

# Automatic Generation of Fuzzy Classification Rules Using Granulation-Based Adaptive Clustering

Mohammed Al-Shammaa\*, Maysam F. Abbod

Department of Electronic and Computer Engineering  
Brunel University

London, United Kingdom

Mohammed.Al-Shammaa@brunel.ac.uk, Maysam.Abbod@brunel.ac.uk

**Abstract**— A central problem of fuzzy modelling is the generation of fuzzy rules that fit the data to the highest possible extent. In this study, we present a method for automatic generation of fuzzy rules from data. The main advantage of the proposed method is its ability to perform data clustering without the requirement of predefining any parameters including number of clusters. The proposed method creates data clusters at different levels of granulation and selects the best clustering results based on some measures. The proposed method involves merging clusters into new clusters that have a coarser granulation. To evaluate performance of the proposed method, three different datasets are used to compare performance of the proposed method to other classifiers: SVM classifier, FCM fuzzy classifier, subtractive clustering fuzzy classifier. Results show that the proposed method has better classification results than other classifiers for all the datasets used.

**Keywords**—Fuzzy systems; granular computing; data clustering; data classification.

## I. INTRODUCTION

There have been many approaches proposed to achieve the task of classifying data into predefined categories. In the field of computational intelligence, fuzzy classification systems have been used in many practical classification problems such as: medical applications [1]-[2], fingerprint classification [3], industry [4], security [5] or financial applications [6], etc. Fuzzy systems are characterized by their interpretability and ability to deal with incompleteness of knowledge.

The central problem of fuzzy modelling is the generation of fuzzy rules that fit the data to the highest possible extent. The task of generating and learning fuzzy rules from numerical data has been addressed by different approaches, some examples are: heuristic procedures [7], neuro-fuzzy techniques [8], clustering methods [9]-[11], data mining techniques [12], and genetic algorithms [13].

For many applications, data clustering is used for the generation of fuzzy membership functions that describe the data. Data clustering involves grouping data points that have a certain level of similarity into clusters. However, this similarity does not necessarily represent the discrimination factor of the desired classification. That is, a given cluster may contain data

samples that belong to different classes. Furthermore, data clustering usually requires pre-defining some clustering parameters (i.e. number of clusters) that affect the clustering process, hence the resultant fuzzy rules. Finding the best clustering parameters for a given application is not a straightforward process.

There have been several attempts to automatically generate fuzzy rules from data. An alternative method that is based on subtractive clustering optimized using genetic algorithm is presented in [14]. Another method that achieves structure identification and parameter estimations of fuzzy inference systems via unsupervised learning is presented in [15].

In this study, a method for fuzzy rules generation is presented. The proposed method uses a modified data granulation algorithm to create data clusters and selects the best granulation level for the desired clustering.

Information granulation is a computing paradigm of information processing that handles information as complex entities called information granules. Information granulation mimics human thinking and reasoning by grouping similar items or elements into granules at a certain level of resolution. Information granulation provides a framework of knowledge extraction from data at different levels of abstraction/precision called granulation levels [16][17].

This paper is arranged as follows. A brief description of fuzzy rule generation methods is given in section II. The proposed method is presented in section III. Section IV shows experimental results and comparison of the proposed method to other competence methods. The conclusion is given in section V.

## II. FUZZY RULES GENERATION

Fuzzy inference systems (FIS) consist of three main components: a rule base, a database that defines the fuzzy membership functions and a reasoning mechanism that performs the inference process.

Information required to design FIS and create its rules can be obtained from the knowledge of human experts. However, for complex systems with large number of inputs and fuzzy variables, some computational methods, such as data clustering

---

\* PhD student, sponsored by the Higher Committee for Education Development (HCED), Iraq.

and granular computing can be used to generate fuzzy rules from data automatically.

### A. Data Clustering

Data clustering involves grouping data points that have a certain level of similarity into clusters. Input membership functions can be obtained by projecting data clusters onto the dimension of the inputs. Usually, clustering methods require specifying some clustering parameters such as the number of clusters prior to the clustering process. These parameters greatly affect the clustering process.

One of the widely used clustering algorithms is Subtractive Clustering (SC) [10]. SC selects data points that have high “potentials” to be a cluster center. The potential of a data point, which is defined as the density of the neighboring data points, is a function of the Euclidean distances to all other data points. SC does not require specifying the number of clusters but it requires specifying other clustering parameters such as radii of influence, squash factor, accept ratio and reject ratio.

Another clustering algorithm is fuzzy c-mean (FCM) [18]. Unlike hard clustering where each data element belongs to exactly one cluster, in FCM, each data point belongs to a cluster to some degree that is specified by a membership grade. In fuzzy clustering, data elements can belong to more than one cluster and associated with each element is a set of membership levels.

### B. Granular Computing

Granulation is a computing paradigm of information processing that handles information as complex entities called information granules. It provides a framework that encourages processing data in a way that extract knowledge from data at various levels of resolution or scales. Hence, it embraces all methods that have flexibility and adaptability in the level of knowledge extraction.

One method of performing the granulation of data is the data granulation algorithm [17] which is achieved by an iterative process involving two steps:

- Form a new information granule by merging the two most compatible information granules together.
- Repeat the process of forming new information granules until the desired data abstraction level is reached.

## III. GRANULATION-BASED ADAPTIVE CLUSTERING

In this study, a modified data granulation algorithm is used to construct clusters for the task of data classification. One modification is that each cluster contains only data samples that belong to the same class. In addition, each cluster has its own set of weights that is used to compute weighted Euclidean distance at a given level of granulation. The distance function uses a weighting variable for each direction of each dimension (i.e. input variable). Thus, the resulted clusters may have different spread widths in different directions. The set of weights of each cluster are computed based on the relative distances between the cluster and each cluster of differing

class. The farther these clusters in a given direction, the easier for the cluster to grow in that direction.

Fig. 1-3 show an illustrative example of data granulation of 20 2-D data samples in two classes (10 points for each) at three different levels of granulation – from finest granulation in Fig.1 to coarsest granulation in Fig.3. Coarser granulation results in less number of clusters.

The pseudo-code of the proposed method is given in Algorithm 1 and a flowchart of the method is shown in Fig. 4. Granulation-Based Adaptive Clustering (GBAC) creates data clusters at different levels of granulation and selects the best clustering results according to its specificity and cardinality. It involves a process of merging two clusters into new cluster with coarser granulation. This process is repeated iteratively until no further granulation is possible.

Initially, each data point is considered as a granule (data cluster). Any two granules of the same class are merged into a new cluster if the distance between them is less than a certain value (L) that reflects the granulation level. Decreasing L results in finer granulation while increasing it results in coarser granulation. Then, the centroid of the new cluster is computed and the merging process continues until the smallest distance of any two clusters of the same class is larger than L.

L is gradually increased from the smallest distance between two points to the largest distance to create clusters at different granulation levels. For each granulation level, data granulation is evaluated and the best granulation is chosen. The evaluation of clustering includes two main measures: specificity and cardinality. Specificity measures how well data points of each class are separated from those of differing class. Specificity measure encourages towards finer granulation, hence smaller clusters. On the other hand, cardinality measures how well clustering is successful in containing more similar points into the same cluster, that is encouraging coarser granulation and larger clusters. The proposed method selects the best clustering that compromise both measures. Cardinality is simply the average of sizes of clusters while specificity is computed as the reciprocal of average spread of each cluster. Cluster spread is the average distance of cluster points to its center. That is:

$$\text{Specificity} = K / \sum_k (\sum_j D_{kj} / S_k) \quad (1)$$

where K is the number of clusters,  $D_{kj}$  is the distance between center of cluster k and data point j within the cluster and  $S_k$  ( $k=1,2,3,\dots,K$ ) is the size of (number of data points in) cluster k.

As it can be noted in Algorithm 1, GBAC does not require predefining any parameters including number of clusters, although it can be easily constrained to a given range of granulation levels or number of clusters.

Final resulted clusters are used to generate fuzzy rules by projecting the clusters onto each dimension. At this stage, data clusters with a few data samples can be considered outliers. Since the spread width of the cluster may differ in each direction of the same dimension, the resulted fuzzy membership function may be asymmetric. A Gaussian

membership function with different standard deviation parameters for each side is used. Further optimization of premise part parameters is performed in a second stage using euro-fuzzy training (ANFIS model). The identification of consequence part parameters is performed at this stage as well. A Sugeno fuzzy model with linear consequence part is considered. Consequent part parameters are generated using linear regression.

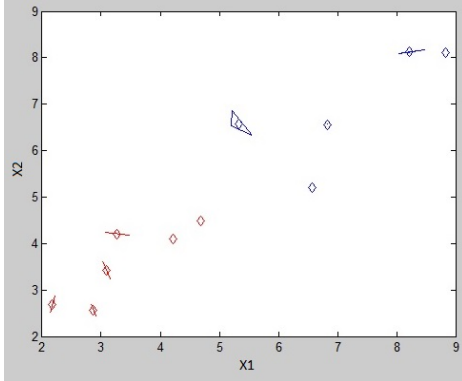


Fig. 1. Illustrative example of data granulation,  $L=0.3$ .

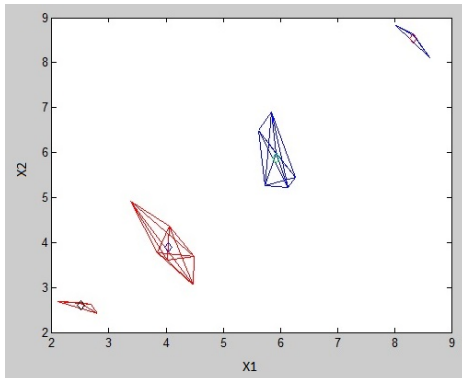


Fig. 2. Illustrative example of data granulation,  $L=1$ .

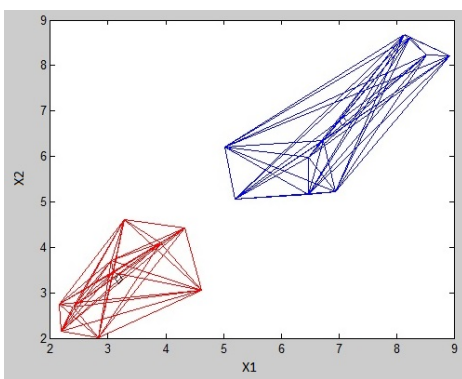


Fig. 3. Illustrative example of data granulation,  $L=3$ .

#### Algorithm 1. Data Clustering using GBAC

Input: Data:  $N \times I$  matrix,  $N$  is the number of samples,  $I$  is the number of input variables; Labels:  $N \times 1$

Output:  $C_k$  ( $k=1,2,3 \dots K$ ):  $K$  cluster centers,  $K$  is determined by the algorithm;  $E_{k,j}$  ( $i=1,2,3 \dots J_k$ ): data points that belong to cluster  $k$ ;  $W_{k,I}$ : set of weights for each cluster.

- 1: set each point as a granule
- 2: for each level of granulation  $L$
- 3: for each granule  $G_k$
- 4: find the nearest same-label granule  $G_m$  and compute the distance  $D_{km}$  using weighted Euclidean distance function with weight set  $W_k$
- 5: if the distance  $D_{km}$  is less or equal to  $L$
- 6: merge  $G_m$  into  $G_k$
- 7: compute new center  $C_k$
- 8: compute new weight set  $W_k$
- 9: repeat steps 4-8 until no two granules have distance less than  $L$
- 10: compute Specificity  $S_l$  and Cardinality  $C_l$
- 11: if  $S_l$  is greater than best specificity  $BS$  and  $C_l$  is greater than best cardinality  $BC$
- 12: set  $BS = S_l$ ,  $BC = C_l$ , and best granulation is  $G_k$  ( $k=1,2,3 \dots K$ ) and best weight sets are  $W_k$  ( $k=1,2,3 \dots K$ )
- 13: increase  $L$  and repeat steps 3-10 until  $L$  is maximum

#### IV. EXPERIMENTAL STUDY

To evaluate performance of the proposed method, three different datasets are used to compare its performance to other classifiers: SVM classifier, FCM fuzzy classifier, subtractive clustering fuzzy classifier.

Each data set is divided into two partitions: modeling (80%) and testing (20%). Modeling partition is used by GBAC to build a FIS and to fine-tune the resulted FIS using a neuro-fuzzy system. The neuro-fuzzy system used is "Adaptive Network-based Fuzzy Inference System" (ANFIS). ANFIS, introduced by Jang (1993), is a method for tuning FIS antecedent and consequent parameters through a hybrid learning rule which combines the back-propagation gradient descent method and least-squares method [19]. Two measures are used to evaluate the classification performance: accuracy and area under curve (AUC). These are computed for each data set using the testing partition. Results are compared to those of SVM classifier and ANFIS models generated FCM (with different number of clusters) and subtractive clustering method (with different radii values). Testing results for each model are the average results of 20 evaluations each with different random data sampling.

The three data sets we used are: Pima data set, BUPA data set and Haberman's survival data set, and can be found at University of California in Irvine (UCI) Machine Learning Repository [20]. All data were normalized before being used in training or testing.

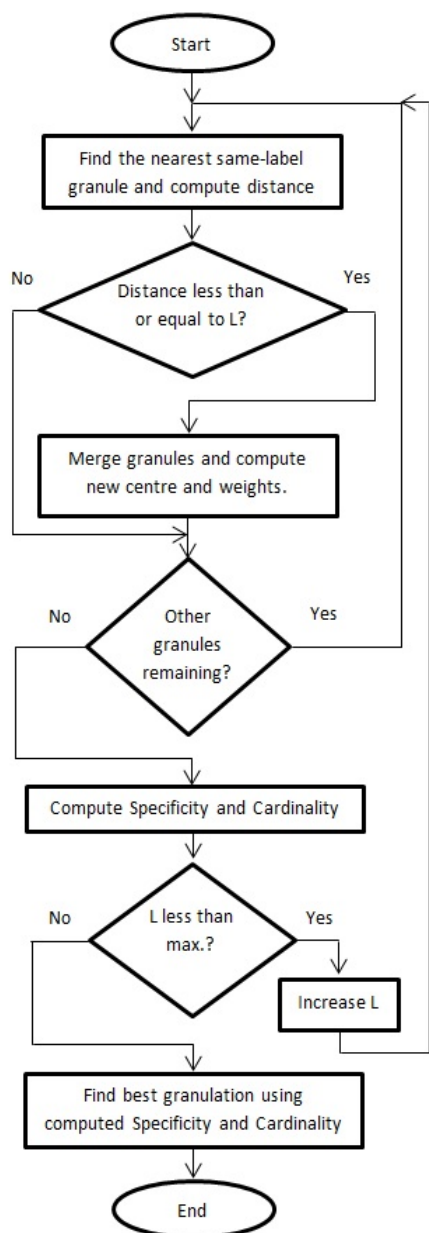


Fig. 4. Flowchart of the proposed method.

Pima data set contains 768 records of female Pima Indian patients (at least 21 years old) from USA tested for diabetes. It has 8 input variables and a class distribution of 268/500. BUPA data set contains 345 records of liver patients from USA. It has 6 input variables and a class distribution of 200/145. Haberman's dataset contains 306 cases from a study conducted the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. It has 3 input variables and a class distribution of 81/225.

Classification results for the three datasets are given in Tables I-III. GBAC performance is compared to 3 FCM models (2, 3 and 5 clusters), 5 SC models (radii of 0.3, 0.5, 0.7, 1 and 1.2) and 3 SVM models (with linear, Gaussian Radial

Basis Function RBF kernel and Multilayer Perceptron MLP kernel). Fig. 5-13 show the receiver operating characteristic (ROC) curves of GBAC, FCM and SC classifiers for the three datasets.

Results show that GBAC has better classification accuracy and AUC than other compared classifiers for all the datasets. However, compared to best FCM or SC model, the improvement in terms of accuracy and AUC is not significant, which makes the main advantage of GBAC is its ability to perform data clustering without the requirement of predefining any parameters including number of clusters.

TABLE I. CLASSIFICATION RESULTS FOR PIMA DATA SET

<i>Classifier</i>	<i>Accuracy</i>	<i>Area Under Curve</i>
GBAC	78.64%	82.04%
FCM 2	77.44%	80.84%
FCM 3	77.95%	79.60%
FCM 5	76.30%	77.67%
SC 0.3	75.29%	74.30%
SC 0.5	77.18%	80.59%
SC 0.7	77.21%	80.50%
SC 1	78.08%	81.96%
SC 1.2	77.92%	81.61%
SVM Linear	75.03%	73.97%
SVM RBF	73.80%	73.77%
SVM MLP	68.90%	66.77%

TABLE II. CLASSIFICATION RESULTS FOR BUPA DATA SET

<i>Classifier</i>	<i>Accuracy</i>	<i>Area Under Curve</i>
GBAC	75.29%	76.04%
FCM 2	73.90%	75.34%
FCM 3	73.01%	72.17%
FCM 5	71.69%	70.22%
SC 0.3	68.75%	64.49%
SC 0.5	73.75%	72.50%
SC 0.7	74.41%	74.46%
SC 1	74.56%	73.86%
SC 1.2	74.41%	74.82%
SVM Linear	67.57%	67.69%
SVM RBF	66.32%	64.75%
SVM MLP	61.54%	61.44%

TABLE III. CLASSIFICATION RESULTS FOR HABERMAN'S DATA SET

<i>Classifier</i>	<i>Accuracy</i>	<i>Area Under Curve</i>
GBAC	77.24%	67.11%
FCM 2	74.22%	66.85%
FCM 3	76.81%	63.16%
FCM 5	74.83%	63.11%
FCM 7	75.00%	60.81%
SC 0.3	72.76%	53.13%
SC 0.5	74.14%	57.51%
SC 0.7	75.78%	63.83%
SC 1	76.81%	67.99%
SC 1.2	76.55%	66.77%
SVM Linear	74.66%	59.40%
SVM RBF	75.52%	61.68%
SVM MLP	71.55%	58.45%

### V. CONCLUSION

The flexibility of granular computing in handling data at different levels of abstraction/precision makes it a natural choice for fuzzy rules extraction. In this study, we presented a method for fuzzy rules generation that is based on granular computing. It performs data clustering without requiring predefinition of any parameters such as the number of clusters. The proposed method performs data clustering at different levels of granulation and selects the best clustering to generate fuzzy rules. Classification accuracy and AUC of the proposed method were compared to those of other classifiers: SVM classifier, FCM fuzzy classifier and SC fuzzy classifier. Results show that GBAC has better classification accuracy and AUC than other compared classifiers for the three datasets used. This suggests that using granular computing in fuzzy rule generation is a promising approach.

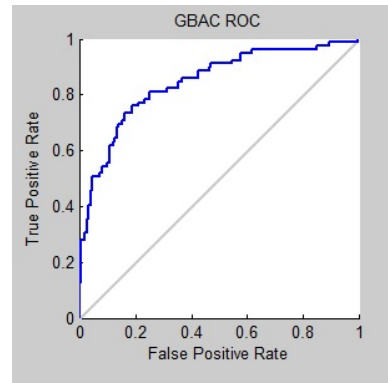


Fig. 5. GBAC ROC curve for Pima Dataset.

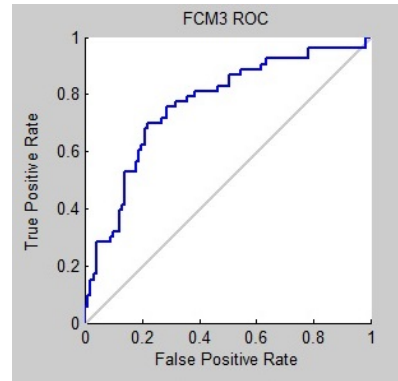


Fig. 6. FCM ROC curve for Pima Dataset

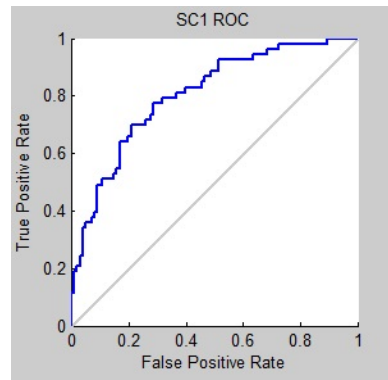


Fig. 7. SC ROC curve for Pima Dataset

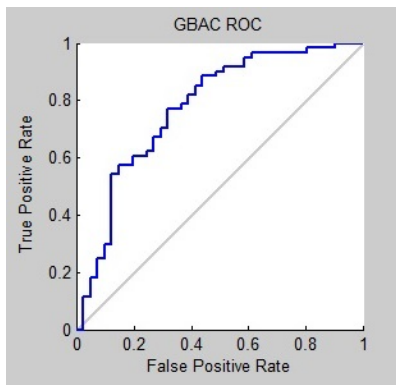


Fig. 8. GBAC ROC curve for Bupa Dataset

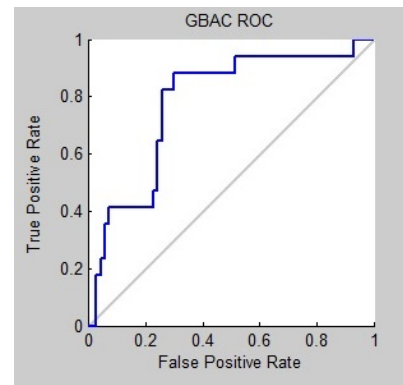


Fig. 11. GBAC ROC curve for Haberman's Dataset

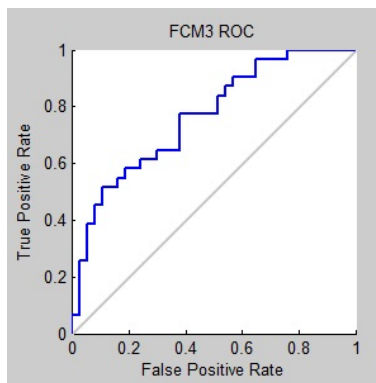


Fig. 9. FCM ROC curve for Bupa Dataset

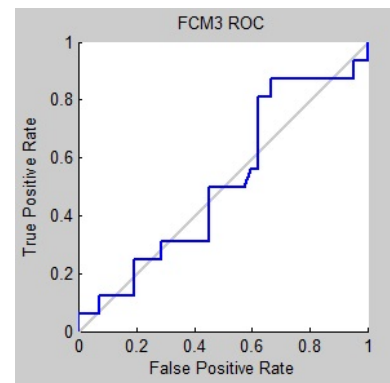


Fig. 12. FCM ROC curve for Haberman's Dataset

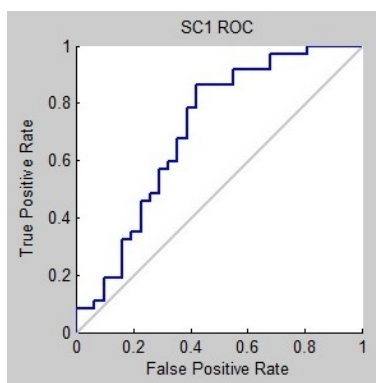


Fig. 10. SC ROC curve for Bupa Dataset

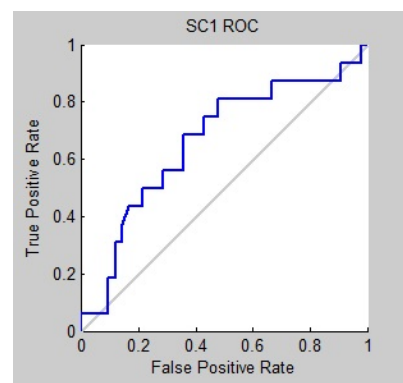


Fig. 13. SC ROC curve for Haberman's Dataset

## REFERENCES

- [1] S., J. Antonio, et al. "Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system." *Applied Soft Computing* 20 (2014): 103-111.
- [2] J., Sanz, et al. "A case study on medical diagnosis of cardiovascular diseases using a genetic algorithm for tuning fuzzy rule-based classification systems with interval-valued fuzzy sets." *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, 2011 IEEE Symposium on. IEEE, 2011.
- [3] G., Mikel, et al. "A preliminary study on fingerprint classification using fuzzy rule-based classification systems." *Fuzzy Systems (FUZZ-IEEE)*, 2014 IEEE International Conference on. IEEE, 2014.
- [4] A. F., Sallama, and M. F. Abbod. "Neuro-fuzzy system for power generation quality." *Innovative Smart Grid Technologies-Middle East (ISGT Middle East)*, 2011 IEEE PES Conference on. IEEE, 2011.
- [5] T. P. Fries, "Evolutionary optimization of a fuzzy rule-based network intrusion detection system." *Fuzzy Information Processing Society (NAFIPS)*, 2010 Annual Meeting of the North American. IEEE, 2010.
- [6] J., Sanz, et al. "A Compact Evolutionary Interval-Valued Fuzzy Rule-Based Classification System for the Modeling and Prediction of Real-World Financial Applications with Imbalanced Data."
- [7] G. S. Abe, and M. S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification", *IEEE Trans. Fuzzy Systems*, vol. 3, no. 1, pp. 18-28, 1995.
- [8] D. Nauck, and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data", *Fuzzy Sets and Systems*, vol. 89, no. 3, pp. 277-288, 1997.
- [9] J. A. Roubos, M. Setnes, and J. Abonyi, "Learning fuzzy classification rules from labeled data", *Inform. Sci.*, vol. 150, no. 1-2, pp. 77-93, 2003.
- [10] S. L. Chiu, "Extracting fuzzy rules from data for function approximation and pattern classification," *Fuzzy Information Engineering: A Guided Tour of Applications*, pp. 1-10, 1997.
- [11] A. F. Gómez-Skarmeta, M. Delgado, and M. A. Vila, "About the use of fuzzy clustering techniques for fuzzy model identification," *Fuzzy Sets Syst.*, vol. 106, no. 2, pp. 179-188, Sep. 1999.
- [12] Y. Chung Hu, and G. Hshiang Tzeng, "Elicitation of classification rules by fuzzy data mining", *Engrg. Appl. of Artificial Intelligence*, vol. 16, no. 7-8, pp. 709-716, 2003.
- [13] L. Snchez, I. Couso, J. A. Corrales, O. Cordn, M. J. Del Jesus, and F. Herrera, "Combining GP operators with SA search to evolve fuzzy rule-based classifiers", *Inform. Sci.*, vol. 136, no. 1-4, pp. 175-191, 2001.
- [14] M., Al-Shammaa, and M. F. Abbod, "Automatic Generation of Fuzzy Classification Rules from Data," in *Recent Advances in Neural Networks and Fuzzy Systems*, Venice, Italy, March 2013, pp. 74-79.
- [15] Er, Meng Joo, and Yi Zhou. "Automatic generation of fuzzy inference systems via unsupervised learning." *Neural Networks* 21.10 (2008): 1556-1566.
- [16] L. A. Zadeh, "Towards a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic," *Fuzzy Sets and Systems*, Vol. 19, 1997, pp. 111-127.
- [17] G. Panoutsos, M. Mahfouf, A neural-fuzzy modelling framework based on granular computing: concepts and applications, *Fuzzy Set. Syst.* 161 (21) (2010)2808-2830.
- [18] J.C., Bezdec, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York, 1981.
- [19] J. Jang, "ANFIS: adaptive-network-based fuzzy inference system." *Systems, Man and Cybernetics*, IEEE Transactions on 23.3 (1993): 665-685.
- [20] A. Frank and A. Asuncion, UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science, 2010.