Fuzzy Class Binarization using Coupled Map Lattices

Jonatan Gomez* and Robert Kozma
Computer Science Division, The University of Memphis, TN 38152 USA
* and Departamento de Ingeniera de Sistemas, Universidad Nacional de Colombia
{jgomez,rkozma}@memphis.edu

Abstract— **The paper presents a class binarization that combines fuzzy classifiers and coupled map lattices. First, a classification problem is divided into several two-class problems following a extended version of a fuzzy round robin class binarization scheme; next, a fuzzy classifier is generated using any machine learning technique for each two-class problem (we use evolution of fuzzy rules in this paper); finally, the generated fuzzy classifiers are integrated into a 2-dimensional coupled map lattice. The answer of the classifier to a sample is determined by the dynamics of the lattice when it is initialized with the answers given by each fuzzy classifier. Experiments are conducted with various publicly available data sets.**

Keywords. Coupled Map Lattices, Fuzzy Class Binarization, Fuzzy Classification, Evolution of Fuzzy Rules

I. Introduction

Classification is a supervised learning technique that takes labeled data samples and generates a model (classifier) that classifies new data samples in different predefined groups or classes [1]. Classification has been extensively studied in machine learning and data mining [1], [2], [3], and has received special attention of soft-computing techniques such as fuzzy Logic [4], [5], [6], neural networks [7], [8], and evolutionary algorithms [9], [10], [11]. Due to high interpretability of fuzzy rule based classifiers (**FRBC**) and the ability of evolutionary algorithms (**EA**) in finding good solutions, some research work has focused in developing evolutionary techniques for generating FRBC [12], [13], [9], [14], [6], [11].

Because several machine-learning techniques have been designed for solving two-class classification problems, but not for solving multi-class problems, some techniques (class binarization techniques) have been developed for transforming a multi-class problem into several two-class problems [15]. For each of these two-class problems, a classifier is trained using a specific learning mechanism, for example a neural network or evolution of fuzzy rules. Finally, those classifiers are combined following the applied binarization scheme. Although used with relative success, these techniques depend on the correct operation of each of their two-class classifiers. If one of those two-classifiers fails, the performance of the full classifier is highly affected.

The main purpose of the work presented in this paper is to develop a fuzzy class binarization technique using a coupled map lattices [16]. By using such coupled map lattice structure we are increasing the robustness of the classifier while increasing the performance achieved by the classifier. Here, the answer of the classifier to a sample is determined by the dynamics of the lattice when it is initialized with the fuzzy classification given by each two-class fuzzy classifier. This paper is divided in six sections. Section 2 gives a simple overview of coupled map lattice theory; Section 3 describes the current class binarization, both crisp and fuzzy; Section 4 presents the proposed fuzzy class binarization technique based on coupled map lattices; Section 5 shows some experiments performed along with the analysis of results. Section 6 draws some conclusions.

II. Coupled Map Lattices

Coupled map lattice (**CML**) theory grew out of studies on collective movements of coupled oscillators [16], [17]. A CML can be defined in any d-dimensional space ($d = 1, 2, ..$) but in this paper, we consider only 2-dimensional CML of size $N \times M$.

Definition 1: A 2-**D** Coupled Map Lattice is a 2-dimensional lattice of size $N \times M$ such that each site $x(i,j)$ evolves according to equation 1.

$$x(i,j)_{t+1} = (1-\varepsilon)\, f\left(x_t\left(i,j\right)\right) + \frac{\varepsilon}{s\left(N_{x(i,j)}\right)} \sum_{y \in N_{x(i,j)}} f\left(y_t\right) \tag{1}$$

where, $N_{x(i,j)}$ is a collection of sites which defines the neighborhood of site $x(i,j)$, $s\left(N_{x(i,j)}\right)$ indicates the number of neighbors, f is a non-linear function that determines the coupling value of a site, and ε is a parameter that determines the coupling level or spatial correlation between sites. By varying the coupling parameter ε and the structure of the neighborhood $N_{x(i,j)}$ it is possible to simulate systems that interact locally up to systems that interact globally [17].

Several non-linear functions have been used into a CML. In the following discussion we use the well-known logistic function:

$$f(x) = 1 - \alpha x^2 \tag{2}$$

where, $0 \leq \alpha \leq 2$ is a suitable parameter that allows to modify the dynamics of the system and introduce chaotic behavior into the CML. In order to introduce chaos to the CML, αshould be set to a value higher than 1.82.

III. Fuzzy Class Binarization

A class binarization technique transforms a multi-class classification problem into several two-class classification problems [15]. Basically, there are two fuzzy class binarization strategies: Unordered, and round robin.

A. *Fuzzy Unordered Class Binarization*

Fuzzy unordered class binarization transforms an *m*-classes problem into *m* two-class problems, where the *i*-th classifier that is generated using the samples of class *i* as negative samples and samples of the other classes *(j=1..m, j≠i)* as positive samples. A new sample is classified according to algorithm 1. We have successfully applied this binarization technique with a maximum defuzzyfication technique in [12], [18], [19].

Algorithm 1 Fuzzy Unordered Classification

```
CLASSIFY( $classifier[1..m]$, $sample$ )
$winners = \emptyset$
for $i = 1$ to $m$ do
  $winners = winners \cup \{\ \mu_{negative}\ (classifier_i\ ,\ sample)\ \}$
return DEFUZZY( $winners$ )
```

B. *Fuzzy Round Robin Class Binarization*

Fuzzy round robin class binarization transforms an *m*-class problem into $\frac{m(m-1)}{2}$ two-class problems by defining one classifier $classifier_{i,j}$ for each pair of classes $\langle i, j \rangle$. Each classifier is generated using only the samples of the two corresponding classes. A new sample is classified according to algorithm 2.

Algorithm 2 Fuzzy round robin classification

```
CLASSIFY( $classifier[1..m-1][i+1..m]$, $sample$ )
for $i = 1$ to $m$ do
  $vote[i] = 0$
for $i = 1$ to $m - 1$ do
  for $j = i + 1$ to $m$ do
    $vote[i] = vote[i] + \mu_i\ (classifier[i][j], sample)$
return DEFUZZY( NORMALIZE($vote$) )
```

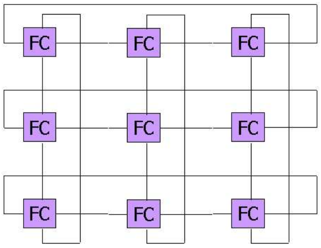


Fig. 1. Neighborhood definition for a three-class classification problem.

IV. Classification using Cell Organized Recognition ALgorithms (CORAL)

CORAL combines both fuzzy unordered and fuzzy round robin class binarization schemes in one binarization scheme. In this way, CORAL transforms a *M*-class problem into M^2 two-class problems by defining two classifiers $classifier_{i,j}$ for each pair of different classes $\langle i, j \rangle$ and one classifier $classifier_i$ for each different class. Every classifier $classifier_{i,j}$ is trained with only samples of class i and class j by considering class i as the negative class (class j is consider the positive class). Therefore, CORAL trains a classifier $classifier_{i,j}$ following the round robin training strategy. Now, every classifier $classifier_i$ is trained with samples of class i as negative and samples of other classes as positive. Then, a classifier $classifier_i$ is trained following the unordered training strategy.

Notice that, it is possible to organize the two-class classifiers in a two dimensional lattice x, such that, $x(i, j) = classifier_{i,j}\ i \neq j$ and $x(i, i) = classifier_i$. Such lattice has the following properties:

- A classifier in row i considers class i as negative.
- A classifier in column j (exception done with the diagonal element) considers class j as positive.
- A classifier in the diagonal position i considers class i as negative.

Therefore, elements in the same row or in the same column are correlated. Using these properties, it is possible to define the neighborhood of a site $x(i, j)$ (two-class classifier) as follows (equation 3):

$$\begin{array}{rl} N_{x(i,j)} = & \{x(i,k) | k = 1, 2, .., M\, and\, k \neq j\} \cup \\ & \{x(k,j) | k = 1, 2, .., M\, and\, k \neq i\} \end{array} \tag{3}$$

Sites in the same row or column of a given site belong to the neighborhood of such site. The size of the neighborhood is $2\,(M - 1)$ where M is the number of classes. Figure 1 shows the neighborhood definition for a three-class classification problem.

In order to use this lattice for classifying a data sample $sample$, some state concept and dynamics should be associated to it. The state of a site $x(i, j)$ on any time

$t \geq 0$ indicates the confidence degree that such site has on classifying the data sample into class i. The initial state of site $x(i,j)$ is defined as the membership degree of the sample to class i generated by the classifier associated to such site, see Equation 4:

$$x_0(i,j) = \begin{cases} \mu_{i,classifier_i}(sample) & i = j \\ \mu_{i,classifier_{i,j}}(sample) & i \neq j \end{cases} \tag{4}$$

Now, the state of a site $x(i,j)$ evolves on time according to equation 5:

$$x_{t+1}(i,j) = (1-\varepsilon)\, x_t\,(i,j) + \tfrac{\varepsilon}{2(M-1)} * \left[\sum_{\substack{k=1,\\ k \neq j}}^{M} f\left(x_t(i,k)\right) + \sum_{\substack{k=1,\\ k \neq i}}^{M} f\left(1.0 - x_t(k,j)\right) \right] \tag{5}$$

where $f(x)$ is the logistic function, see equation 2. We divided the neighborhood in two groups, row-neighborhood and column-neighborhood because the state of them is considered different for the site in evolution: On one hand, the state of a site of the row-neighborhood is used directly because such site considers class i as negative, like site $x(i,j)$; On the other hand, the state of a site in the column neighborhood is negated (fuzzy negation) because such site considers class j as negative, opposite to the site $x(i,j)$. Finally, the classification of a data sample is determined by a defuzzy operator that is applied to the state of the sites in the diagonal after some number of iterations MAX of the CML. Algorithm 3 shows the CORAL classification.

Algorithm 3 CORAL Classification Algorithm

```
CLASSIFY( $classifier[1..M][1..M]$, $MAX$, $sample$ )
for $i = 1$ to $M$ do
  for $j = 1$ to $M$ do
    $x[i][j]$ =FUZZY( $classifier[i][j]$, $i$, $sample$)
for $i = 1$ to $MAX$ do
  $x$ =ITERATE_CML( $x$ )
for $i = 1$ to $M$ do
  $vote[i] = x[i][i]$
return DEFUZZY( NORMALIZE($vote$) )
```

V. Experimentation

In order to test the proposed approach we used the evolution of fuzzy rules proposed by Gomez et al in [12] for evolving a fuzzy rule for each two-class classification problem generated with CORAL. Basically, the evolutionary algorithm evolves a heap tree representing the condition part of the fuzzy rule. A heap tree is a binary tree that is filled completely on all the levels except possibly the last level that is filled from left to right [20]. The conclusion part is not encoded since only two-classes define the classification problem and one rule is being generated for discriminating the positive class [12]. A data sample is classified as positive with the truth-value (**TV**) of the fuzzy rule R and classified as negative with TV equal to the fuzzy negation of the TV of R $(1 - TV(R))$.

TABLE I

Test bed

DATA SET	CLASSES	DIM.	SAMPLES
IRIS	3	4	150={50, 50, 50}
WINE	3	13	178={59, 71, 48}
GLASS	6	9	214={70, 76, 17, 13, 9, 29}

TABLE II

Fuzzy Logic Operators

OPERATOR		TRUTH-VALUE
Not	$\neg p$	$\neg p = 1 - p$
Average And	$p \wedge q$	$p \wedge q = \frac{2*p*q}{p+q}$
Restricted Sum Or	$p \vee q$	$p \vee q = min\left\{1, p+q\right\}$

A. Experimental Settings

Three benchmark data sets (publicly available), were used as test bed. See table I. A ten folding cross-validation was applied to each data set. In this way, the data set was divided into 10 randomly selected groups. Each group was used as testing data set of the classifier trained with the additional 9 groups. The reported results are the average over these 10 different tests.

For evolving the fuzzy rule associated with each two-class problem we used the Simple Hybrid Adaptive Evolutionary Algorithm (**SHAEA**) proposed by Gomez et al. [21]. The SHAEA approach adapts the genetic operator probabilities at the same time it is finding the solution of the problem. SHAEA was executed for 100 iterations using 100 individuals as population and variable length crossover, single bit mutation, gene addition and gene deletion as genetic operators. These operators are described by Gomez et al in [12]. We used the fuzzy logic operators shown in table II, and five fuzzy sets per attribute as shown in figure 2.

We used a coupling parameter of $\varepsilon = 0.5$ in order to give more importance to the site than to its neighbors and iterated the CORAL classifier 100 periods of time using the logistic function with parameter $\alpha = 1.5$.

B. Results and Analysis

Several performance metrics have been defined for determining the quality of a classifier. Most of them are summarized in a confusion matrix. A confusion matrix contains the

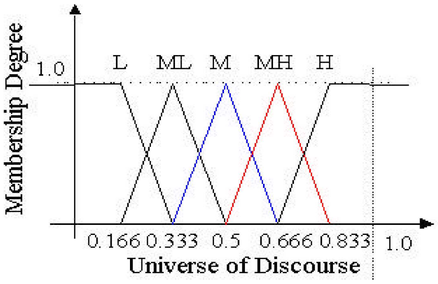


Fig. 2. Fixed Collection of Fuzzy Sets

TABLE III
CONFUSION MATRIX

	PREDICTED	
ACTUAL	**Positive**	**Negative**
Positive	*PP*	*PN*
Negative	*NP*	*NN*

classification performance (actual versus predicted class) reached by a classifier system over a set of samples [1]. The confusion matrix for a two-class (Negative and Positive) classifier is shown in table III.

From the confusion matrix, several performance metrics can be generated. One of the most important metric is the accuracy, which determines the proportion of samples classified correctly, see equation 6:

$$AC = \frac{PP + NN}{PP + PN + NP + NN} \tag{6}$$

1) CORAL Dynamics: We took one of the 10 classifiers generated for the WINE data set and plotted the state through the time of the classifier sites when a data sample was presented to such classifier. We selected three data samples, one per each class, with the condition that they were not used in the training process. Figure 3 shows the evolution of the state for sites $x(1,1)$, $x(2,1)$, and $x(3,1)$ of the CORAL classifier. We observed similar behavior for sites $x(1,2)$, $x(2,2)$, and $x(3,2)$; and $x(1,3)$, $x(2,3)$, and $x(3,3)$, respectively.

Since the logistic function is not introducing chaos to the CML, the state of a site has a non-chaotic behavior with a two-state limit orbit. This behavior indicates that CORAL is able to reach a stable condition. As expected, one of these states is approximated when CORAL executes an even number of iterations while the other state is approximated when CORAL executes an odd number of iterations. Similar behavior was observed for the two data samples belonging to class 2 and class 3. Figure 4 shows the separated evolution of the limit orbit states for each diagonal site and each of the three data samples used.

Figure 4 shows that CORAL classifier should be allowed to execute an even number of iterations in order to obtain an optimal classification rate. As shown, CORAL does not requires many iterations in order to reach a good classification performance: around 20 iterations the performance is close to optimum.

TABLE IV
PERFORMANCE OF CORAL, FUZZY UNORDERED (FUB) AND FUZZY ROUND ROBIN (FRRB).

	IRIS	**WINE**	**GLASS**
CORAL	**95.33**	**96.04**	**68.09**
FUB	96.00	94.31	61.17
FRRB	94.66	95.49	60.02

2) Comparing CORAL against Fuzzy Round Robin and Fuzzy Unordered Binarization Schemes: Table IV shows the performance (accuracy) reached by using **CORAL** after 20 iterations, Fuzzy Unordered binarization (**FUB**) and Fuzzy Round Robin Binarization (**FRRB**). In general, CORAL performs better than FUB and FRRB for these data sets.

3) Robustness: In order to determine the robustness of the proposed approach, different level of damage were introduced into the classifier generated by CORAL for the WINE data set. We simulate the failure of a site by initializing it with the value 0.5 instead of using the fuzzy classifier associated to the site. In this way, a site $x(i,j)$ is considered failing when its associated classifier is not able to determine the class of any data sample.

We simulated the CORAL classifier when a single site was failing: (1,1), (2,2), (3,3), and (2,1). Figure 5.a shows the evolution of the average performance reached by the failing CORAL classifiers.

As expected the performance of the classifier is not strongly affected when one site is failing. In some cases, the performance of the classifier is the same or better than the original classifier. Moreover, the worst performance reached by these classifiers after 20 iterations was 94.72 (failing site : (2,1)).

Also, we determine the performance of the CORAL classifier when different groups of three sites are failing: diagonal sites, same row sites, same column sites and sites (1,2), (2,3) and (3,1). Figures 5.b and 5.c show the evolution of the average performance reached by these failing CORAL classifiers.

Again, CORAL is able to perform well when the amount of damage introduced is 33.3%. The worst performance reached by CORAL after 20 iterations was 85.88% of accuracy when the first row of sites is failing. Otherwise, the accuracy reached was higher than 90.00%.

4) Comparison with Results Reported in the Literature: We took the results produced by our approach and compared them against some results reported in the literature,

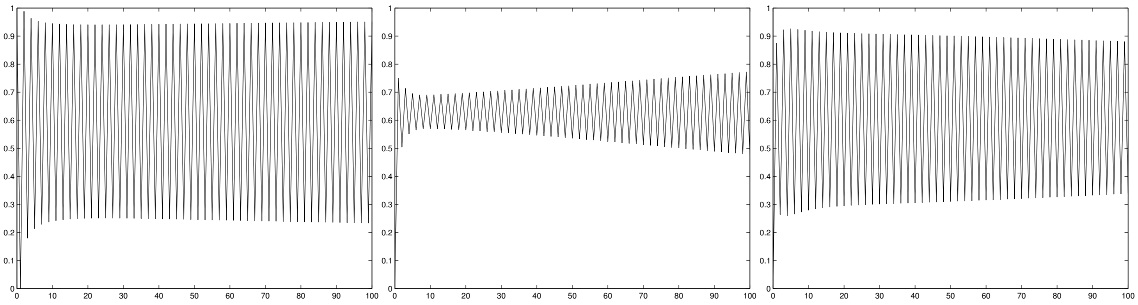


Fig. 3. CORAL Dynamics for a data sample belonging to class 1 in the WINE data set.

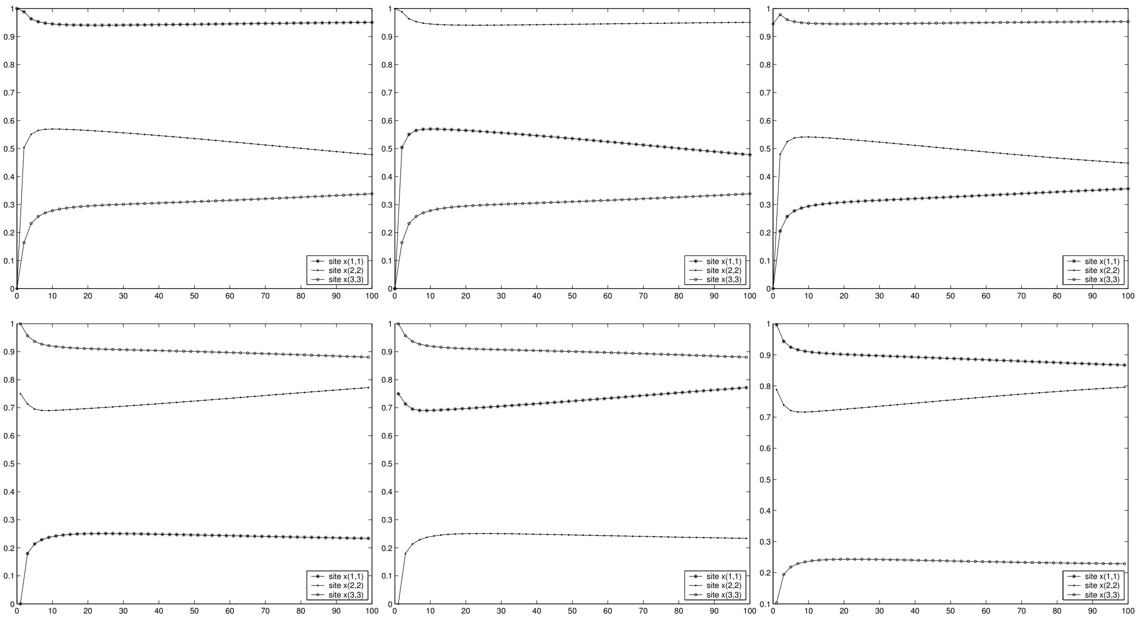
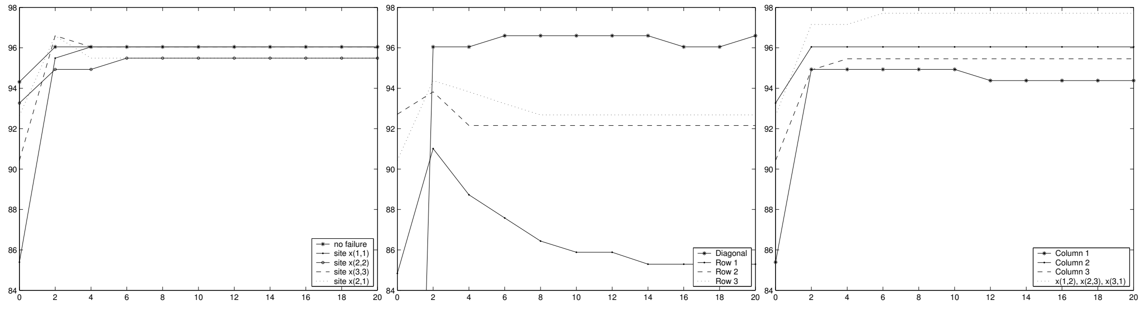


Fig. 4. Separated dynamics of two-states that define the limit orbit for the diagonal sites. First row shows the even state (state reached by CORAL in an even number of iterations), while second row shows the odd state. Column 1 (2and 3) when CORAL is classifying a data sample of class 1 (2 and 3 respectively).



(a) (b) (c)

Fig. 5. Evolution of the accuracy of CORAL when 3 sites are failing. (a) none site failing, (b) Failing diagonal and rows, (c) Failing columns and sites $\{(1,2), (2,3), (3,1)\}$

TABLE V

COMPARATIVE PERFORMANCE OF THE PROPOSED APPROACH

Method	WINE	IRIS	GLASS	Statistical Test
CORAL	**96.04**	**95.33**	**68.09**	**10-cross-validation**
CTree	94.11	93.73	-	10-cross-validation
QDA	99.40	-	-	Leave-one-out
LDA	98.90	-	59.70	Leave-one-out
GAP-sel	97.20	94.70	70.10	None
GAP-par	93.30	96.00	67.30	None
kNN	95.50	-	72.00	Leave-one-out
SSV	98.30	-	-	10-cross-validation
FSM	96.10	-	-	10-cross-validation

see table V.Results reported for QDA, LDA, C4.5, kNN, SSV and FSM taken from [22]. Results for GAP taken from [23], where the number of fuzzy rules was close to the number of classes. Results for CTree are taken from [12]. Table V shows that our results (first row) compare well with alternative approaches. Although all of these results were obtained with different statistical validation methods (leave-one-out, or 10-cross-validation) or not statistical validation, the values reported here are an indicative of the performance of the proposed approach.

VI. CONCLUSIONS AND FUTURE WORK

We proposed a technique for class binarization based on Coupled Map Lattices and fuzzy logic that uses evolution of fuzzy rules as proof of concept. The results indicate that the performance of the proposed approach (CORAL) is better than other binarization schemes. Also, the performance of the classifier is into strongly affected when different levels of failures are introduced to the lattice.

Our future work will determine the effect of introducing chaos into the CORAL classifier by varying the logistic function parameter α between 1 and 2. Also, we plan to train the CORAL classifier in an on-line and partially unsupervised manner using the CORAL system dynamics. Our idea is to updated each site of a CORAL classifier with unlabeled data samples that are classified on such sites.

Acknowledgments This work was partially supported by NSF grant EIA-0130352.

REFERENCES

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[2] R. S. Michalski, I. Bratko, and M. Kubat, *Machine Learning and Data Mining: Methods and Applications*. J. Wiley —& Sons, 1998.

[3] R. Holte, "Very simple classification rules perform well in most common used datasets," *Machine Learning*, no. 11, pp. 63–91, 1993.

[4] Y.-C. Hu, R.-S. Chen, and T. G-H., "Finding fuzzy classification rules using data mining techniques," *Pattern Recognition Letters*, no. 24, pp. 509–519, 2003.

[5] Q. Shen and A. Chouchoulas, "A rough-fuzzy approach for generating classification rules," *Pattern Recognition*, no. 35, pp. 2425–2438, 2002.

[6] A. Gonzalez and R. Prez, "Completeness and consistency conditions for learning fuzzy rules," *Fuzzy Sets and Systems*, no. 96, pp. 37–51, 1998.

[7] A. Nurnberger, W. Pedrycz, and R. Kruse, *Classification: Neural Net Approaches, In: Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, 2002.

[8] R. Haskell, "Neuro-fuzzy classification and regression trees," *Machine Learning*, no. 11, pp. 63–91, 1993.

[9] H. Ishibuchi and T. Nakashima, "Liguistic rule extraction by genetics-based machine learning," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO'00*, pp. 195–202, 2000.

[10] A. Giordana and L. Saitta, "Regal: An integrated system for learning relations using genetic algorithms," in *Proceedings of the Second International Workshop on Multi-strategy Learning*, pp. 234–249, 1993.

[11] K. De Jong and W. Spears, "Learning concept classification rules using genetic algorithms," in *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 651–656, 1991.

[12] J. Gomez, D. Dasgupta, O. Nasraoui, and F. Gonzalez, "Complete expression trees for evolving fuzzy classifier systems with genetic algorithms," in *Proceedings of the North American Fuzzy Information Processing Society Conference NAFIPS-FLINTS 2002*, pp. 469–474, 2002.

[13] M. V. Fidelis, H. S. Lopes, and A. A. Freitas, "Discovering comprehensible classification rules with a genetic algorithm," in *Proceedings of Congress on Evolutionary Computation (CEC)*, pp. 805–810, 2000.

[14] J. Liu and J. Kwok, "An extended genetic rule genetic algorithm," in *Proceedings of Congress on Evolutionary Computation (CEC)*, pp. 458–263, 2000.

[15] J. Frnkranz, "Round robin classification," *Machine Learning Research*, no. 2, pp. 721–747, 2002.

[16] K. Kaneko, *Theory and Applications of Couple Map Lattices*. John Wiley, 1993.

[17] R. Kozma, "Intermediate-range coupling generates low-dimensional attractors in the chaotic region of one-dimensional lattices," *Physics Letters A*, no. 244, pp. 85–91, 1998.

[18] J. Gomez and D. Dasgupta, "Evolving fuzzy rules for intrusion detection," in *Proceedings of the Third Annual IEEE Information Assurance Workshop 2002 Conference*, pp. 68–75, 2002.

[19] D. Dasgupta and F. Gonzalez, "Evolving complex fuzzy classifier rules using a linear tree genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO'01*, pp. 299–305, 2001.

[20] T. Cormer, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. McGraw Hill, 1990.

[21] J. Gomez, D. Dasgupta, and F. Gonzalez, "Using adaptive operators in genetic search," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, June 2003.

[22] W. Duch, "Data sets used for classification: Comparison of results," in *http://www.phys.uni.torun.pl/kmk/projects/datasets.html*.

[23] T. Murata, S. Kawakami, H. Nozawa, M. Gen, and H. Ishibuchi, "Three-objective genetic algorithms for designing compact fuzzy rule-based systems for pattern classification problems," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO'01*, pp. 485–492, 2001.