52
yeast4										
Acc.	96.52	96.63	81.81	86.66	92.57	92.05	89.31	88.54	85.47	88.88
GM	2.11	6.32	78.28	80.18	69.12	74.76	76.55	83.22	74.41	78.20
TP _{rate}	0.65	1.96	75.16	74.51	51.45	60.55	66.01	78.43	64.61	68.55
TN _{rate}	100.00	100.00	82.04	87.09	94.03	93.17	90.14	88.90	86.21	89.60

The following parameters were empirically selected for the binary gravitational algorithm: 750 iterations, 15 particles, $G_0 = 10$, $\alpha = 10$, and $\epsilon = 0.01$. The parameters of the continuous algorithm did not differ from those used at the first stage of the experiment. Table 3 shows the results of the classifier on the selected feature sets before parameter tuning (GSA_b) and after optimization ($GSA_b + GSA_c$). In the following table and further, formatting the cells according to a color scale was used to visualize the results. The values presented in each row were compared with each other. The hue of the color depended on the relative magnitude of the value compared to other cells in the row. Thus, the worst results are marked in red, the best are highlighted in green, the remaining values are colored in intermediate colors.

Table 3. The results of constructing fuzzy classifiers on imbalanced datasets obtained with feature selection and parameter tuning.

γ	0	0	1	1	0.5	0.5
	GSA _b	GSA _b + GSA _c	GSA _b	GSA _b + GSA _c	GSA _b	GSA _b + GSA _c
Dataset	vehicle0					
Features	10.20		7.60		9.00	
Accuracy	83.33	84.43	80.61	77.07	81.09	84.28
GM	66.40	67.25	75.28	78.01	71.78	78.28
TP _{rate}	47.24	47.91	67.34	83.08	58.79	69.35
TN _{rate}	94.44	95.67	84.70	75.22	87.94	88.87
Dataset	newthyroid2					
Features	3.60		3.20		3.20	
Accuracy	99.53	99.07	98.60	98.45	98.60	98.45
GM	98.52	97.03	99.16	98.66	99.16	98.66
TP _{rate}	97.14	94.29	100.00	99.05	100.00	99.05
TN _{rate}	100.00	100.00	98.33	98.33	98.33	98.33
Dataset	segment0					
Features	7.20		6.60		6.80	
Accuracy	97.36	97.88	96.45	98.73	97.40	98.60
GM	95.76	96.80	96.28	98.79	97.08	98.34
TP _{rate}	93.62	95.34	96.05	98.89	96.66	97.97
TN _{rate}	97.98	98.30	96.51	98.70	97.52	98.70
Dataset	page-blocks0					
Features	3.80		4.20		2.80	
Accuracy	93.60	94.49	88.54	88.13	92.20	92.59
GM	67.85	74.65	74.14	81.93	73.31	76.89
TP _{rate}	46.69	56.65	60.00	75.19	56.17	61.96
TN _{rate}	98.94	98.80	91.80	89.61	96.30	96.07
Dataset	vowel0					
Features	6.20		6.60		6.60	
Accuracy	88.86	92.11	87.45	97.64	88.25	97.20
GM	85.64	75.59	90.02	96.97	88.94	94.85
TP _{rate}	82.22	67.78	93.33	96.30	90.00	92.22
TN _{rate}	89.53	94.54	86.86	97.77	88.08	97.70
Dataset	cleveland-0vs4					
Features	4.00		6.80		6.60	
Accuracy	93.78	93.79	88.70	92.06	85.86	89.97
GM	39.17	47.80	82.38	82.46	68.01	66.57
TP _{rate}	30.77	33.33	76.92	74.36	53.85	48.72
TN _{rate}	98.78	98.58	89.63	93.50	88.41	93.29
Dataset	ecoli4					
Features	3.00		3.20		3.00	
Accuracy	98.21	98.02	96.13	94.14	97.92	97.12
GM	89.01	86.89	87.35	84.36	85.81	87.11
TP _{rate}	80.00	76.67	80.00	76.67	75.00	78.33
TN _{rate}	99.37	99.37	97.15	95.25	99.37	98.31
Dataset	yeast4					
Features	3.20		3.20		2.40	
Accuracy	96.23	96.23	78.24	84.05	87.26	90.43
GM	6.30	6.30	66.99	77.69	67.05	79.26
TP _{rate}	1.96	1.96	58.82	71.90	52.94	69.28
TN _{rate}	99.58	99.58	78.93	84.48	88.49	91.18

Table 4 shows fuzzy classifiers based on the best feature sets. The best sets here are those that gain the highest averaged value of the objective function with a given value γ over five samples.

Table 4. The results of constructing fuzzy classifiers on the best feature sets found by the binary gravitational algorithm.

Metrics	Results		
DataSet	vehicle0		
γ	0	1	0.5
Features	1, 4, 8, 9, 10, 13, 14, 15, 16	1, 4, 6, 7, 9, 10, 12, 13, 15, 16, 18	1, 5, 7, 9, 10, 11, 12, 15, 16, 17, 18
F	9	11	11
Acc.	85.07	78.30	84.00
GM	66.87	82.86	80.58
TP _{rate}	46.40	93.33	75.88
TN _{rate}	96.96	73.64	86.50
Dataset	newthyroid2		
γ	0	1	0.5
Features	1, 2, 3, 5	1, 2, 5	1, 2, 5
F	4	3	3
Acc.	99.53	99.84	99.53
GM	98.52	99.51	99.32
TP _{rate}	97.14	99.05	99.05
TN _{rate}	100.00	100.00	99.63
Dataset	segment0		
γ	0	1	0.5
Features	1, 4, 6, 11, 14, 18, 19	1, 6, 8, 14, 16, 18	6, 8, 11, 14, 18, 19
F	7	6	6
Acc.	98.93	99.08	99.02
GM	98.22	99.08	98.66
TP _{rate}	97.26	99.09	98.18
TN _{rate}	99.21	99.07	99.16
Dataset	page-blocks0		
γ	0	1	0.5
Features	1, 2, 5, 10	4, 10	4, 10
F	4	2	2
Acc.	94.77	91.75	92.85
GM	77.06	84.92	81.52
TP _{rate}	60.35	77.22	70.60
TN _{rate}	98.68	93.41	95.39
Dataset	vowel0		
γ	0	1	0.5
Features	5, 6, 7, 8, 9, 10, 13	4, 5, 6, 7, 9, 13	4, 5, 6, 7, 8, 13
F	7	6	6
Acc.	96.39	98.18	97.74
GM	82.83	98.11	97.41
TP _{rate}	70.00	98.15	97.04
TN _{rate}	99.03	98.18	97.81
Dataset	cleveland-0vs4		
γ	0	1	0.5
Features	4, 8, 10	1, 4, 7, 9, 10, 13	10, 12
F	3	6	2
Acc.	94.72	93.04	93.02
GM	54.96	85.19	86.17
TP _{rate}	41.03	76.92	82.05
TN _{rate}	98.98	94.31	93.90
Dataset	ecoli4		
γ	0	1	0.5
Features	5, 6, 7	2, 3, 4, 5, 7	2, 3, 5, 7
F	3	5	4
Acc.	98.71	96.13	97.92
GM	88.90	90.11	93.88
TP _{rate}	80.00	85.00	90.00
TN _{rate}	99.89	96.84	98.42
Dataset	yeast4		
γ	0	1	0.5
Features	1, 2, 3, 7, 8	1, 3, 5	1, 3
F	5	3	2
Acc.	95.62	84.05	91.19
GM	19.73	79.93	80.40
TP _{rate}	9.8	76.47	70.59
TN _{rate}	98.67	84.32	91.93

5. Discussion

To confirm the effectiveness of the gravitational algorithm for optimizing the fuzzy classifier of imbalanced data, we performed a five-stage comparison.

The task of the first stage was to check the quality of the fuzzy classifier in the presence of feature selection. For this purpose, we compared the results of fuzzy classifiers constructed on complete datasets (Table 1, average values for three runs) with those built on abbreviated sets of features (Table 3). In both cases, the results obtained after setting the GSA_c parameters were taken into account. Table 5 shows the results of the pairwise comparison of the number of features by Wilcoxon's sign rank criterion for linked samples. The significance level is 0.05; the null hypothesis states that the difference median between the two samples is zero.

Table 5. The results of comparing classifiers by the number of selected features.

Feature Sets	Standardized Test Statistic	<i>p</i> -Value	Null Hypothesis
$F_{all} - F_{bin}, \gamma = 0$	2.521	0.012	Reject
$F_{all} - F_{bin}, \gamma = 1$	2.521	0.012	Reject
$F_{all} - F_{bin}, \gamma = 0,5$	2.524	0.012	Reject
$F_{bin}, \gamma = 0 - F_{bin}, \gamma = 1$	0	1	Retain
$F_{bin}, \gamma = 0 - F_{bin}, \gamma = 0.5$	0.851	0.395	Retain
$F_{bin}, \gamma = 1 - F_{bin}, \gamma = 0.5$	0.638	0.524	Retain

The first three rows of the table are the comparison of the number of features in the original set (F_{all}) and in the selected feature sets (F_{bin}). The last three rows are the comparison of the number of features when using the GSA_b with different values of the coefficient γ in the fitness function.

On the basis of the results of the verification, we conclude that the binary gravitational algorithm can significantly reduce the number of features working with imbalanced data in the wrapper mode of the fuzzy classifier. In addition, there is no significant difference in the number of features when using one or another value of γ .

Table 6 shows the results of comparing the performance indexes for classifiers built on complete and selected sets of features when changing the priority coefficient γ in the fitness function. The obtained values of the Wilcoxon's sign rank criteria are grouped for each of four quality indexes (the total accuracy, the geometric mean, the percentage of correctly classified instances of the positive class, and the percentage of correctly identified instances of the negative class).

Table 6. The results of comparing classification performance indexes in the absence and presence of feature selection performed using the binary gravitational search algorithm.

Metric	γ	Standardized Test Statistic	<i>p</i> -Value	Null Hypothesis
Accuracy (all - bin)	0	-2.197	0.028	Reject
	1	-0.98	0.327	Retain
	0.5	-1.68	0.093	Retain
GM (all - bin)	0	-1.82	0.069	Retain
	1	-1.4	0.161	Retain
	0.5	-2.24	0.025	Reject
TP _{rate} (all - bin)	0	-1.544	0.123	Retain
	1	-1.051	0.293	Retain
	0.5	-2.036	0.042	Reject
TN _{rate} (all - bin)	0	-0.73	0.465	Retain
	1	-0.594	0.553	Retain
	0.5	-0.877	0.38	Retain

Thus, the results of the first stage of the comparison show that the use of the GSA_b for selecting features in the wrapper mode of the fuzzy classifier of imbalanced data significantly reduces the number of features while maintaining or increasing the quality of classification.

In the second stage, the effectiveness of the binary gravitational algorithm was tested in comparison with popular methods of selecting features. We used a random search (RS) and a filtering algorithm based on mutual information (MI).

The filter was executed as follows: The value of mutual information was calculated for each feature with three randomly-selected neighbors. Next, the algorithm found the arithmetic mean of these values. The set of selected features included only those variables whose mutual information exceeded the value of the arithmetic mean. Both algorithms were run three times, among the obtained feature sets, those with the best accuracy were selected. Fuzzy classifiers were constructed on the selected feature sets using the algorithm based on extreme values of classes. The obtained values were compared with the results of fuzzy classifiers built on the feature sets found by the GSA_b (Table 3). In this case, we considered the results without optimizing parameters. The average performance indexes of the classifiers are given in Table 7 (*F* is the number of features).

Table 7. The results of constructing fuzzy classifiers obtained with the different feature selection algorithms.

Alg.	GSA _b			RS	MI	Alg.	GSA _b			RS	MI
	$\gamma = 0$	$\gamma = 1$	$\gamma = 0.5$				$\gamma = 0$	$\gamma = 1$	$\gamma = 0.5$		
Data	vehicle0					Data	vowel0				
<i>F</i>	10.20	7.60	9.00	6.60	8.40	<i>F</i>	6.20	6.60	6.60	5.80	5.40
<i>Acc.</i>	83.33	80.61	81.09	70.08	79.43	<i>Acc.</i>	88.86	87.45	88.25	77.93	85.29
<i>GM</i>	66.40	75.28	71.78	62.67	65.45	<i>GM</i>	85.64	90.02	88.94	81.13	75.59
<i>TP_{rate}</i>	47.24	67.34	58.79	53.29	48.22	<i>TP_{rate}</i>	82.22	93.33	90.00	77.17	70.00
<i>TN_{rate}</i>	94.44	84.70	87.94	75.26	89.02	<i>TN_{rate}</i>	89.53	86.86	88.08	85.56	86.55
Data	newthyroid2					Data	cleveland-0_vs_4				
<i>F</i>	3.60	3.20	3.20	2.80	3.00	<i>F</i>	4.00	6.80	6.60	6.40	3.00
<i>Acc.</i>	99.53	98.60	98.60	95.35	99.53	<i>Acc.</i>	93.78	88.70	85.86	53.51	98.22
<i>GM</i>	98.52	99.16	99.16	94.85	98.52	<i>GM</i>	39.17	82.38	68.01	39.52	88.49
<i>TP_{rate}</i>	97.14	100.0	100.0	94.29	97.14	<i>TP_{rate}</i>	30.77	76.92	53.85	53.75	80.00
<i>TN_{rate}</i>	100.0	98.33	98.33	95.56	100.0	<i>TN_{rate}</i>	98.78	89.63	88.41	56.67	99.37
Data	segment0					Data	ecoli4				
<i>F</i>	7.20	6.60	6.80	10.60	9.40	<i>F</i>	3.00	3.20	3.00	6.40	4.80
<i>Acc.</i>	97.36	96.45	97.40	90.99	91.12	<i>Acc.</i>	98.21	96.13	97.92	96.73	87.34
<i>GM</i>	95.76	96.28	97.08	88.67	85.08	<i>GM</i>	89.01	87.35	85.81	68.70	88.90
<i>TP_{rate}</i>	93.62	96.05	96.66	85.72	77.85	<i>TP_{rate}</i>	80.00	80.00	75.00	50.00	91.11
<i>TN_{rate}</i>	97.98	96.51	97.52	91.86	93.33	<i>TN_{rate}</i>	99.37	97.15	99.37	99.68	86.96
Data	page-blocks0					Data	yeast4				
<i>F</i>	3.80	4.20	2.80	6.80	5.60	<i>F</i>	3.20	3.20	2.40	6.40	3.00
<i>Acc.</i>	93.60	88.54	92.20	81.49	87.65	<i>Acc.</i>	96.23	78.24	87.26	94.21	91.24
<i>GM</i>	67.85	74.14	73.31	59.80	51.98	<i>GM</i>	6.30	66.99	67.05	29.46	62.79
<i>TP_{rate}</i>	46.69	60.00	56.17	42.04	31.82	<i>TP_{rate}</i>	1.96	58.82	52.94	16.00	45.64
<i>TN_{rate}</i>	98.94	91.80	96.30	85.98	94.00	<i>TN_{rate}</i>	99.58	78.93	88.49	97.00	92.88

Table 8 demonstrates the results of a pairwise comparison of the performance indexes of the obtained systems by the criterion of Wilcoxon’s sign ranks for linked samples. Here *STS* is the standardized test statistic, *p* is the *p*-value, and *NH* is the null hypothesis. The left half of Table 8 shows the results of the comparison with the random search algorithm, the right half of the table demonstrates the comparison with the filter based on mutual information.

Table 8. Comparison of fuzzy classifier results obtained using different algorithms for feature selection.

Algorithm	STS	<i>p</i>	NH	Algorithm	STS	<i>p</i>	NH
<i>Features</i>							
RS - GSA ($\gamma = 0$)	0.981	0.326	Retain	MI - GSA ($\gamma = 0$)	0.281	0.778	Retain
RS - GSA ($\gamma = 1$)	1.123	0.261	Retain	MI - GSA ($\gamma = 1$)	0.421	0.674	Retain
RS - GSA ($\gamma = 0.5$)	1.122	0.262	Retain	MI - GSA ($\gamma = 0.5$)	0.35	0.726	Retain
<i>Accuracy</i>							
RS - GSA ($\gamma = 0$)	-2.521	0.012	Reject	MI - GSA ($\gamma = 0$)	-1.859	0.063	Retain
RS - GSA ($\gamma = 1$)	-1.4	0.161	Retain	MI - GSA ($\gamma = 1$)	-0.14	0.889	Retain
RS - GSA ($\gamma = 0.5$)	-1.96	0.05	Reject	MI - GSA ($\gamma = 0.5$)	-0.7	0.484	Retain
<i>GM</i>							
RS - GSA ($\gamma = 0$)	-1.26	0.208	Retain	MI - GSA ($\gamma = 0$)	-0.169	0.866	Retain
RS - GSA ($\gamma = 1$)	-2.521	0.012	Reject	MI - GSA ($\gamma = 1$)	-1.68	0.093	Retain
RS - GSA ($\gamma = 0.5$)	-2.521	0.012	Reject	MI - GSA ($\gamma = 0.5$)	-1.26	0.208	Retain
<i>TP_{rate}</i>							
RS - GSA ($\gamma = 0$)	-0.14	0.889	Retain	MI - GSA ($\gamma = 0$)	0.338	0.735	Retain
RS - GSA ($\gamma = 1$)	-2.521	0.012	Reject	MI - GSA ($\gamma = 1$)	-1.82	0.069	Retain
RS - GSA ($\gamma = 0.5$)	-2.371	0.018	Reject	MI - GSA ($\gamma = 0.5$)	-0.84	0.401	Retain
<i>TN_{rate}</i>							
RS - GSA ($\gamma = 0$)	-2.383	0.017	Reject	MI - GSA ($\gamma = 0$)	-2.197	0.028	Reject
RS - GSA ($\gamma = 1$)	-1.12	0.263	Retain	MI - GSA ($\gamma = 1$)	0.84	0.401	Retain
RS - GSA ($\gamma = 0.5$)	-1.682	0.092	Retain	MI - GSA ($\gamma = 0.5$)	-0.14	0.889	Retain

The algorithms are statistically indistinguishable by the number of selected features. But the value of the standardized test statistic shows that fuzzy classifiers, constructed on the features selected by the gravitational search algorithm, have higher classification quality values in most cases. Hence, the binary gravitational algorithm is more preferable for imbalanced data classification in contrast to the random search or the filter based on mutual information.

In the third stage of the comparison, we compared our results with fuzzy classifiers based on imbalanced data preprocessed by the SMOTE algorithm. We used a realization of the algorithm from the open library [41] and all parameters were taken by default. After applying SMOTE, the number of instances of the positive and negative classes was equal. Next, we conducted five-fold cross-validation. Fuzzy classifiers were constructed with the algorithm based on the extreme values of the classes. The feature selection was not produced. Table 9 presents the results of fuzzy classifiers averaged over five samples.

Table 9. Results of fuzzy classifiers after using the over-sampling algorithm.

Metrics	vhc0	nth2	sgm0	pb10	vw10	clv04	ecl4	yst4
<i>Accuracy</i>	66.46	99.17	89.97	68.49	50.00	95.57	83.91	72.29
<i>GM</i>	60.50	99.16	89.93	63.19	0.00	95.50	83.90	72.07
<i>TP_{rate}</i>	69.68	98.33	88.62	73.31	0.00	94.01	83.78	73.79
<i>TN_{rate}</i>	63.26	100.00	91.31	63.68	100.00	97.14	84.04	70.80

We compared the obtained results with the results demonstrated in Table 2, where fuzzy classifiers were constructed on complete sets of imbalanced data and optimized by the continuous GSA. The Wilcoxon's criterion values for the third stage are presented in Table 10.

Table 10. Comparison of fuzzy classification results with and without preprocessing.

Metrics	Algorithms	STS	<i>p</i>	NH
Accuracy	SMOTE - GSA ($\gamma = 0$)	-1.96	0.05	Reject
	SMOTE - GSA ($\gamma = 1$)	-1.96	0.05	Reject
	SMOTE - GSA ($\gamma = 0.5$)	-1.96	0.05	Reject
GM	SMOTE - GSA ($\gamma = 0$)	0.84	0.401	Retain
	SMOTE - GSA ($\gamma = 1$)	-1.4	0.161	Retain
	SMOTE - GSA ($\gamma = 0.5$)	-0.84	0.401	Retain
TPrate	SMOTE - GSA ($\gamma = 0$)	1.4	0.161	Retain
	SMOTE - GSA ($\gamma = 1$)	-0.14	0.889	Retain
	SMOTE - GSA ($\gamma = 0.5$)	0.84	0.401	Retain
TNrate	SMOTE - GSA ($\gamma = 0$)	-1.26	0.208	Retain
	SMOTE - GSA ($\gamma = 1$)	-0.84	0.401	Retain
	SMOTE - GSA ($\gamma = 0.5$)	-1.12	0.263	Retain

The comparison shows that fuzzy classifiers constructed on the original datasets and tuned by GSA_c in relation to fuzzy classifiers built on oversampled data demonstrate better overall accuracy with comparable recognition quality of a positive class. Therefore, if for the classification task it is important not only to classify the positive class correctly, but also not to receive large losses in the recognition of a negative class, then a fuzzy classifier with parameter tuning with the GSA_c is a more preferable tool.

At the next stage of comparison, the feature selection was carried out on the oversampled data. Table 11 presents the results of fuzzy classification averaged over five samples on subsets of features obtained by the random search algorithm.

Table 11. Results of fuzzy classifier construction on features selected after using the SMOTE algorithm.

Metrics	vhc0	nth2	sgm0	pb10	vw10	clv04	ecl4	yst4
<i>F</i> .	8.40	3.00	8.80	1.20	5.60	1.00	6.20	2.00
<i>Acc</i> .	60.51	97.78	86.00	62.05	49.38	90.98	86.52	50.70
<i>GM</i>	44.98	97.75	85.75	48.90	14.62	90.91	86.20	11.57
<i>TP</i> _{rate}	69.22	100.00	92.26	71.35	6.25	93.39	87.67	40.77
<i>TN</i> _{rate}	51.78	95.56	79.75	52.74	92.50	88.57	85.38	60.63

Table 12 presents the values of the performance indexes obtained after selecting features by the filter based on mutual information.

Table 12. Results of constructing fuzzy classifiers on subsets of features found by the filter after using the oversampling algorithm.

Metrics	vhc0	nth2	sgm0	pb10	vw10	clv04	ecl4	yst4
<i>F</i> .	8.60	1.80	12.00	4.00	6.00	5.00	5.20	5.00
<i>Acc</i> .	69.47	94.72	90.40	53.81	52.19	91.30	89.48	71.49
<i>GM</i>	66.65	94.55	90.34	27.49	9.84	90.86	89.44	68.40
<i>TP</i> _{rate}	72.75	89.44	89.08	62.83	5.00	86.75	88.78	71.49
<i>TN</i> _{rate}	66.20	100.00	91.72	44.78	99.38	95.87	90.18	71.50

We compared these values with the results of constructing fuzzy classifiers with feature selection and parameter tuning using the GSA on the initial datasets (Table 3). Table 13 shows the results of the comparison by the Wilcoxon test.

Table 13. Comparison of the results of constructing fuzzy classifiers on oversampled and origin data using the selection of features.

Metrics	Algorithm 1	Algorithm 2	Standardized Test Statistic	<i>p</i> -Value	Null Hypothesis
<i>Features</i>	SMOTE + RS	GSA ($\gamma = 0$)	−0.841	0.4	Retain
		GSA ($\gamma = 1$)	−0.631	0.528	Retain
		GSA ($\gamma = 0.5$)	−0.7	0.484	Retain
	SMOTE + MI	GSA ($\gamma = 0$)	0.983	0.326	Retain
		GSA ($\gamma = 1$)	0.771	0.441	Retain
		GSA ($\gamma = 0.5$)	0.84	0.401	Retain
<i>Accuracy</i>	SMOTE + RS	GSA ($\gamma = 0$)	−2.521	0.012	Reject
		GSA ($\gamma = 1$)	−2.521	0.012	Reject
		GSA ($\gamma = 0.5$)	−2.24	0.025	Reject
	SMOTE + MI	GSA ($\gamma = 0$)	−2.521	0.012	Reject
		GSA ($\gamma = 1$)	−2.521	0.012	Reject
		GSA ($\gamma = 0.5$)	−2.38	0.017	Reject
<i>GM</i>	SMOTE + RS	GSA ($\gamma = 0$)	−0.84	0.401	Retain
		GSA ($\gamma = 1$)	−1.823	0.068	Retain
		GSA ($\gamma = 0.5$)	−1.963	0.05	Reject
	SMOTE + MI	GSA ($\gamma = 0$)	−0.42	0.674	Retain
		GSA ($\gamma = 1$)	−1.82	0.069	Retain
		GSA ($\gamma = 0.5$)	−1.54	0.123	Retain
<i>TP_{rate}</i>	SMOTE + RS	GSA ($\gamma = 0$)	1.26	0.208	Retain
		GSA ($\gamma = 1$)	−0.98	0.327	Retain
		GSA ($\gamma = 0.5$)	0	1	Retain
	SMOTE + MI	GSA ($\gamma = 0$)	0.98	0.327	Retain
		GSA ($\gamma = 1$)	−0.84	0.401	Retain
		GSA ($\gamma = 0.5$)	0.14	0.889	Retain
<i>TN_{rate}</i>	SMOTE + RS	GSA ($\gamma = 0$)	−2.521	0.012	Reject
		GSA ($\gamma = 1$)	−2.521	0.012	Reject
		GSA ($\gamma = 0.5$)	−2.521	0.012	Reject
	SMOTE + MI	GSA ($\gamma = 0$)	−2.028	0.043	Reject
		GSA ($\gamma = 1$)	−1.68	0.093	Retain
		GSA ($\gamma = 0.5$)	−1.68	0.093	Retain

The results demonstrate that fuzzy classifiers optimized by the gravitational search algorithm show better results than fuzzy classifiers constructed on selected sets of features after data oversampling using the SMOTE.

The last stage of the comparison was to check the effectiveness of the fuzzy classifier using the GSA for selecting features and tuning parameters relative to the state-of-art classification algorithms. Using the open sklearn library, the following classifiers were built on complete data sets: Gaussian naive Bayes (GNB), logistic regression classifier (LR), decision tree classifier (DT), multilayer perceptron classifier (MLP), linear support vector classifier (LSV), K-nearest neighbors classifier with $k = 3$ (3NN), AdaBoost classifier (AB), random forest classifier (RF), and gradient boosting for classification (GB) [42]. All algorithm parameters were used by default.

Table 14 contains the results of constructing various classifiers on selected data sets. The last three columns show the fuzzy classifiers from Table 4.

The obtained values were compared using the criterion of Wilcoxon's sign ranks for linked samples (Tables A1–A4). The fuzzy classifier demonstrates results comparable with analogues in terms of the overall accuracy and the geometric mean but has fewer features. It shows the best results for the TP_{rate} value when the coefficient γ is equal to one. With the coefficient γ is equal to 0.5, the fuzzy classifier shows statistically comparable results with analogues by the value of TP_{rate} and yields only to three algorithms by the value of TN_{rate}.

Table 14. The results of constructing various classification algorithms on imbalanced datasets.

Data Sets	Classification Algorithms									Fuzzy Classifiers		
	GNB	LR	DT	MLP	LSV	3NN	AB	RF	GB	$\gamma = 0$	$\gamma = 1$	$\gamma = 0.5$
vhc0												
Acc.	64.9	96.6	93.6	98.1	96.8	94.8	96.2	95.6	96.5	85.1	78.3	84.0
GM	70.7	95.6	91.7	97.4	96.0	92.3	95.6	94.1	95.0	66.9	82.9	80.6
TP _{rate}	85.4	94.0	88.4	96.0	94.5	87.9	94.5	91.5	92.5	46.4	93.3	75.9
TN _{rate}	58.6	97.4	95.2	98.8	97.5	96.9	96.8	96.9	97.7	97.0	73.6	86.5
nth2												
Acc.	96.3	98.1	95.8	97.7	98.1	98.1	99.1	96.7	97.2	99.5	99.8	99.5
GM	96.5	95.3	91.1	95.0	96.5	95.1	98.2	91.5	91.6	98.5	99.5	99.3
TP _{rate}	97.1	91.4	85.7	91.4	94.3	91.4	97.1	85.7	85.7	97.1	99.0	99.0
TN _{rate}	96.1	99.4	97.8	98.9	98.9	99.4	99.4	98.9	99.4	100.0	100.0	99.6
sgm0												
Acc.	83.3	99.7	99.2	99.7	96.8	99.3	99.6	99.4	99.3	98.9	99.1	99.0
GM	89.2	99.3	98.4	98.4	97.6	99.1	99.1	98.5	98.3	98.2	99.1	98.7
TP _{rate}	98.5	98.8	97.3	99.1	99.1	98.8	98.5	97.3	97.0	97.3	99.1	98.2
TN _{rate}	80.7	99.9	99.5	99.8	96.4	99.4	99.7	99.8	99.7	99.2	99.1	99.2
pbl0												
Acc.	88.7	94.1	95.3	95.4	93.9	95.3	94.3	95.7	96.5	94.8	91.8	92.9
GM	65.4	74.9	85.1	86.1	71.8	83.4	84.1	86.3	87.7	77.1	84.9	81.5
TP _{rate}	47.4	58.1	74.6	76.0	53.3	71.2	73.5	76.4	79.2	60.3	77.2	70.6
TN _{rate}	93.4	98.1	97.7	97.6	98.5	98.1	96.6	97.9	98.4	98.7	93.4	95.4
vw10												
Acc.	93.7	91.2	95.2	94.7	89.3	94.4	96.2	95.5	97.0	96.4	98.2	97.7
GM	87.3	71.0	86.2	80.5	65.8	78.3	81.7	78.5	84.2	82.8	98.1	97.4
TP _{rate}	81.1	58.9	77.8	73.3	55.6	63.3	68.9	63.3	74.4	70.0	98.1	97.0
TN _{rate}	95.0	94.4	97.0	96.9	92.7	97.6	98.9	98.8	99.2	99.0	98.2	97.8
clv04												
Acc.	87.9	95.4	91.3	95.9	92.5	95.9	93.1	93.1	93.0	94.7	93.0	93.0
GM	84.9	82.0	60.1	80.2	60.8	80.2	55.2	37.0	45.5	55.0	85.2	86.2
TP _{rate}	84.6	69.2	46.2	69.2	46.2	69.2	38.5	23.1	53.8	41.0	76.9	82.1
TN _{rate}	88.1	97.5	95.0	98.1	96.3	98.1	97.5	98.8	96.3	99.0	94.3	93.9
ecl4												
Acc.	81.2	93.4	94.6	94.0	94.0	93.4	95.8	96.7	96.7	98.7	96.1	97.9
GM	83.9	85.7	74.6	83.6	88.6	85.7	81.8	78.2	84.5	88.9	90.1	93.9
TP _{rate}	95.0	80.0	60.0	75.0	85.0	80.0	70.0	65.0	75.0	80.0	85.0	90.0
TN _{rate}	80.4	94.3	96.8	95.3	94.6	94.3	97.5	98.7	98.1	99.9	96.8	98.4
yst4												
Acc.	16.0	96.6	96.0	95.3	96.5	96.8	96.4	96.0	96.4	95.6	84.1	91.2
GM	34.6	30.1	54.7	44.7	6.3	47.5	47.2	30.1	45.1	19.7	79.9	80.4
TP _{rate}	96.1	11.8	31.4	27.5	2.0	23.5	23.5	11.8	23.5	9.8	76.5	70.6
TN _{rate}	13.1	99.7	98.3	97.7	99.9	99.4	99.0	99.0	99.0	98.7	84.3	91.9

Thus, if the chosen priority coefficient γ is zero, the proposed metric represents the overall accuracy. Then the classifier focuses on recognizing a negative class, and as a result, the model has a low value of the Type I error, but a high value of the Type II error.

In the case when γ is equal to 1, the function will be identical to the geometric mean. Then, the efficiency of the fuzzy classifier with respect to the positive class will increase. As a result, the Type II error will decrease, but the Type I error can increase significantly.

When using coefficient γ close to 0.5, a system with low values of both errors will be obtained simultaneously. The proposed metric can be useful for such data as vowel0, ecoli4, and yeast4, when a high-quality classification of one class can lead to large losses in the ability of the model to recognize another class.

6. Conclusions

We considered the possibility of applying the gravitational search algorithm to improve the efficiency of the fuzzy classifier in the presence of data imbalance. The binary GSA reduced the space of input features by selecting informative feature subsets in the wrapper mode for a fuzzy classifier. The continuous GSA helped to improve the quality of classification. We proposed a new metric that could

influence the final performance indexes of the model by choosing the priority coefficient. The function with the ability to change priority between the number of correctly defined positive and negative classes allowed the developer to flexibly configure the fuzzy classifier. In future works, we plan to further study the impact of the coefficient in the metric on the result and make proposals about the recommended value of the coefficient for certain characteristics of the dataset.

Author Contributions: Conceptualization and methodology, M.B. and I.H.; software, M.B.; validation, I.H. and M.B.; investigation, K.S. and I.H.; writing, I.H. and M.B.; writing—review and editing, M.B., I.H., and A.K.; supervision, A.K. and A.S.; project administration, A.K. and A.S.; funding acquisition, A.S.

Funding: This research was supported by the Ministry of Education and Science of Russian Federation, Government Order no. 2.8172.2017/8.9 (TUSUR).

Conflicts of Interest: The authors declare no conflict of interest. The sponsors had no role in the design, execution, interpretation, or writing of the study.

Appendix A

Table A1. The results of the comparison of various classification algorithms with fuzzy classifiers optimized with the gravitational search algorithm by the value of the overall accuracy.

Algorithms	FC, $\gamma = 0$			FC, $\gamma = 1$			FC, $\gamma = 0.5$		
	STS	p	NH	STS	p	NH	STS	p	NH
GNB	−2.521	0.012	Reject	−2.521	0.012	Reject	−2.521	0.012	Reject
LR	−0.21	0.833	Retain	0.7	0.484	Retain	0.42	0.674	Retain
DT	−0.56	0.575	Retain	0.42	0.674	Retain	0.14	0.889	Retain
MLP	−0.14	0.889	Retain	1.122	0.262	Retain	0.7	0.484	Retain
LSV	−1.12	0.263	Retain	0.14	0.889	Retain	−0.28	0.779	Retain
3NN	0	1	Retain	0.98	0.327	Retain	0.7	0.484	Retain
AB	−0.14	0.889	Retain	0.98	0.327	Retain	0.7	0.484	Retain
RF	−0.491	0.624	Retain	1.26	0.208	Retain	0.771	0.441	Retain
GB	0.07	0.944	Retain	1.183	0.237	Retain	0.845	0.398	Retain

Table A2. The results of the comparison of various classification algorithms with fuzzy classifiers optimized with the gravitational search algorithm by the value of the geometric mean accuracy of each class.

Algorithms	FC, $\gamma = 0$			FC, $\gamma = 1$			FC, $\gamma = 0.5$		
	STS	p	NH	STS	p	NH	STS	p	NH
GNB	0.28	0.779	Retain	−2.521	0.012	Reject	−2.521	0.012	Reject
LR	0.421	0.674	Retain	−1.54	0.123	Retain	−1.54	0.123	Retain
DT	1.12	0.263	Retain	−1.82	0.069	Retain	−1.68	0.093	Retain
MLP	1.26	0.208	Retain	−1.4	0.161	Retain	−1.26	0.208	Retain
LSV	−0.7	0.484	Retain	−1.963	0.05	Reject	−1.82	0.069	Retain
3NN	1.26	0.208	Retain	−1.521	0.128	Retain	−1.26	0.208	Retain
AB	0.84	0.401	Retain	−1.69	0.091	Retain	−1.26	0.208	Retain
RF	0	1	Retain	−1.68	0.093	Retain	−1.68	0.093	Retain
GB	0.84	0.401	Retain	−1.54	0.123	Retain	−1.4	0.161	Retain

Table A3. The results of the comparison of various classification algorithms with fuzzy classifiers optimized with the gravitational search algorithm by the value of the true positive rate.

Algorithms	FC, $\gamma = 0$			FC, $\gamma = 1$			FC, $\gamma = 0.5$		
	STS	p	NH	STS	p	NH	STS	p	NH
GNB	1.859	0.063	Retain	-0.42	0.674	Retain	0.42	0.674	Retain
LR	0.338	0.735	Retain	-2.24	0.025	Reject	-1.54	0.123	Retain
DT	1.014	0.31	Retain	-2.521	0.012	Reject	-1.82	0.069	Retain
MLP	1.54	0.123	Retain	-2.028	0.043	Reject	-1.26	0.208	Retain
LSV	-0.28	0.779	Retain	-1.992	0.046	Reject	-1.68	0.093	Retain
3NN	1.521	0.128	Retain	-2.521	0.012	Reject	-1.4	0.161	Retain
AB	1.014	0.31	Retain	-2.24	0.025	Reject	-1.4	0.161	Retain
RF	-0.169	0.866	Retain	-2.524	0.012	Reject	-1.68	0.093	Retain
GB	1.4	0.161	Retain	-2.24	0.025	Reject	-1.54	0.123	Retain

Table A4. The results of the comparison of various classification algorithms with fuzzy classifiers optimized with the gravitational search algorithm by the value of the true negative rate.

Algorithms	FC, $\gamma = 0$			FC, $\gamma = 1$			FC, $\gamma = 0.5$		
	STS	p	NH	STS	p	NH	STS	p	NH
GNB	-2.521	0.012	Reject	-2.366	0.018	Reject	-2.521	0.012	Reject
LR	-1.12	0.263	Retain	1.26	0.208	Retain	0.98	0.327	Retain
DT	-2.38	0.017	Reject	1.183	0.237	Retain	0.98	0.327	Retain
MLP	-1.54	0.123	Retain	1.26	0.208	Retain	1.12	0.263	Retain
LSV	-1.68	0.093	Retain	0.56	0.575	Retain	0.42	0.674	Retain
3NN	-1.54	0.123	Retain	1.26	0.208	Retain	1.332	0.183	Retain
AB	-1.54	0.123	Retain	2.383	0.017	Reject	1.96	0.05	Reject
RF	-1.262	0.207	Retain	2.103	0.035	Reject	2.1	0.036	Reject
GB	-0.631	0.528	Retain	2.38	0.017	Reject	2.1	0.036	Reject

References

- Peng, L.; Zhang, H.; Yang, B.; Chen, Y. A new approach for imbalanced data classification based on data gravitation. *Inf. Sci.* **2014**, *288*, 347–373. [[CrossRef](#)]
- Special Issue on Recent advances in Theory, Methodology and Applications of Imbalanced Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 763. [[CrossRef](#)]
- He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Know. Data Eng.* **2009**, *21*, 1263–1284. [[CrossRef](#)]
- Ali, A.; Shamsuddin, S.M.; Ralescu, A. Classification with class imbalance problem: A review. *Int. J. Adv. Soft Comput. Appl.* **2013**, *5*, 1–30.
- Mathew, J.; Pang, C.K.; Luo, M.; Leong, W.H. Classification of Imbalanced Data by Oversampling in Kernel Space of Support Vector Machines. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4065–4076. [[CrossRef](#)] [[PubMed](#)]
- Bardamova, M.; Konev, A.; Hodashinsky, I.; Shelupanov, A. A Fuzzy Classifier with Feature Selection Based on the Gravitational Search Algorithm. *Symmetry* **2018**, *10*, 609. [[CrossRef](#)]
- He, H.; Ma, Y. (Eds.) *Imbalanced Learning: Foundations, Algorithms, and Applications*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2013; p. 216.
- Hand, D. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Mach. Learn.* **2009**, *77*, 103–123. [[CrossRef](#)]
- Ferri, C.; Haernandez-Orallo, J.; Modroiou, R. An experimental comparison of performance measures for classification. *Pattern Recognit. Lett.* **2009**, *30*, 27–38. [[CrossRef](#)]
- Fernandez, J.C.; Carbonero, M.; Gutierrez, P.A.; Hervas-Martinez, C. Multi-objective evolutionary optimization using the relationship between F1 and accuracy metrics in classification tasks. *Appl. Intell.* **2019**, *49*, 3447–3463. [[CrossRef](#)]

11. Lopez, V.; Fernandez, A.; Garcia, S.; Paladec, V.; Herrera, F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* **2013**, *250*, 113–141. [[CrossRef](#)]
12. Lopez, V.; del Rio, S.; Benitez, J.M.; Herrera, F. Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Set Syst.* **2015**, *258*, 5–38. [[CrossRef](#)]
13. Vluymans, S.; Tarrago, D.S.; Saeys, Y.; Cornelis, C.; Herrera, F. Fuzzy rough classifiers for class imbalanced multi-instance data. *Pattern Recognit.* **2016**, *53*, 36–45. [[CrossRef](#)]
14. Fernández, A.; Del Jesus, M.J.; Herrera, F. Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced datasets. *Int. J. Approx. Reason.* **2009**, *50*, 561–577. [[CrossRef](#)]
15. Villarino, G.; Gómez, D.; Rodríguez, J.T.; Montero, J. A bipolar knowledge representation model to improve supervised fuzzy classification algorithms. *Soft Comput.* **2018**, *22*, 5121–5146. [[CrossRef](#)]
16. Haixiang, G.; Li, Y.; Shang, J.; Mingyun, G.; Yuanyue, H.; Gong, B. Learning from class-imbalanced data: Review of methods and application. *Expert Syst. Appl.* **2017**, *73*, 220–239. [[CrossRef](#)]
17. 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](#)]
18. Liu, G.; Yang, Y.; Li, B. Fuzzy rule-based oversampling technique for imbalanced and incomplete data learning. *Knowl. Based Syst.* **2018**, *158*, 154–174. [[CrossRef](#)]
19. D’Addabbo, A.; Maglietta, R. Parallel selective sampling method for imbalanced and large data classification. *Pattern Recognit. Lett.* **2015**, *62*, 61–67. [[CrossRef](#)]
20. Diez-Pastor, J.F.; Rodriguez, 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](#)]
21. Saez, 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](#)]
22. Lin, W.-C.; Tsai, C.-F.; Hu, Y.-H.; Jhang, J.-S. Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **2017**, *409–410*, 17–26. [[CrossRef](#)]
23. Ofek, N.; Rokach, L.; Stern, R.; Shabtai, A. Fast-CBUS: a fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing* **2017**, *243*, 88–102. [[CrossRef](#)]
24. Diao, R. Feature Selection with Harmony Search and Its Applications. Available online: https://www.researchgate.net/publication/283652269_Feature_selection_with_harmony_search_and_its_applications (accessed on 10 March 2019).
25. Witten, I.H.; Frank, E. *Data Mining Practical Machine Learning Tools and Techniques*, 2nd ed.; Morgan Kaufmann: Amsterdam, The Netherlands, 2011; 558p.
26. Liu, H.; Yu, L. Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502.
27. Senthamarai Kannan, S.; Ramaraj, N. A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowl. Based Syst.* **2010**, *23*, 580–585. [[CrossRef](#)]
28. Bonnländer, B.; Weigend, A. Selecting input variables using mutual information and nonparametric density estimation. *Int. Symp. Artif. Neural Netw.* **1994**, *49*, 42–50.
29. Du, L.; Xu, Y.; Zhu, H. Feature Selection for Multi-Class Imbalanced Data Sets Based on Genetic Algorithm. *Ann. Data Sci.* **2015**, *2*, 293–300. [[CrossRef](#)]
30. Hernandez, J.C.H.; Duval, B.; Hao, J.-K. A genetic embedded approach for gene selection and classification of microarray data. In *Lecture Notes in Computer Science. Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. 5th European Conference, EvoBIO 2007, Valencia, Spain, 11–13 April 2007*; Marchiori, E., Moore, J.H., Rajapakse, J.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4447, pp. 90–101. [[CrossRef](#)]
31. Moayedikia, A.; Ong, K.-L.; Boo, Y.L.; Yeoh, W.G.S.; Jensen, R. Feature selection for high dimensional imbalanced class data using harmony search. *Eng. Appl. Artif. Intell.* **2017**, *57*, 38–49. [[CrossRef](#)]
32. Hodashinsky, I.; Sarin, K. Feature Selection for Classification through Population Random Search with Memory. *Autom. Remote Control* **2019**, *80*, 324–333. [[CrossRef](#)]
33. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]

34. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. BGSa: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [[CrossRef](#)]
35. Özkaraca, O.; Keçebaş, A. Performance analysis and optimization for maximum exergy efficiency of a geothermal power plant using gravitational search algorithm. *Energy Convers. Manag.* **2019**, *185*, 155–168. [[CrossRef](#)]
36. Ma, C.; Jiang, Y.; Li, T. Gravitational Search Algorithm for Microseismic Source Location in Tunneling: Performance Analysis and Engineering Case Study. *Rock Mech. Rock Eng.* **2019**, 1–18. [[CrossRef](#)]
37. Mahanipour, A.; Nezamabadi-pour, H. GSP: an automatic programming technique with gravitational search algorithm. *Appl. Intell.* **2019**, *49*, 1502–1516. [[CrossRef](#)]
38. Mahanipour, A.; Nezamabadi-pour, H. A multiple feature construction method based on gravitational search algorithm. *Expert Syst. Appl.* **2019**, *127*, 199–209. [[CrossRef](#)]
39. Pelusi, D.; Mascella, R.; Tallini, L.; Nayak, J.; Naik, B.; Abraham, A. Neural network and fuzzy system for the tuning of Gravitational Search Algorithm parameters. *Expert Syst. Appl.* **2018**, *102*, 234–244. [[CrossRef](#)]
40. Knowledge Extraction Based on Evolutionary Learning. Available online: <http://keel.es> (accessed on 10 May 2019).
41. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **2017**, *18*, 1–5.
42. Scikit-learn. User Guide. Supervised Learning. Available online: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning (accessed on 13 August 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).