Fuzzy Rule Based Classification Systems versus Crisp Robust Learners Trained in Presence of Class Noise's Effects: a Case of Study

José A. Sáez, Julián Luengo, Francisco Herrera
Department of Computer Science and Artificial Intelligence
University of Granada, CITIC-UGR
Granada, Spain, 18071
e-mail: {smja, julianlm, herrera}@decsai.ugr.es

***Abstract*—The presence of noise is common in any real-world dataset and may adversely affect the accuracy, construction time and complexity of the classifiers in this context. Traditionally, many algorithms have incorporated mechanisms to deal with noisy problems and reduce noise's effects on performance; they are called robust learners. The C4.5 crisp algorithm is a well-known example of this group of methods. On the other hand, models built by Fuzzy Rule Based Classification Systems are widely recognized for their robustness to imperfect data, but also for their interpretability.**

The aim of this contribution is to analyze the good behavior and robustness of Fuzzy Rule Based Classification Systems when noise is present in the examples' class labels, especially versus robust learners. In order to accomplish this study, a large number of datasets are created by introducing different levels of noise into the class labels in the training sets. We compare a Fuzzy Rule Based Classification System, the *Fuzzy Unordered Rule Induction Algorithm*, with respect to the C4.5 classic robust learner which is considered tolerant to noise. From the results obtained it is possible to observe that Fuzzy Rule Based Classification Systems have a good tolerance, in comparison to the C4.5 algorithm, to class noise.

***Keywords*-Noisy Data; Class Noise; Fuzzy Rule Based Systems; Robust Learners; Classification.**

I. Introduction

Fuzzy Rule Based Classification Systems (FRBCSs) [1], [2] are widely used due to their ability to build a linguistic model interpretable to the users with the possibility of mixing different information such as that proceeding from expert knowledge and information from mathematical models or empirical measures. Among the applications of FRBCSs we can find proposals in a variety of fields, including standard classification [3], [4], detection of intrusions [5] or medical applications [6].

One goal of classification algorithms is to form a generalization from a set of labeled training instances so that classification accuracy for previously unobserved instances is maximized. Hence the accuracy of the model created by any induction-based learning algorithm is determined by the quality of training data upon which this model is built. Data quality is determined by several components [7], among which are the source of that data and the input of the data, inherently subject to error. Thus, real-world datasets rarely lack these types of error and they usually have corruptions that can affect the interpretations, decisions taken and the models created from the data.

Therefore, the maximum achievable accuracy depends not only on the quality of the data, but also on the appropriateness of the chosen learning algorithm for the data. Knowing what kind of classification algorithms are more suitable when working with noisy data is a challenging question.

In this work we will analyze the suitability of FRBCSs, specifically we will focus on the *Fuzzy Unordered Rule Induction Algorithm* (FURIA) [4], when dealing with noise in examples' class labels and we will compare it to the C4.5 crisp algorithm [8] which is considered tolerant to noise and can be translated as a rule set. When training a classifier with problems with noise, the capability of this classifier to avoid the overfitting of the new characteristics introduced by the noisy examples is a key question [9]. Due to the inherent characteristics of fuzzy rules and the inference process of the FRBCSs that differ from those of the classic crisp systems, models obtained by FRBCSs are expected to absorb noise and work better than crisp interval rules used by robust learners such as C4.5. These characteristics enable the creation of a better generalization from the instances of the problem, since they better avoid the overfitting of noisy data and, therefore, obtain more robust and accurate models.

In order to carry out this comparison, we will consider 19 datasets from the KEEL-dataset repository [10]. Four different levels of noise are taken into account in the experimentation: 5%, 10%, 15% and 20%. Thus, 76 new synthetic datasets are created with class noise in the training sets. As we will consider two different types of class noise, the number of datasets created is doubled, for an experimentation with a total of 171 datasets. We will obtain the test accuracy of the models created with all the classification algorithms and we will use the Wilcoxon's statistical test [11] in order to check the significance of the differences found. We will propose a measure to quantify the degradation of the test accuracy of the models with the introduction of noise with respect to the original obtained without noise. We will also check the number of rules of each model in order to see how the size of the models is

978-1-4577-1675-1

affected by the noise.

The remainder of this paper is organized as follows. Section II presents an introduction to classification with noisy data. Next, Section III describes the FRBCS used in our work. Section IV shows the details of the experimental framework, which summarizes the datasets used, the validation scheme and the process to build the noisy datasets, along with the parameters used by the classification algorithms, and the scheme of comparisons. Section V includes the analysis and the experimental results obtained by the FURIA algorithm versus the C4.5 robust learner. Next, in Section VI we analyze the causes of the good behavior of FRBCSs when dealing with class noise. Finally, in Section VII we make some concluding remarks.

II. Classification with Noisy Data

Real-world data is never perfect and often suffers from corruptions that may harm interpretations of the data, models created and decisions made. In classification, noise can negatively affect the system performance in terms of classification accuracy, time in building, size and interpretability of the classifier built [12].

The quality of any dataset is determined by a large number of components as described in [7]. Some of these are the source of the data and the input of the data, which are inherently subject to error.

Class labels and attributes are two information sources which can influence the quality of a classification dataset. The quality of the class labels represents whether the class of each instance is correctly assigned; and the quality of the attributes indicates how well the attributes characterize instances for classification purposes.

Based on these two information sources which define the quality of a classification dataset we can distinguish two types of noise in a given dataset [13]: class noise and attribute noise.

1) Class noise or labeling errors occur when an instance belongs to the incorrect class. Class noise can be attributed to several causes, including subjectivity during the labeling process, data entry errors, or inadequacy of the information used to label each object. There are two possible types of class noise:
   - Contradictory examples: the same examples appear more than once and are labeled with different classes [14].
   - Misclassifications: instances are labeled with the wrong classes [15].
2) Attribute noise is used to refer to corruptions in the values of one or more attribute of instances in the dataset. Examples of attribute noise include: erroneous attribute values, missing or unknown attribute values, and incomplete attributes or "do not care" values.

The two most common approaches to noisy data in the literature are robust learners and noise preprocessing techniques:

- Robust learners are characterized by being less influenced by noisy data. An example of a robust learner is the C4.5 algorithm [8]. C4.5 uses pruning strategies to reduce the chances of trees being built with noise in the training data [16]. However, when the noise level becomes relatively high, even a robust learner may obtain a poor performance.
- Noise preprocessing techniques try to remove the negative impact of noise in the datasets prior to creating a model over the original data. Among these techniques, the most well-known methods are noise filtering ones. Their objective is to identify noisy instances which can be eliminated from the training data [17], [18].

In this contribution, we study mislabeled data as noise because it is very common in real-world data [12], [15]. These errors can be produced in situations where different classes have similar symptoms, as generally happens on the class boundaries. Furthermore, we compare the behavior of the FRBCS considered in our work with the well-known C4.5 robust learner. We want to verify that the effect of class noise on the accuracy and size of the models created by the FURIA algorithm is lower than on the models built by the C4.5 robust learner.

III. Fuzzy Rule Based Classification Systems

This section describes the basis of the fuzzy model that we have used in our study. First we introduce the basic notation that we will use later to describe the FRBCS. Next we describe the FURIA method in Subsection III-A.

Any classification problem consists of w training patterns $x_p = (x_{p1}, \dots, x_{pn})$, $p = 1, 2, \dots, w$, labeled with one of M possible classes $\mathbb{L} = \{\lambda_1, \dots, \lambda_M\}$, where x_{pi} is the i-th attribute value ($i = 1, 2, \dots, n$) of the p-th training pattern. In this paper, we use fuzzy rules with a single class and a rule weight associated to this class in the consequent [19]:

$$\begin{array}{c}\text{Rule } R_j : \texttt{IF } x_1 \texttt{ is } A_j^1 \texttt{ AND } \dots \texttt{ AND } x_n \texttt{ is } A_j^n \\ \texttt{THEN CLASS} = C_j \texttt{ WITH } RW_j\end{array} \quad (1)$$

where R_j is the label of the j-th rule, $x = (x_1, \dots, x_n)$ is an n-dimensional pattern vector, A_j^i is an antecedent fuzzy set, C_j is a class label and RW_j is the rule weight [20].

A. Fuzzy Unordered Rule Induction Algorithm

FURIA [4] builds upon the RIPPER interval rule induction algorithm [21]. The model built by FURIA uses fuzzy rules of the form given in Equation (1) where A_j^k is a fuzzy set $I^F = (\phi^{s,L}, \phi^{c,L}, \phi^{c,U}, \phi^{s,U})$ with a trapezoidal membership function

$$I^F(v) = \begin{cases} 1, & \text{if} \quad \phi^{c,L} \leq v \leq \phi^{c,U} \\ \frac{v-\phi^{s,L}}{\phi^{c,L}-\phi^{s,L}}, & \text{if} \quad \phi^{s,L} \leq v \leq \phi^{c,L} \\ \frac{\phi^{s,U}-v}{\phi^{s,U}-\phi^{c,U}}, & \text{if} \quad \phi^{c,U} \leq v \leq \phi^{s,U} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

and $C_j \in \mathbb{L} = \{\lambda_1, \ldots, \lambda_M\}$ is a class label. The rule weight RW_j of the rule R_j is computed as

$$RW_j = \frac{2\frac{|D_T^{(c)}|}{|D_T|} + \sum_{x \in D_T^{(c)}} \mu_{R_j}(x)}{2 + \sum_{x \in D_T} \mu_{R_j}(x)} \quad (3)$$

where D_T denotes the training set instances, $D_T^{(c)}$ denotes the subset of training instances with the label λ_c and $\mu_{R_j}(x) = \prod_{i=1\ldots n} I_i^F(x_i)$

To assign an output to a new example, suppose that fuzzy rules $R_1, \ldots, R_k$ have been learned for class λ_c. For a new query instance x, the support of this class is defined by

$$s_c(x) = \sum_{j=1}^{k} \mu_{R_j}(x) RW_j \quad (4)$$

The class predicted by FURIA is the one with maximal support. In the case of a tie, a decision in favor of the class with the highest frequency is made. When the query is not covered by any rule, a rule stretching method is proposed based on modifying the rules in a local way so as to make them applicable to the query. In order to do this it is checked the order in which the antecedents appear in the rule, and all premises from the first one that do not match the new instance are eliminated.

FURIA builds the fuzzy rule base by means of these two steps:

1) Learn a rule set for every single class λ_c of the problem, using a one-versus-all decomposition. In order to do this, the RIPPER algorithm is used, which consists of two fundamental steps described in [21]: the building and the optimization phase.
2) Obtain the fuzzy rules by means of fuzzifying the final rules from the above step. Each rule is fuzzified retaining the same structure as the original rule and replacing original intervals in the antecedent with fuzzy intervals. To fuzzify an interval, it is required to compute the four parameters needed for the trapezoidal fuzzy set from the original interval (complete procedure is described in [4]).

IV. Experimental Framework

In this section, we first describe the original datasets our experimentation is based on in Subsection IV-A. Then, in Subsection IV-B, the noise introduction process over the above mentioned original datasets and the class noise levels in order to create the final datasets are presented. Section IV-C indicates the parameters for the classification algorithms used for this work. Finally, Section IV-D establishes the comparison methodology carried out between the FRBCS and the robust learner considered.

A. Original Datasets

The experimentation has been based on 19 datasets taken from the KEEL-dataset repository[1] [10]. Table I summarizes the properties of the originally selected datasets. For each dataset, the number of instances (#Ins), the number of numeric attributes (#Att) along with the number of real and integer attributes (R/I) and the number of classes (#Cla) are presented.

Table I
Original Datasets Used from the KEEL-Dataset Repository

Dataset	#Ins	#Att (R/I)	#Cla	Dataset	#Ins	#Att (R/I)	#Cla
contraceptive	1,473	9 (0/9)	3	satimage	6,435	36 (0/36)	7
ecoli	336	7 (7/0)	8	segment	2,310	19 (19/0)	7
glass	214	9 (9/0)	7	sonar	208	60 (60/0)	2
heart	270	13 (1/12)	2	spambase	4,597	57 (57/0)	2
ionosphere	351	33 (32/1)	2	thyroid	7,200	21 (6/15)	3
iris	150	4 (4/0)	3	twonorm	7,400	20 (20/0)	2
page-blocks	5,472	10 (4/6)	5	wdbc	569	30 (30/0)	2
penbased	10,992	16 (0/16)	10	wine	178	13 (13/0)	3
pima	768	8 (8/0)	2	yeast	1,484	8 (8/0)	10
ring	7,400	20 (20/0)	2				

The accuracy estimation of each classifier is obtained by means of 5 runs of a stratified 5-fold cross-validation. The dataset is divided into 5 partition sets with equal numbers of examples and maintaining the proportion between classes in each fold. Each partition set is used as a test for the model learned from the four remaining partitions. This procedure is repeated 5 times. We use 5 partitions since if each partition has a large number of examples the noise's effects will be more notable, facilitating their analysis.

B. Process for Inducing Noise in Datasets

The initial amount of noise present in the previous datasets is unknown so we cannot make any assumptions about this base noise level. Therefore, as we want to control the level of noise in the existing data, we use a manual mechanism to add noise to each dataset.

From the 19 original datasets from the KEEL-dataset repository we have created new noisy datasets considering the introduction of class noise in the training sets. We have taken into account four levels of noise: $x = 5\%$, $x = 10\%$, $x = 15\%$ and $x = 20\%$. Introducing noise only in training sets and testing the models built over clean test sets will let us to check how classifier's generalization capability is affect by the noise's effect.

In order to introduce a level of class noise $x\%$ in a dataset, we use two different schemes:

- *Pairwise class noise scheme*. Class noise is introduced into the datasets following the pairwise scheme used

[1] http://www.keel.es/datasets.php

in [15]: given a pair of classes (X, Y), with X the majority class and Y the second majority class, and a noise level $x\%$, an instance with the label X has a probability of $x\%$ of being incorrectly labeled as Y.

- *Random class noise scheme*. We have also used a more general class noise scheme than that described above. In this scheme, a level of noise of $x\%$ supposes that exactly $x\%$ of the examples are corrupted. The class labels of these examples are aleatory changed by different ones within the domain of the class.

In order to create a noisy synthetic dataset from the original one, the noise is introduced consistently by means of the following steps:

1) A level of noise $x\%$ of a concrete type of class noise is introduced into a copy of the full original dataset.
2) Both datasets, the original one and the noisy copy, are partitioned into 5 equivalent folds, i.e. the examples within each fold of the noisy copy are the same as those within the corresponding fold of original dataset.
3) We use a 5-fold cross-validation scheme for new synthetic datasets. The datasets are created by building the training sets with the noisy copy and the test sets with the original copy.

In this manner, we have created 76 datasets with the pairwise class noise scheme and 76 with the random class noise scheme. The total number of datasets of the experimentation is therefore 171.

C. Parameters Configuration

The classification algorithms have been executed with the KEEL tool[2] [22] using the best parameters on average as shown in Table II.

Table II
PARAMETERS CONFIGURATION FOR CLASSIFICATION ALGORITHMS

FURIA	C4.5
• Number of folds: $f = 3$ • Num. of optimizations: $k = 2$ • Min. instances per premise: $i = 2$	• Confidence level: $c = 0.25$ • Min. instances per leaf: $i = 2$ • Prune after the tree building

D. Comparison methodology

In order to check which kind of algorithms, FRBCSs or robust crisp methods, are more tolerant when dealing with class noise, we compare the FURIA fuzzy method with the C4.5 crisp robust learner. We perform this comparison training the methods with noisy data, and testing the models are over clean data. In order to be able to carry out this study we use three distinct methods:

1) The mean accuracy provided by the classification algorithms over the test sets for each level of induced noise, defined as its performance averaged across all classification problems. Over the test accuracy results, we also use the Wilcoxon's signed ranks statistical test [11] with a level of significance of $\alpha = 0.05$. For each level of noise, we compare an FRBCS versus a crisp method using the Wilcoxon's test and we obtain the p-values associated with these comparisons.
2) We use the *relative loss of accuracy* (*RLA*) (Equation 5) to observe the form in which the accuracy of the model is affected when increasing the levels of noise with respect to the case with no noise:
$$RLA_{x\%} = \frac{Acc_{0\%} - Acc_{x\%}}{Acc_{0\%}} \tag{5}$$
where $RLA_{x\%}$ is the relative loss of accuracy at a level of noise $x\%$, $Acc_{0\%}$ is the mean accuracy in test in the original case, that is, with 0% of induced noise, and $Acc_{x\%}$ is the mean accuracy in test with a level of noise $x\%$.
3) We also use the *relative increase of rules* (*RIR*) (Equation 6) since another aspect that can be affected by the noise is the model's size [12], [18] and therefore, the number of rules can be related to the robustness of the model learned:
$$RIR_{x\%} = \frac{Rules_{x\%} - Rules_{0\%}}{Rules_{0\%}} \tag{6}$$
where $RIR_{x\%}$ is the relative increase of rules at a level of noise $x\%$, $Rules_{0\%}$ is the mean number of rules of the model learned from the training set with no additional noise, and $Rules_{x\%}$ is the mean number of rules of the model learned from the training set with a level of noise $x\%$.

V. CLASS NOISE'S EFFECT ON CLASSIFIERS' PERFORMANCE

In this section we focus on the analysis of the behavior of the FURIA fuzzy method versus the C4.5 algorithm when training with noisy data and the models are tested over clean test sets.

Table III shows the results of both schemes of class noise considered. The first part of the table shows the mean accuracy in test at each level of induced noise. Along with these results, the second part of the table shows the Wilcoxon's test p-values.

The mean accuracy in test of FURIA is always better than that of C4.5 for each level of induced noise in both noise schemes. This clearly shows the better performance of the FRBCS when training with data with class noise. From the associated p-values (considering a level of significance of $\alpha = 0.05$) we can say that there are significant differences in the results. This occurs with both class noise schemes for all levels of noise. However, it highlights the better behavior of FURIA with the random class noise scheme with respect to C4.5, due to the latter's test accuracy being more affected

[2]www.keel.es

Table III
RESULTS ON DATASETS WITH CLASS NOISE: TEST ACCURACY AND RELATED p-VALUES

		Mean accuracy in test					p-values for class noise				
	Noise %	**0%**	**5%**	**10%**	**15%**	**20%**	**0%**	**5%**	**10%**	**15%**	**20%**
Pairwise	FURIA	85.81	85.37	84.74	84.23	83.10	1.1444E-5	1.9074E-5	9.652E-4	1.6404E-4	2.022E-3
	C4.5	83.93	83.66	82.81	82.25	81.41					
Random	FURIA	85.81	85.17	84.54	84.06	83.66	1.1444E-5	7.63E-6	3.356E-4	3.814E-5	1.1444E-5
	C4.5	83.93	82.97	82.38	81.69	80.28					

Table IV
RESULTS ON DATASETS WITH CLASS NOISE: RELATIVE LOSS OF ACCURACY IN TEST AND RELATIVE INCREASE OF RULES IN TRAINING

		Relative loss of accuracy				Relative increase of rules			
	Noise %	**5%**	**10%**	**15%**	**20%**	**5%**	**10%**	**15%**	**20%**
Pairwise	FURIA	0.0051	0.0127	0.0191	0.0331	-0.03	-0.07	-0.08	-0.10
	C4.5	0.0035	0.0141	0.0212	0.0323	0.05	0.11	0.16	0.18
Random	FURIA	0.0073	0.0152	0.0207	0.0253	-0.01	-0.04	-0.07	-0.08
	C4.5	0.0124	0.0204	0.0285	0.0466	0.10	0.14	0.27	0.30

than that of the former one than in the case of the pairwise class noise scheme. The p-values also reflect this fact, since lower p-values are generally obtained for the random class noise scheme.

In order to obtain an approximation of the greater or lower robustness of the considered methods against class noise, Table IV shows the averages of the results of relative loss of accuracy in test of each classification algorithm and the relative increase of rules for each level of induced noise and both class noise schemes.

As is shown in Table IV, the *RLA* is lower for FURIA than for C4.5 at all considered levels of noise for both class noise schemes. However, with 5% and 20% of the pairwise class noise scheme this does not occur, although these values are very close. This again shows the greater robustness of FURIA when dealing with mislabeled data.

Regarding the *RIR*, for both class noise schemes, the results obtained by the FURIA fuzzy method must be highlighted. These results are indeed reduced with respect to the case with no noise when higher levels of class noise are introduced in the datasets. The number of rules of the FURIA algorithm is on average much better than that of the C4.5 algorithms. FURIA's rule stretching method can influence in this fact. We may conclude that the FURIA algorithm has greater robustness against class noise with respect to C4.5.

VI. REASONS OF FRBCSS' BETTER PERFORMANCE WITH DATA WITH CLASS NOISE

In this section we perform the analysis of the reasons why FRBCSs present greater robustness than crisp robust methods when dealing with data with class noise. This better behavior is due to FRBCSs having a series of properties that make them different from most of the crisp systems when dealing with class noise. Some of these properties, the most general, that we can emphasize are:

1) *The use of fuzzy sets in the antecedents of the rules, instead of crisp intervals*. This lets, for instance, to give more or less importance to the class of an example predicted by a rule, according to whether this example falls in one area or another of the membership function of the antecedents of this rule. Noisy examples can fall in areas with a lower value of the membership function of the antecedents of the rule while belonging to a different class to that predicted by the rule. Thus, these noisy examples will be influenced to a lower degree by the prediction of this rule.
2) *The assignment of a weight to each fuzzy rule*. This, along with the fuzzy sets in the antecedents, enables an overlapping between fuzzy rules. If several rules cover an example, the rule weights (and the membership function in the fuzzy sets) will let to determine the most appropriate rule that covers this example. The possible overlapping between rules is a very important fact when dealing with noisy data, because it causes the rules to be less affected by the noise that corrupts other rules.
3) *The aggregation of the fuzzy rules' predictions in order to predict the final class of an example*. This is a natural, robust way to deal with noise because the prediction is not only determined by the action of a single rule, but it is determined by the intervention of all or a part of the rules of the model. It is possible to use this thanks to the two above mentioned characteristics.

These properties cause the FRBCSs to be less affected when class noise is induced in datasets. Therefore, these systems achieve a lower overfitting of noise, leading to an increase in the accuracy of labeling test examples.

VII. Concluding Remarks

In this contribution we have analyzed the advantages of FRBCSs when dealing with data with class noise. The good performance and tolerance of the FURIA fuzzy method compared with the C4.5 crisp robust learner when class noise is present has been highlighted.

We have considered two different kinds of class noise: the pairwise class noise scheme and the random class noise scheme. Based on them, we have created 76 datasets with the former one and 76 datasets with the latter, by introducing noise in the training partitions and the models have been evaluated over clean test sets.

The results obtained have indicated that FRBCSs have better test accuracy and a better robustness in terms of the model's size when training with data with class noise than classic crisp robust learners.

Acknowledgment

Supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01. J. A. Sáez holds an FPU scholarship from the Spanish Ministry of Education and Science.

References

[1] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2004.

[2] L. Kuncheva, *Fuzzy classifier design*. Springer-Verlag, 2000.

[3] J. C. Hühn and E. Hüllermeier, "FR3: a fuzzy rule learner for inducing reliable classifiers," *IEEE Transactions on Fuzzy Systems*, vol. 17, pp. 138–149, 2009.

[4] ——, "FURIA: An Algorithm for Unordered Fuzzy Rule Induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[5] C.-H. Tsang, S. Kwong, and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection," *Pattern Recognition*, vol. 40, pp. 2373–2391, 2007.

[6] G. Schaefer, M. Zviek, and T. Nakashima, "Thermography based breast cancer analysis using statistical features and fuzzy classification," *Pattern Recognition*, vol. 42, no. 6, pp. 1133–1137, 2009.

[7] R. Y. Wang, V. C. Storey, and C. P. Firth, "A Framework for Analysis of Data Quality Research," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, pp. 623–640, 1995.

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

[9] C.-M. Teng, "Correcting Noisy Data," in *Proceedings of the Sixteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1999, pp. 239–248.

[10] 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.

[11] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[12] X. Zhu and X. Wu, "Class Noise vs. Attribute Noise: A Quantitative Study," *Artificial Intelligence Review*, vol. 22, pp. 177–210, 2004.

[13] X. Wu, *Knowledge acquisition from databases*. Norwood, NJ, USA: Ablex Publishing Corp., 1996.

[14] M. A. Hernández and S. J. Stolfo, "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem," *Data Mining and Knowledge Discovery*, vol. 2, pp. 9–37, 1998.

[15] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *Proceeding of the Twentieth International Conference on Machine Learning*, 2003, pp. 920–927.

[16] J. R. Quinlan, "The Effect of Noise on Concept Learning," in *Machine learning: an artificial intelligence approach*. Morgan Kaufmann Publishers, 1986, ch. 6, pp. 149–166.

[17] D. Gamberger, R. Boskovic, N. Lavrac, and C. Groselj, "Experiments With Noise Filtering in a Medical Domain," in *Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers, 1999, pp. 143–151.

[18] C. E. Brodley and M. A. Friedl, "Identifying Mislabeled Training Data," *Journal of Artificial Intelligence Research*, vol. 11, pp. 131–167, 1999.

[19] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, vol. 52, pp. 21–32, 1992.

[20] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, pp. 506–515, 2001.

[21] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann Publishers, 1995, pp. 115–123.

[22] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. 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, pp. 307–318, 2009.