A Preliminary Study on Overlapping and Data Fracture in Imbalanced Domains by means of Genetic Programming-based Feature Extraction

Jose G. Moreno-Torres, Francisco Herrera
Department of Computer Science and Artificial Intelligence
Universidad de Granada, 18071 Granada, Spain.
(jose.garcia.mt,herrera)@decsai.ugr.es

Abstract—**The classification of imbalanced data is a well-studied topic in data mining. However, there is still a lack of understanding of the factors that make the problem difficult. In this work, we study the two main reasons that make the classification of imbalanced datasets complex: overlapping and data fracture. We present a Genetic Programming-based feature extraction method driven by Rough Set Theory to help visualize the data in a bidimensional graph, to better understand how the presence of overlapping and data fractures affect classification performance.**

***Keywords*-imbalanced data; overlapping; data fracture; feature extraction; genetic programming; rough set theory;**

I. Introduction

The classification of imbalanced data is a priority issue in the literature nowadays [1], [2]. Most of the approaches presented are based on preprocessing the data, whether it is by oversampling the minority class or undersampling the majority one. Excellent accuracy results have been obtained, but there is still room to improve.

This contribution does not seek to better the classification performance obtained by existing proposals, but rather to analyze the two main factors where the real complexity of the problems lies:

- Overlapping: The examples of the minority class share a region with the majority one, where all the examples are intertwined. This is a problem intrinsic to the data. This issue has been studied in [3], [4].
- Data Fracture: There is a change in data distribution between the training and test sets, often in the minority class. The incidence of this issue depends on the partitioning of the data. The problem of data fracture (or dataset shift, as some authors call it) is relatively new [5], [6], [7], [8], and we are not aware of any studies regarding imbalanced datasets published so far.

To help perform a visual analysis of the data, we propose the 'Genetic Programming-based feature extraction using Rough Set Theory' algorithm (GP-RST), which is based on the application of the GP paradigm [9] as a feature extraction tool, using RST [10] techniques to estimate the fitness of individuals. It obtains a transformation from the original feature space into a bidimensional one where the classes are as separated as possible; serving both as a visualization tool and as a competitive preprocessing technique for imbalanced datasets. GP-RST is more suitable for the visualization of imbalanced domains than other feature extraction techniques because the fitness is calculated for each class and then aggregated, being therefore independent of the class imbalance in the training set.

The application of GP-RST has permitted the discovery of three possible situations, which are all easily visualizable in the bidimensional feature space it extracts:

1 The dataset presents a low amount of overlapping and data fracture, resulting in a good behavior both in terms of training and test classifier performance.
2 The dataset presents a high amount of overlapping, resulting in a poor classifier performance both in training and test.
3 There is a significant amount of data fracture, which produces an overfitting issue leading to a big gap between training and test set performance.

This contribution is organized as follows: We begin with some notation specifications in section II. In section III we briefly introduce the relevant concepts of RST. Section IV includes a description of the GP-RST algorithm, while section V shows the experimental procedure and classifier performance results. Section VI presents a visual analysis in terms of overlap and data fracture. Lastly, some concluding remarks are made in section VII.

II. Notation

A classification problem is considered with:

- A set of input attributes $A = \{a_i / i = 1, ..., n_v\}$, where n_v is the number of features of the problem.
- A set of values for the target variable (class) $C = \{C_j / j = \{1, \cdots, n_c\}\}$, where n_c is the number of different values for the class variable.
- A set of examples $E = \{e^h = (e^h_1, ..., e^h_{n_v}, C^h) / h = 1, \cdots, n_e\}$, where C^h is the class label for the example e^h, and n_e is the number of examples.
- The range of a variable i is defined as $range_i = (e^m_i) - (e^n_i) / \forall h \colon (e^m_i >= e^h_i \ \& \ e^n_i <= e^h_i)$.
- The number of examples of class C_j in E is noted as $n_e^{C_j}$.

When applying GP-RST to obtain new features,

- The set of new features is noted as $Y = \{y_1, y_2\}$,
- The new features are functional mappings of A, represented as
$$Y = \{f_1(A), f_2(A)/f_i(A) = f_i(a_1, \cdots, a_{n_v})\}$$
- The result of applying a function f_i to a sample e^h is denoted as $f_i(e^h) = f_i(e_1^h, \cdots, e_{n_v}^h)$.
- $E^{'}$ results of applying f_1, f_2 to a set of examples E:
$$E^{'} = \{e^{'h} = (f_1(e^h), f_2(e^h), C^h)/h = 1, \cdots, n_e\}$$

III. Introduction to Rough Set Theory

This section includes the definition of the RST concepts that are relevant to this work. For an in-depth study of the topic, see [10].

- Information System and Decision System: Let a set of attributes $A = \{a_1, a_2, \ldots, a_{n_v}\}$ and a non-empty, finite set called the universe U, with instances described using the attributes a_i; Information System is the name given to the pair (U, A). If a new attribute d called decision is attached to each element of U, indicating the decision made in that state or situation, then a Decision System is created $(U, A \cup \{d\})$, where $d \notin A$ is the decision attribute.
- The attribute of decision d induces a partition of the object universe U. Let a set of integer numbers$\{1, \ldots, l\}$, $X_i = \{x \in U : d\,(x) = i\}$, then $\{X_1, \ldots X_l\}$ is a collection of equivalence classes, called decision classes, where two objects belong to the same class if they have the same decision attribute value. In the case of this contribution, d corresponds to the class variable.
- The novelty of the RST are the lower and upper approximations of a subset $X \subseteq U$. These concepts were originally introduced in reference to an indiscernibility relation R. In classical RST, R is defined as an equivalence relation. This approach is extended by accepting that objects that are not indiscernible but sufficiently close or similar can be grouped into the same class. The aim is to construct a similarity relation $R^{'}$ from the indiscernibility relation R by relaxing the original conditions for indiscernibility.
- The similarity relation used in this work is defined as
$$R^{'}(x, y) = \begin{cases} 1 & \forall i(|x_i - y_i| < 0.1 * range_i) \\ 0 & otherwise \end{cases} \quad (1)$$
- The approximation of the set $X \subset U$, using the similarity relation $R^{'}$, has been induced as a pair of sets called lower approximation of X and upper approximation of X. The lower approximation $B_*(X)$ of X is defined as shown in equation 2.
$$B_*(X) = \{x \in X : R^{'}(x) \subseteq X\} \quad (2)$$
- Within RST, the meaning of the lower approximation of a decision system is of great interest for the analysis of new feature spaces. It consists of the objects that with absolute certainty belong to one class or another, guaranteeing that these instances are free of noise.
- Taking into account the equation defined in 2, the quality of the approximation of X is defined for the relation $R^{'}$ as:
$$\gamma(X) = \frac{|B_*(X)|}{|X|} \quad (3)$$

IV. A Genetic Programming-based feature extraction method driven by Rough Set Theory(GP-RST)

In this section we first present a formal expression of the problem at hand in subsection IV-A, followed by a general description of the GP-RST method in subsection IV-B, and we finish with a detailed explanation of the fitness calculation procedure in subsection IV-C.

A. Formal definition of the problem

The problem we are attempting to solve is, given a classification problem with a set of attributes A, and a set of examples E, obtain $f_1(A)$ and $f_2(A)$ such that $fitness(f_1(E), f_2(E))$ is maximized. The fitness calculation is based on the estimation of the separability between the classes through the maximization of the quality of approximation (Equation 3) for each class.

B. General description of GP-RST

Genetic Programming is an evolutionary computation technique that evolves expressions defined by a context-free grammar, by generating a starting population and applying crossover and mutation operators over it repeatedly, selecting on each generation the best potential solutions (expressions) according to a given fitness evaluation formula.

The GP-RST algorithm is a simple extension of a standard GP procedure with the following tweaks:

- It simultaneously evolves two trees, one for each dimension in the new feature space.
- It uses $\{x_1, ..., x_{n_v}, e\}$ as its terminal set, effectively evolving functional mappings of X.
- It uses $\{+, -, \times, \div\}$ as its function set.

C. Fitness evaluation

The fitness evaluation procedure, as has been expressed before, is based upon RST, more specifically it is associated to the quality of approximation of each of the classes.

V. Experimental Framework and Results

This section begins with a general description of the experimental procedure, followed by an enumeration of the datasets used in subsection V-A, then the specific parameters chosen for the experimentation can be seen in subsection

Algorithm 1 Fitness evaluation procedure

```
1. Obtain $E' = \{e'^h = (f_1(e^h), f_2(e^h), C^h)/h = 1, ..., n_e\}$,
   where $f_1$ and $f_2$ are the expressions encoded on
   each of the trees of the individual being evaluated.
2. For each class label $C_i \in C : i = 1, ..., n_c$,
      2.1 Build a rough set $X_i$ containing all the elements of
          class $C_i$.
      2.2 Calculate the lower approximation of $X_i$, $B_*(X_i)$.
      2.3 The fitness of the chromosome for class $C_i$ is
          estimated as the quality of the approximation over
          $X_i$, $\gamma(X_i)$.
3. The fitness of the chromosome is the geometric mean of
   the ones obtained for each class:
   $fitness = \sqrt[n_c]{\prod_{i=1}^{n_c} \gamma(X_i)}$.
```

V-B. Finally, the classifier performance results are presented in subsection V-C.

The effectiveness of the preprocessing methods was measured in terms of classifier performance. Since the classical accuracy measures are not suitable to highly imbalanced domains, the performance was measured using the geometric mean of the accuracies per class [11]:

$$ClassifierPerformance = \sqrt{\frac{TP}{TP+FN} * \frac{TN}{TN+FP}} \tag{4}$$

where TP, TN, FP and FN stand for True Positives, True Negatives, False Positives and False Negatives respectively. The classifier used for all experiments was C4.5 [12], since it is a fast and efficient classifier that has been commonly used in the literature regarding imbalanced datasets. In any case, the choice of classifier does not have any influence in the visual analysis.

The testing procedure utilized was the standard in the literature, using a 5-fold cross validation technique where only the training set was used to do the preprocessing. We tested three different cases:

- The original dataset with no preprocessing, denoted as 'None'.
- The bidimensional dataset that results from applying GP-RST.
- SMOTE with ENN cleaning [13], a hybrid preprocessing method that first oversamples the minority class using SMOTE [14], and then cleans up the borders using the Edited Nearest Neighbor rule.

A schematic representation of the experimental procedure can be found in Figure 1. The GP implementation was based on the Open Beagle library [15], and we used the KEEL software [16] to carry out all the experiments and the statistical tests.

A. Datasets

The datasets used in this study were obtained from the KEEL dataset repository [17], which are in turn variations of well known UCI datasets [18]. Table I presents the datasets, detailing the number of samples n_s, number of variables n_v and Imbalance Ratio (IR), which is calculated as $\frac{n_s^{majorityClass}}{n_s^{minorityClass}}$. All the datasets used are binary classification problems, but the GP-RST algorithm is capable of working with multiclass problems without any modifications.

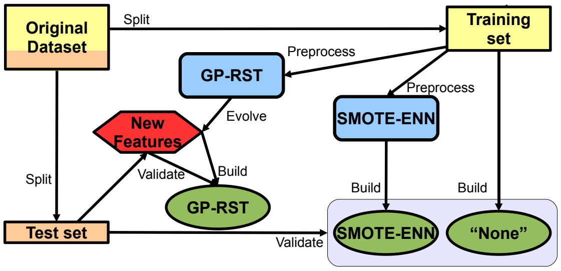


Figure 1. Schematic representation of the experimental procedure

Table I
DATASETS USED FOR THE EXPERIMENTAL STUDY

Dataset	IR	n_v	n_s
ecoli_0137v26	39.14	7	281
yeast_2v8	23.10	8	482
glass_5	22.78	9	214
shuttle_2v4	20.50	9	129
glass_016v5	19.44	9	184
pageblocks_13v4	15.86	10	472
ecoli_4	15.80	7	336
glass_4	15.46	9	214
yeast_1v7	14.30	7	459
glass_2	11.59	9	214
glass_016v2	10.29	9	192
yeast_2v4	9.08	8	516

B. Parameters

This subsection presents the parameter values chosen for the GP evolution. In this work, we decided not to squeeze the maximum performance from the method but to focus on the interpretation of the visual results, so most of the parameters were fixed to common default values.

The specific values for the parameters are presented in Table II.

C. Classifier performance results

This subsection presents the results obtained by the different preprocessed datasets, in terms of the test-set classifier performance (see Equation 4) obtained by the C4.5 classifier. They are shown in Table III.

To check whether the differences in performance are significant, we performed a statistical analysis of the results by means of a non-parametric test.

In [19], [20], [21], [22] a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers

Table II
EVOLUTIONARY PARAMETERS FOR THE GP-RST PROCEDURE

Parameter	Value
Number of trees	2
Population size	10000
Duration of the run	200 generations
Selection operator	Tournament, no replacement
Tournament size	3
Crossover operator	One-point crossover
Crossover probability	0.9
Mutation operator	Replacement & Swap
Replacement mutation prob	0.001
Swap mutation prob	0.01
Max depth, swapped-in subtree	5

Table III
CLASSIFIER PERFORMANCE RESULTS

	C4.5 performance		
Dataset	None	GP-RST	SMOTE-ENN
ecoli_0137v26	0.8436	0.8405	0.7462
yeast_2v8	0.2226	0.6635	0.7542
glass_5	0.8776	0.8798	0.9405
shuttle_2v4	0.9129	0.9877	1.0000
glass_016v5	0.7389	0.9320	0.9943
pageblocks_13v4	0.9989	0.9764	0.9989
ecoli_4	0.7985	0.8916	0.8563
glass_4	0.7228	0.8683	0.7746
yeast_1v7	0.5719	0.5464	0.4828
glass_2	0.2407	0.2394	0.6976
glass_016v2	0.0000	0.0000	0.5333
yeast_2v4	0.7921	0.7996	0.8770
Average	0.6434	0.7188	0.8046

are recommended. One of them is the Wilcoxon Signed-Ranks Test [23], [24], which is the test that we have selected to do the comparisons. A complete description of the Wilcoxon Signed-Ranks Test and other non-parametric tests for pairwise and multiple comparisons, together with software for their use, can be found in the website available at http://sci2s.ugr.es/sicidm/.

We evaluated the methods by performing all pairwise comparisons among them, including the option of not doing any preprocessing, denoted as 'None'. The results are presented in Table IV.

Table IV
WILCOXON SIGNED-RANKS TEST RESULTS

Comparison	R^+	R^-	p-value (two-tailed)
None v GP-RST	15	51	0.05372
None v SMOTE-ENN	13	53	0.0392
GP-RST v SMOTE-ENN	28	50	0.2005

From the results shown in Table IV, we can extract the following conclusions:

- Both GP-RST and SMOTE-ENN significantly outperform not doing anything.
- GP-RST performs slightly worse than SMOTE-ENN, but the difference is not statistically significant.

VI. GRAPHICAL ANALYSIS OF OVERLAPPING AND DATA FRACTURE

In this section we present a set of sample visualizations of the bidimensional datasets obtained by GP-RST.

A. *Good behavior*

Figure 2 shows a case where GP-RST succeeded in finding a bidimensional mapping of the original features in the ecoli4 dataset where both classes are easily separable in the training set, and such a separation generalizes well to the test set. This is the ideal case, one where a classifier performs very well, both in training and test.

B. *Overlap*

Figure 3 presents a case where, due to the complex overlap between classes in the original dataset, the GP-RST procedure was not successful in finding a bidimensional mapping where they were separable. The classifier performance on the preprocessed dataset was as bad as it was without preprocessing. This is the type of issue that was studied by [3], [4].

C. *Data fracture*

Figure 4 shows a case where partial success was achieved in the classification of the training set, but none of the examples in the test set belong to the area where the classes are separable.

This issue is the one we would like to raise awareness about. Even though most authors know about the overlap problem, the data fracture one is usually not considered, and needs to be taken into account when analyzing the performance of new methods in imbalanced domains.

VII. CONCLUSIONS

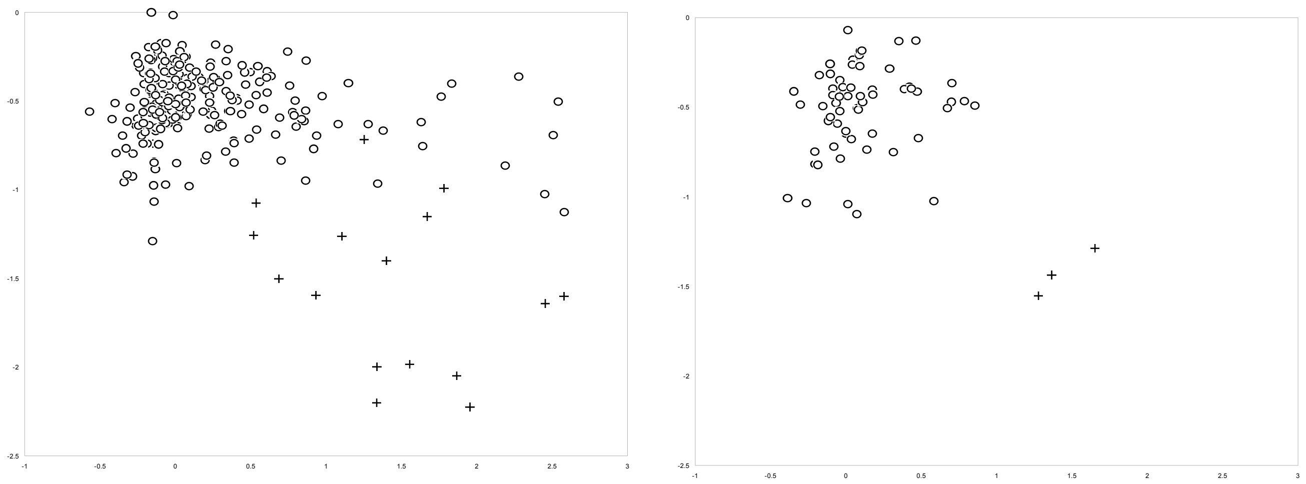
We have presented GP-RST, a GP-based feature extractor that employs RST techniques to estimate the fitness of individuals. We have shown GP-RST to be a competitive preprocessing method for highly imbalanced datasets, with the added advantage of providing bidimensional representations of the datasets it preprocesses, which are easily intepreted.

We have, through the analysis of the visual representations of the preprocessed datasets, observed a data fracture problem between training and test sets, specially in the minority class, that is affecting the classification performance.

We believe this discovery is very relevant since it challenges the usual assumptions when experimenting with preprocessing for highly imbalanced data. We intend to further study the issue, to test the hypothesis that data fracture is playing a major role in the complexity of classification in imbalanced domains.

ACKNOWLEDGMENTS

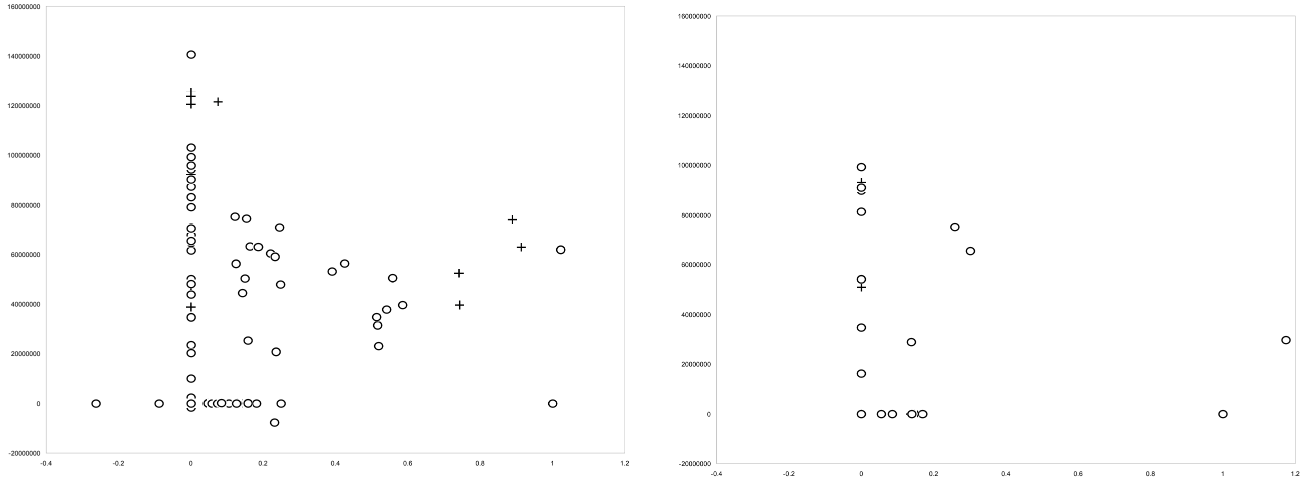
Jose García Moreno-Torres is currently supported by a FPU grant from the Ministerio de Educación y Ciencia of the Spanish Government.



(a) Training set (0.9682)

(b) Test set (0.9919)

Figure 2. Example of good behavior, dataset ecoli_4, 5^{th} partition. Classifier performance in parenthesis.



(a) Training set (0.3779)

(b) Test set (0.0000)

Figure 3. Example of bad behavior by overlap, dataset glass_016v2, 4^{th} partition. Classifier performance in parenthesis.

References

[1] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, September 2009.

[2] Y. M. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of Imbalanced Data: A Review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.

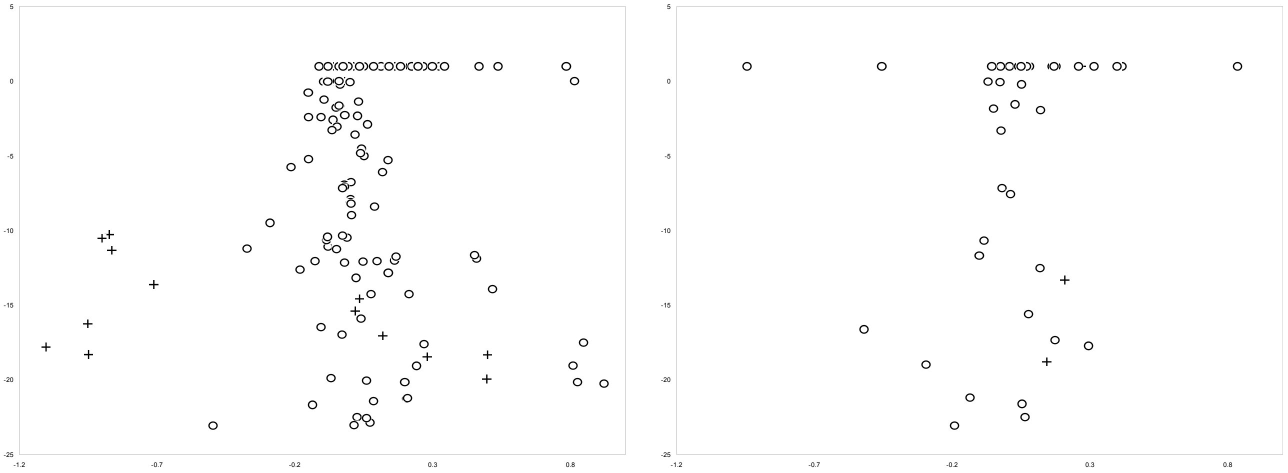
[3] V. García, R. A. Mollineda, and J. S. Sánchez, "On the k-NN performance in a challenging scenario of imbalance and overlapping," *Pattern Analysis & Applications*, vol. 11, no. 3-4, pp. 269–280, 2008.

[4] M. Denil and T. P. Trappenberg, "Overlap versus Imbalance," in *Canadian Conference on AI*, 2010, pp. 220–231.

[5] R. Alaiz-Rodríguez and N. Japkowicz, "Assessing the impact of changing environments on classifier performance," in *Canadian AI'08: Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 13–24.

[6] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[7] D. A. Cieslak and N. V. Chawla, "A framework for monitoring classifiers' performance: when and why failure occurs?" *Knowledge and Information Systems*, vol. 18, no. 1, pp. 83–108, 2009.



(a) Training set (0.6794)

(b) Test set (0.0000)

Figure 4. Another example of bad behavior by data fracture, dataset glass_2, 2^{nd} partition. Classifier performance in parenthesis.

[8] J. G. Moreno-Torres, X. Llorà, D. E. Goldberg, and R. Bhargava, "Repairing Fractures between Data using Genetic Programming-based Feature Extraction: A Case Study in Cancer Diagnosis," *Information Sciences, In Press*, 2010.

[9] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press, 1992.

[10] Z. Pawlak, *Rough Sets. Theoretical Aspects of Reasoning about Data*. Dordrecht: Kluwer Academics, 1991.

[11] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[12] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[13] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[15] C. Gagné and M. Parizeau, "Genericity in evolutionary computation software tools: Principles and case study," *International Journal on Artificial Intelligence Tools*, vol. 15, no. 2, pp. 173–194, 2006.

[16] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. D. Jesus, S. Ventura, J. M. Garrell, J. Otero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "Keel: A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 307–318, 2009.

[17] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," *Journal of Multiple-Valued Logic and Soft Computing, In Press*, 2010.

[18] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[19] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[20] S. García and F. Herrera, "An Extension on 'Statistical Comparisons of Classifiers over Multiple Data Sets' for all Pairwise Comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[21] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for Genetics-Based Machine Learning: Accuracy and Interpretability," *Soft Computing*, vol. 13, no. 10, pp. 959–977, 2009.

[22] ——, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[23] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[24] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures (4th Edition)*. Chapman & Hall/CRC, 2007.