

An Architecture for Constructing Fuzzy Regression Tree Forests Using Opt-aiNet

Fathi Gasir
Intelligent Systems Group
School of CMDT
MMU
Manchester, UK
f.gasir@mmu.ac.uk

Zuhair Bandar
Intelligent Systems Group
School of CMDT
MMU
Manchester, UK
z.bandar@mmu.ac.uk

Keeley Crockett
Intelligent Systems Group
School of CMDT
MMU
Manchester, UK
k.crockett@mmu.ac.uk

Abstract— This paper presents a new approach to combining multiple fuzzy regression trees, which are induced by applying the modified Elgasir fuzzy regression tree algorithm. This method utilises Trapezoidal membership functions for fuzzification and the Takagi-Sugeno fuzzy inference to obtain the final predicted values. A modified version of Artificial Immune Network model (opt-aiNet) is used for the simultaneous optimization of the membership functions across all trees within the forest. Boston housing and Abalone are two real-world datasets from the UCI repository used to evaluate the proposed approach. The empirical results have showed that fuzzy regression tree forests reduce the error rate compared with single fuzzy regression tree.

Keywords: fuzzy regression tree forests, Fuzzy inference system, Fuzzy Regression tree, Data mining, Machine learning, Evolutionary algorithms, Artificial Immune system.

I. INTRODUCTION

Decision trees are a transparent method of representing a set of IF-THEN rules in a graphical way and are generated by learning and reasoning from feature-based examples. There are two well known and distinctive methods for generating decision trees: Regression trees [1,2,3,4,5,6] which are designed to predict continuous outcomes and Classification trees [7,8,9,10,11] which predict discrete outcome classes. AID [12], CHAID [13] and CART (classification and regression tree) [14,15] are crisp regression trees algorithms that describe the approach of building regression trees based on information theory. A well documented weakness of these algorithms is that they induce trees with sharp decision boundaries when the decision space of continuous attributes is partitioned. Moreover the output of crisp regression trees are generated using a discrete function which clearly decreases the prediction accuracy and affect the robustness of the generation crisp regression trees [1,2,3,4].

The application of fuzzy logic to the crisp regression trees has shown improvement in the performance of the traditional regression trees [1...6,16]. This has been achieved using fuzzy membership functions at each crisp regression tree node to soften the sharp node boundaries. The Fuzzy CHAID Induction Algorithm was first proposed by Fowdar et al [5,6] to generate regression trees by using the CHAID

algorithm and optimising the amount of Fuzzification using a Genetic Algorithm (GA). The main weakness of Fowdars' approach was in the defuzzification strategy as it could lead to less accurate prediction as a result of using one of varied defuzzification methods to calculate the final output.

The Elgasir algorithm [2] is a method for creating fuzzy regression trees [1] by formulating trapezoidal membership functions for each node in the pre-generated crisp regression trees. This algorithm has shown the capability of overcoming the weakness of crisp decision trees by applying appropriate membership functions for all branch split points, and using Takagi-Sugeno fuzzy inference technique [17] to generate a final continuous output value. Elgasir has successfully alleviated Fowdar's defuzzification problem as a result of having a single numeric value as the output of fuzzy inference [17]. Fuzzification has been successfully used with multiple decision trees models [16,23...26] to induce fuzzy decision tree forests in order to improve the predictive power of a single fuzzy decision tree. Although the use of multiple regression trees has been used to improve the performance of a single fuzzy regression tree [16], little work has been done in this field.

Immune algorithms have been used to solve optimization problems [18] such as: design optimization task for electromagnetic devices [19], optimizing the dynamic function [20] and optimization of a weapon-target assignment problem [21]. An adapted version of Artificial Immune Network model (opt-aiNet) [22] was proposed in [2] to enable the algorithm to be applied to the optimization of fuzzy regression trees and the algorithm was successfully applied to the Elgasir [2]. The adapted version of an artificial immune network model (opt-aiNet) [2] is used in this work for optimizing the proposed framework.

This paper proposes a new approach which applies a modified version of the Elgasir algorithm to the task of creating fuzzy regression tree forests. The modification of the Elgasir algorithm has been done based on an approach by Crockett et al for growing a fuzzy classification based decision tree forest. In the modified Elgasir algorithm, a number of fuzzy regression trees are generated from one training sample, where each tree represents a different view

of the problem domain and will represent different associations of independent variables for the same dataset.

The organisation of this paper as follows: Section II provides some background information on Fuzzy Forests. Section III provides a brief outline of the Elgasir algorithm. Section IV describes the methodology used for constructing fuzzy regression tree forests. Section V describes the conducted experiments and section VI presents the results and discussion. Finally the paper is concluded in section VII.

II. FUZZY FORESTS

Fuzzy multiple decision trees forests models [16, 23...26] have been used to combine more than one fuzzy decision tree in order to improve the predictive power of a single fuzzy decision tree. Different approaches have been proposed for classification problems [23...26]. These approaches can be described as tree-based ensemble classifiers [23], but they differ by the way that used to induce multiple trees. Bonissone et al [23] have applied fuzzy logic and fuzzy sets to random forest to produce multiple fuzzy classifier system. In Bonissone's method a fuzzy random forest is created by combining several individual classifiers based on Breiman's methodology [27]. This method used a random subset of the total set of available attributes to construct each node. Marsala and Bouchon-Meunier presented a method for generating forests of fuzzy decision trees [24], where each fuzzy decision tree of the forest is induced from a random subset of the whole training dataset. With these two methods there is no guarantee that the information space has been explored exhaustively. Cezary and Janikow [25] induced fuzzy decision forests by allowing the usage of more than one test attribute at a non - leaf node. Although this approach can overcome noisy or missing values problems, but it has been unstable [28]. Crockett et al [26] introduced an approach for growing a fuzzy decision forest. In this approach, multiple classification trees are created from same training sample and the strategy for adding classification tree is by restriction the attribute which has highest-ranking to be the root of the tree. This method was shown to be successful but if the number of trees in the forest has a limit in terms of improving performance and once a certain threshold is reached, overfitting of the training set will start to occur. However it was shown that if the number of trees was kept small, then the results justify the computational expense and provide a practical way of improving the classification accuracy. For regression problems, Haskell [16] proposed a method to induce a fuzzy regression trees forest, which used a new splitting criteria based on a generalized Kolmogorov-Smirnov distance to construct fuzzy regression trees associated with a fuzzy cumulative distribution function. The main problem with this method is that a human expert is needed to provide pre-fuzzified data, despite the number and size of the membership functions being automatically determined in Haskell's method.

III. ELGASIR ALGORITHM

The Elgasir algorithm, originally proposed in [1], combined principles of fuzzy theory and Takagi-Sugeno fuzzy inference [17] to generate fuzzy regression trees. This algorithm does not require pre-fuzzified data and therefore can overcome the problems of unnecessary complexity and poor transparency problems often resulting from algorithms which generate fuzzy trees directly from data which has been pre-fuzzified by experts in the domain. Another advantage of the Elgasir algorithm is that the output of fuzzy inference is a single numeric value as a result of using Takagi-Sugeno fuzzy inference technique. In other words, there is no need for a defuzzification operation to calculate the output of the fuzzy regression tree. This has shown to lead to more accurate results by avoiding doing more calculations to transform the output fuzzy sets into a non-fuzzy output [1,2].

The fuzzy rule base generated by Elgasir is produced by applying Trapezoidal membership function to crisp CHAID regression trees. The algorithm is not restricted to just Trapezoidal membership functions and any other membership function can be applied. In addition, the Elgasir methodology can be applied to any other crisp regression tree algorithm. For the purpose of this work the CHAID regression tree algorithm has been used [13]. Full details of the original Elgasir algorithm can be found in [1].

IV. MODIFIED ELGASIR ALGORITHM FOR CONSTRUCTING FUZZY REGRESSION TREE FORESTS

The modified Elgasir algorithm is the new version of Elgasir algorithm [1], which is used to induce a fuzzy regression tree forest by combining two or more singular fuzzy regression trees, where each one represents a different view of the same domain. The algorithm is optimized using the adapted version of an artificial immune network model (opt-aiNet) [2] in order to establish the optimum fuzzification throughout the tree.

The modified Elgasir algorithm consists of the following stages:

The first stage: multiple crisp regression trees induction

1.1 Creating a series of crisp regression trees:

The CHAID algorithm [13] is used to construct an optimal singular crisp regression tree by applying various values of CHAID algorithm parameters in order to produce a highly optimised regression tree. The attribute selection, branching strategy and terminate growth criteria for growing a crisp tree are based on the information measure entropy. To induce the second crisp regression tree from the same dataset, the attribute which has the smallest adjusted p-value (the highest ranking) is restricted from formulating the root of the second crisp regression tree. This process is repeated in each creating of crisp regression tree. The result is a combination of crisp regression trees, where each tree is

different in representing the association of independent variables for the same dataset.

1.2 Generating IF-THEN rules:

The combination of crisp regression trees are converted into a set of IF-THEN production rules in-order to create a knowledge base using a one-to-one mapping. Each rule represents a path from the root to the leaf node in each crisp CHAID regression tree. The form of the typical rule as following:

$$R_i : \text{IF } x_1 \text{ is } b_{i1} \text{ and } \dots \text{ and } x_k \text{ is } b_{ik} \text{ THEN } y_i = b_{i0} \quad (1)$$

where $R_i (i=1,2,\dots,n)$ denotes the i th rule, $x_j (j=1,2,\dots,k)$ are the input (antecedent) variables, y_i are the rule output variables, $b_{i0} (i=1,2,\dots,n)$ are real numbers.

Due to method used to create each tree there will be no duplication of rules and hence no redundancy.

The Second Stage: applying membership functions and fuzzy inference technique

Appropriate membership functions are applied for all branch split points i within the antecedent rules and the consequent part of rules (1) are parameterized. Then a typical fuzzy rule can be defined as:

$$R_i : \text{IF } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_k \text{ is } A_{ik} \text{ THEN } y_i = \rho_i b_{i0} + a_{i1} \cdot x_1 + \dots + a_{ik} \cdot x_k$$

Let $\rho_i b_{i0} = a_{i0}$ then

$$R_i : \text{IF } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_k \text{ is } A_{ik} \text{ THEN } y_i = a_{i0} + a_{i1} \cdot x_1 + \dots + a_{ik} \cdot x_k \quad (2)$$

where $R_i (i=1,2,\dots,n)$ denotes the i th fuzzy rule, $x_j (j=1,2,\dots,k)$ are the input (antecedent) variables, y_i are the rule output variables, $A_{i1}, \dots, A_{ik}$ are fuzzy sets defined in the antecedent space, and $a_{i0}, a_{i1}, \dots, a_{ik}$ are the consequent parameters.

The inferred global output of the proposed framework is computed by taking the weighted average of the individual rules' contributions defined as:

$$y = \frac{\sum_{i=1}^n \tau_i(x) \cdot (a_{i0} + a_{i1} \cdot x_1 + \dots + a_{ik} \cdot x_k)}{\sum_{i=1}^n \tau_i(x)} \quad (3)$$

where $\tau_i(x)$ is the degree of fulfilment of the i th fuzzy rule, defined by

$$\tau_i(x) = \min\{\mu_{A_{i1}}(x_1), \dots, \mu_{A_{ik}}(x_k)\}$$

In order to Identify the consequent parameters

Let β_i be

$$\beta_i = \frac{\tau_i(x)}{\sum_{i=1}^n \tau_i(x)} \quad (4)$$

then

$$y = \sum_{i=1}^n \beta_i \cdot (a_{i0} + a_{i1} \cdot x_1 + \dots + a_{ik} \cdot x_k)$$

$$y = \sum_{i=1}^n (\beta_i \cdot a_{i0} + \beta_i \cdot a_{i1} \cdot x_1 + \dots + \beta_i \cdot a_{ik} \cdot x_k)$$

When we apply a set dataset $x_{j1}, x_{j2}, \dots, x_{jk}$ then the value of the outputs y_j

where $(j=1,2,\dots,m)$ as

$$\begin{aligned} y_1 &= (\beta_{11} \cdot a_{10} + \beta_{11} \cdot a_{11} \cdot x_{11} + \dots + \beta_{11} \cdot a_{1k} \cdot x_{1k}) + \dots + (\beta_{1n} \cdot a_{n0} + \beta_{1n} \cdot a_{n1} \cdot x_{11} + \dots + \beta_{1n} \cdot a_{nk} \cdot x_{1k}) \\ &\vdots \\ y_m &= (\beta_{m1} \cdot a_{10} + \beta_{m1} \cdot a_{11} \cdot x_{m1} + \dots + \beta_{m1} \cdot a_{1k} \cdot x_{mk}) + \dots + (\beta_{mn} \cdot a_{n0} + \beta_{mn} \cdot a_{n1} \cdot x_{m1} + \dots + \beta_{mn} \cdot a_{nk} \cdot x_{mk}) \end{aligned}$$

The consequent parameters $a_{i0}, a_{i1}, \dots, a_{ik}$

$(i=1,2,\dots,n)$ can be obtained using the least square method by combining previous equations

Using matrixes and vectors;

Least square equation can be expressed as:

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, X = \begin{bmatrix} \beta_{11} \cdot x_{11}, \dots, \beta_{11} \cdot x_{1k}, \dots, \beta_{1n} \cdot x_{11}, \dots, \beta_{1n} \cdot x_{1k} \\ \vdots \\ \beta_{m1} \cdot x_{m1}, \dots, \beta_{m1} \cdot x_{mk}, \dots, \beta_{mn} \cdot x_{m1}, \dots, \beta_{mn} \cdot x_{mk} \end{bmatrix}, A = \begin{bmatrix} a_{10} \\ a_{11} \\ \vdots \\ a_{1k} \\ \vdots \\ a_{n0} \\ a_{n1} \\ \vdots \\ a_{nk} \end{bmatrix}$$

Where

$$Y = XA$$

$X(m \times n(k+1))$ matrix), Y (m vector) and A ($n(k+1)$ vector).

Then the parameter vector A is given by

$$A = (X^T X)^{-1} X^T Y$$

The Third Stage: Optimization

The modified Elgasir algorithm is optimized using the adapted version of an artificial immune network model (opt-aiNet) [2] which is described in Figure 2

```

Initialization: randomly initialize a population
while (stopping criterion is not met)
{
    for each network cell
    {
        Determine the fitness
    }
    -Normalize the vector of fitness's in the interval [0,1],
    where 0 is assigned to the highest fitness, while 1 is
    assigned to the lowest fitness.
    for each network cell
    {
        Generate a number of clones  $N_c$ 
        for each clone
        {
            -Mutate each clone proportionally to the fitness
            of its parent cell.
            -Determine the fitness
            -Select the cell with lowest fitness
            -if fitness of best clone less than its parent
            {
                The clone replaces the parent in the network
            }
        }
    }
    if the average error of the population is not
    significantly different from the previous iteration
    {
        for each network cell
        {
            -calculate the affinity between this
            network cell and the other network cells
            -if two cells have an affinity below a
            pre-defined threshold
            {
                the cell with the highest fitness is
                deleted from the network
            }
        }
        if maximum iteration reached, or there
        has been no change in the number of cells
        in the network since the last iteration
        {
            Terminate the loop
        }
    }
    else
    {
        Introduce a percentage  $d\%$  of randomly
        generated cells
    }
}

```

Fig. 2 The modified algorithm opt-aiNet for Elgasir optimization

In order to determine the optimal domain delimiters for the membership function of each node in the regression tree. The membership function $f(x, a, b, c, d)$ is defined as:

$$f(x, a, b, c, d) = \begin{cases} 0 & \text{for } x < a \\ \frac{x-a}{b-a} & \text{for } a \leq x < b \\ 1 & \text{for } b \leq x < c \\ \frac{d-x}{d-c} & \text{for } c \leq x \leq d \\ 0 & \text{for } d < x \end{cases}$$

where

$$\begin{aligned} a &= i - n\sigma \\ b &= i - \alpha \\ c &= i + \alpha \\ d &= i + n\sigma \end{aligned}$$

where σ is the standard deviation, x is value of attribute A .

$\alpha = m\sigma$ where $m = 0.0001, 0.001, 0.01, 0.1$, n is a real number $n \rightarrow [0, \infty]$

The initial membership functions are selected to give fuzzy rules set that are equivalent to crisp rules set where $\sigma = 0$.

Figure.3 shows a generalised example of the tuning of membership function domain delimiters which shows Trapezoidal membership function f of domain $a \dots d$ that creates a fuzzy region around threshold i .

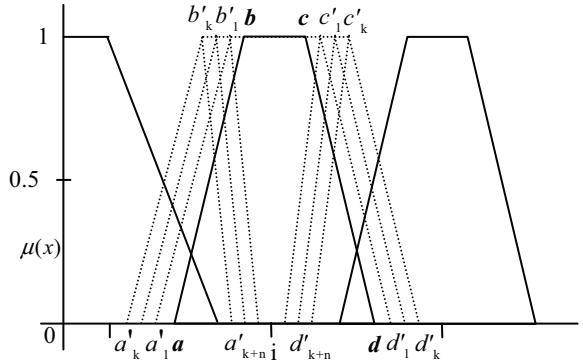


Fig .3 Optimization of Trapezoidal membership function

This optimization system is used to tune the position of the domain delimiters of each membership function, so the points (a, b, c, d) can be directed with the constraint $a \leq b \leq i \leq c \leq d$ to produce (a'_k, b'_k, c'_k, d'_k) . where k and n are natural numbers.

Four domain delimiters (a, b, c, d) are used to represent the membership function within The proposed Elgasir optimization framework as following:

Let Z is a complete set of domain delimiters for a fuzzy regression tree consisting of z branches.

$$Z = \{(a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2), \dots, (a_z, b_z, c_z, d_z)\}$$

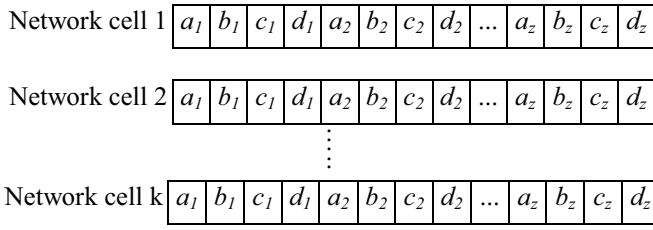


Fig. 4 Network cell representation for fuzzy regression tree.

Figure 4 shows the Network cell representation for membership functions domain delimiters of fuzzy regression tree where a, b, c, d real numbers.

The performance of the tree is evaluated by using the MSE (mean square error) which is defined in equation (5).

$$\text{Performance of fuzzy system} = \frac{\sum_{j=1}^n (\text{PredictedValue} - \text{ActualValue})^2}{n} \quad (5)$$

Where n = Number of records in the dataset.

V. EXPERIMENTS

A. Datasets

Experiments were undertaken on two real-valued datasets which known as Boston housing and Abalone datasets. These datasets have numeric outcomes and were retrieved from the well known UCI machine learning repository [29]. The Boston housing dataset consists of 506 records with thirteen continuous attributes and one binary-valued attribute and it has 229 unique outcome values. This dataset is used to predict the Median value of owner occupied homes. The Abalone dataset comprises 4177 instances featuring eight continuous attributes and it has 28 unique outcome values. This data set is concerned with predicting the age of abalone from physical measurements. The main characteristics of these datasets are presented in Table I.

Name	Number of instances	Unique values
boston housing dataset	506	229
abalone dataset	4177	28

B. Methodology

The cross validation approach was used for each experiment. The dataset is divided to n equal-sized disjoint blocks. Each block in turn is used to evaluate both the crisp and the fuzzy trees, which are generated by the remaining $n-1$ blocks. In this work 10-fold cross validation were applied to both datasets. A singular crisp regression tree was first created using the training dataset with various values of the CHAID regression parameters. These parameters include: merge threshold, split threshold, tree depth, node size and child node size. Each parameter was examined in turn to obtain the optimal crisp regression tree. The attribute which has the smallest adjusted p-value (the highest ranking) was

restricted from formulating the root of the second crisp regression tree. Hence, this tree will be different in representing the association of independent variables for the same dataset. This process was repeated until 5 crisp regression trees were created. The number of trees was empirically defined by a series of initial experiments which obtained the tree growth saturation threshold of 5. Empirically these experiments showed that no further improvement could be achieved from creating further regression trees of the domain. A set of IF-THEN rules were extracted from the combination of crisp CHAID regression trees. Then the fuzzification was applied to these set of IF-THEN rules.

TABLE II- THE OPT-AiNET PARAMETERS VALUES

NAME	VALUE
Iterations number	500
Population size	100
Clones number	15
Suppression threshold	0.175
Average error threshold	0.001
Percentage of newcomers	50%
Affinity proportional	75

The adapted version of an artificial immune network model (opt-aiNet) [2] was used to determine the most appropriate domain delimiters of Trapezoidal membership function. Table II presents a summary of the adapted opt-aiNet parameters values which were empirically defined by a series of initial experiments [2].

VI. RESULTS AND DISCUSSION

Tables III-VI show the results achieved by the proposed framework for both Abalone and Boston housing datasets. The results show a comparison of results between singular crisp regression trees and fuzzy regression tree forest.

TABLE III
SINGULAR CRISP REGRESSION TREES RESULTS
(ABALONE DATASET)

Tree number	Training dataset (error value)	Test dataset (error value)
1	4.5091	4.5138
2	4.4082	4.4162
3	4.4066	4.4151
4	4.5184	4.5431
5	4.5742	4.6029

TABLE IV
FUZZY REGRESSION TREE FORESTS RESULTS
(ABALONE DATASET)

No of trees	Training dataset (error value)	Test dataset (error value)
1	2.5169	2.8624
2	2.3061	2.4711
3	2.3041	2.4708
4	2.3537	2.4885
5	2.6150	2.9797

TABLE V
SINGULAR CRISP REGRESSION TREES RESULTS
(BOSTON HOUSING DATASET)

Tree number	Training dataset (error value)	Test dataset (error value)
1	21.1749	21.5082
2	20.6497	21.0358
3	20.6562	21.0436
4	20.9841	21.4084
5	21.8231	22.0471

TABLE VI
FUZZY REGRESSION TREE FORESTS RESULTS
(BOSTON HOUSING DATASET)

No of trees	Training dataset (error value)	Test dataset (error value)
1	13.3983	15.9891
2	13.2702	15.6417
3	13.2926	15.6801
4	13.3892	15.7990
5	13.9587	16.3765

For both Abalone and Boston housing datasets the tables III and V show the results of five singular crisp regression trees while the tables IV and VI present the results of combining up to 5 fuzzy regression trees. The tables present the mean square error which was obtained by Equation (5).

The results in Tables III-VI confirmed that the proposed method for creating multiple fuzzy regression trees improves the performance of prediction. The Abalone dataset results show that the combination of three fuzzy regression trees obtained the best result which reduced the error rate by 14% compared to singular fuzzy regression tree. The biggest improvement in performance for the proposed method with the Boston housing dataset was obtained by the combination of two fuzzy regression trees which reduced the error rate by 2% compared to singular fuzzy regression tree. The proposed framework is most effective on continuous attributes, which is reflected in the improvement of the Abalone dataset. As the number of fuzzy regression tree in the forest increases up to a certain threshold, the prediction accuracy also improves to degree, and once this threshold is reached, overfitting of the training dataset starts to occur, which is dependent on the type, value, distribution and nature of dataset.

VII. CONCLUSION

This study introduced a new framework to creating a fuzzy regression trees forests using the modified Elgasir algorithm. The CHAID algorithm was selected for this work used to induce crisp regression trees. The modified Elgasir algorithm was applied to generate fuzzy regression trees forests using Trapezoidal membership functions to represent tree nodes and the Takagi-Sugeno fuzzy inference technique to combine values down each path. The adapted version of an artificial immune network model (opt-aiNet) was used to optimize the membership functions of the proposed method. The results from two real-valued datasets show that combine fuzzy regression trees can improve the performance by incorporating information from different perspectives of the

same dataset. This framework offer promising results in using this kind of technique to reduce the error rate of single fuzzy regression trees. Further studies involve investigating the effects of more complex dataset characteristics in the performance of the proposed method, and comparing the results of the new approach with other approaches to inducing multiple fuzzy regression trees.

REFERENCES

- [1] Gasir, F. Bandar, Z. Crockett, K., Elgasir: An algorithm for creating Fuzzy Regression Trees. FUZZ-IEEE International Conference on Fuzzy Systems, Jeju Island, Korea, pp 332-337, 2009.
- [2] Gasir, F. Bandar, Z. Crockett, K. Crispin, A., Immune Engineering for Elgasir algorithm optimization. IEEE World Congress on Computational Intelligence, Barcelona, Spain, pp 1918-1924, 2010.
- [3] Gasir, F. Bandar, Z. Crockett, K., On Application of Artificial Immune System to Optimize Fuzzy Regression Trees. IEEE International Conference on Systems, Man, and Cybernetics, Istanbul, Turkey, pp 2442-2447, 2010.
- [4] Gasir, F. Bandar, Z. Crockett, K., The Effect of Datasets Characteristic in The Induction of Fuzzy Regression Trees Using Elgasir. The International Arab Conference on Information Technology, Benghazi, Libya, 2010
- [5] Fowdar, J. Crockett, K. Bandar, Z., On the Use of Fuzzy Trees for Solving Classification Problems with Numeric Outcomes. 2005 IEEE Conference on Fuzzy Systems, Nevada, USA, 2005.
- [6] Crockett, K., Bandar, Z. Fowdar, J. O'Shea, J., "Genetic Tuning of Fuzzy Inference within Fuzzy Classifier Systems," The International Journal of Expert Systems, Vol 23, No2, pp 63-82, 2006.
- [7] Quinlan, J.R., Induction of Decision Trees, Machine Learning, 1, pp. 81-106, 1986.
- [8] Quinlan, J.R., C4.5 program for machine learning, Morgan Kaufmann Publishers, 1993.
- [9] Quinlan, J.R., (2002) Is C5.0 better than C4.5; available at <http://www.rulequest.com/see5-comparison.html>.
- [10] Leland Wilkinson, Tree Structured Data Analysis: AID, CHAID and CART. Sun Valley, ID, Sawtooth/SYSTAT Joint Software Conference, 1992.
- [11] Servane Gey and Elodie Nedelec, Model Selection for CART Regression Trees, IEEE Transactions on Information Theory, 51, pp.658-670, 2005.
- [12] Morgan, J. N. Sonquist, J. A., Problems in the Analysis of Survey Data and a Proposal, American Statistical Association Journal 58, pp 415-434, 1963.
- [13] Kass, G. V., An Exploratory Technique For Investigating Large Quantities Of Categorical Data, Applied Statistics, 29(2) pp 119-127, 1979.
- [14] Wilkinson, L., Tree Structured Data Analysis: AID, CHAID and CART. Sun Valley, ID, Sawtooth/SYSTAT Joint Software Conference, 1992..
- [15] Gey, S. Nedelec, E., Model Selection for CART Regression Trees, IEEE Transactions on Information Theory, 51, pp.658-670, 2005.
- [16] Haskell, R. E., Regression Tree Fuzzy Systems, Proc. ICSC Symposium on Soft Computing, Fuzzy Logic, Artificial Neural Networks and Genetic Algorithms, University of Reading, Whiteknights, Reading, England, pp. B.1-B.6, 1996.
- [17] Takagi, T. Sugeno, M., Fuzzy identification of a system and its application to modeling and control, IEEE Trans. Syst. Man and Cybern., vol.15, pp.116-132, 1985.
- [18] Cutello, V. Nicosia, G. Oliveto, P. S. Romeo, M., On the Convergence of Immune Algorithms, FOCCI 2007, The First IEEE Symposium on Foundations of Computational Intelligence, 2007.
- [19] Campelo, F., Guimares, F.G., Igarashi, H., Ramirez, J.A., Noguchi, S., A modified immune network algorithm for multi-modal electromagnetic problems, IEEE Trans. Magn. 42 (April (4)) (2006) 1111-1114, 2006.
- [20] Walker, J.H. Garrett, S.M., Dynamic Function Optimization: Comparing the Performance of Clonal Selection and Evolution Strategies, Proceeding of Second International Conference on Artificial Immune Systems (ICARIS2003), Napier University, Edinburgh, UK, September 1-3 2003

- [21] Lee, Z-J. Lee, C-Y. Su, S-F., An immunity based ant colony optimization algorithm for solving weapon-target assignment problem, *Appl. Soft Comput.* 2 (August (1)) (2002) 39–47, 2002.
- [22] De Castro, L.N. Timmis, J., An Artificial Immune Network for Multimodal Function Optimisation. *Proc. Of IEEE World Congress on Evolutionary Computation*. Pp. 669-674, 2002.
- [23] Piero P. Bonissone, José Manuel Cadenas, M. Carmen Garrido, R. Andrés Díaz-Valladares: A fuzzy random forest. *Int. J. Approx. Reasoning* 51(7): 729-747, 2010.
- [24] C. Marsala and B. Bouchon-Meunier. Forest of fuzzy decision trees. In M. Mareš, R. Mesiar, V. Novák, J. Ramík, and A. Štěpánová, editors, *Proceedings of the Seventh International Fuzzy Systems Association World Congress*, volume 1, pages 369–374, Prague, Czech Republic, June 1997.
- [25] C. Z. Janikow and M. Faifer. Fuzzy decision forest. In *Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS'00)*, p 218–221, July 2000.
- [26] Crockett, K, Bandar, Z. Mclean D. Growing a Fuzzy Decision Forest. *The 10th IEEE International Conference on Fuzzy Systems - FUZZ-IEEE 2001 Melbourne Australia*, ISBN 0-7803-7294-X, December 2001.
- [27] L. Breiman, Random forests, *Machine Learning* 45 (1) (2001) 5–32.
- [28] Abu-halaweh, Nael Mohammed, "Integrating Information Theory Measures and a Novel Rule-Set-Reduction Technique to Improve Fuzzy Decision Tree Induction Algorithms" (2009). *Computer Science Dissertations*. Paper 48. http://digitalarchive.gsu.edu/cs_diss/48
- [29] Blake, C. L. and C. J. Merz, *UCI Repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science, 1998.