12-2017

Leveraging the trade-off between accuracy and interpretability in a hybrid intelligent system

Di WANG

Chai QUEK

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Chunyan MIAO

Geok See NG

See next page for additional authors

Leveraging the Trade-off between Accuracy and Interpretability in a Hybrid Intelligent System

Di Wang[†], Chai Quek[‡], Ah-Hwee Tan[†‡] and Chunyan Miao[†‡], Geok See Ng[§], You Zhou[¶*]
[†]Joint NTU-UBC Research Center of Excellence in Active Living for the Elderly
[‡]School of Computer Science and Engineering
Nanyang Technological University, Singapore
[§]School of Information Systems Technology and Design
Singapore University of Technology and Design, Singapore
[¶]College of Computer Science and Technology
Jilin University, Changchun, China
*Corresponding author email: zyou@jlu.edu.cn

***Abstract*—Neural Fuzzy Inference System (NFIS) is a widely adopted paradigm to develop a data-driven learning system. This hybrid system has been widely adopted due to its accurate reasoning procedure and comprehensible inference rules. Although most NFISs primarily focus on accuracy, we have observed an ever increasing demand on improving the interpretability of NFISs and other types of machine learning systems. In this paper, we illustrate how we leverage the trade-off between accuracy and interpretability in an NFIS called Genetic Algorithm and Rough Set Incorporated Neural Fuzzy Inference System (GARSINFIS). In a nutshell, GARSINFIS self-organizes its network structure with a small set of control parameters and constraints. Moreover, its autonomously generated inference rule base tries to achieve higher interpretability without sacrificing accuracy. Furthermore, we demonstrate different configuration options of GARSINFIS using well-known benchmarking datasets. The performance of GARSINFIS on both accuracy and interpretability is shown to be encouraging when compared against other decision tree, Bayesian, neural and neural fuzzy models.**

***Index Terms*—interpretability, neural fuzzy inference system, genetic algorithm, rough set, interpretable rules**

I. Introduction

Neural Fuzzy Inference System (NFIS) [1] or also widely known as Fuzzy Neural Network (FNN) synthesizes the human cognitive and reasoning processes by tolerating imprecise information and handling ambiguous situations. NFIS solves complex problems using linguistic models consisting of highly intuitive and easily comprehensible fuzzy rules. The hybridization performs non-fuzzy or fuzzy operations in different layers of the network to integrate both the learning aptitude of neural networks and the transparency of fuzzy systems.

To better preserve the semantic meanings of the linguistic models, certain level of the rule base's legibility has to be guaranteed. The interpretability improvement is "regarded as one of the most important issues in data-driven fuzzy modeling" [2]. There are two major approaches proposed in the literature to improve a model's interpretability. One is to reduce the complexity after the construction of the model and the other is to define constraints before the construction process. For the first approach, many methods have been proposed, such as rule aggregation [3], rule removal [4], rule transformation [5], feature selection [6], and knowledge reduction (on both rules and features) [7]–[9]. The second approach mainly focuses on controlling the quality and quantity of the derived membership functions [10], [11] and also focuses on defining constraints on both membership functions and rules [12], [13]. Only a few prior studies [14]–[16] provide options or device parameters to leverage the trade-off between accuracy and interpretability.

In this paper, we illustrate how we leverage the trade-off between accuracy and interpretability in an NFIS called Genetic Algorithm and Rough Set Incorporated Neural Fuzzy Inference System (GARSINFIS). In a nutshell, GARSINFIS self-organizes its network structure with a small set of control parameters and constraints. Moreover, it employs and fine-tunes the inference rule base, which is autonomously derived by an iterative clustering algorithm called Genetic Algorithm based Rough Set Clustering (GARSC). Because knowledge reduction is performed and the formations of clusters are iteratively optimized, the derived rule base is highly interpretable and reliable. For performance evaluations, we conduct experiments on well-known benchmarking datasets of different complexity to demonstrate different configuration options of GARSINFIS. The performance of GARSINFIS on both accuracy and interpretability is shown to be encouraging when compared against other decision tree, Bayesian, neural and neural fuzzy models.

The rest of this paper is organized as follows. Section II introduces the GARSC clustering algorithm. Section III presents the system architecture of GARSINFIS, which employs and fine-tunes the fuzzy rules derived by GARSC. Section IV reports the experimental results of applying GARSINFIS in different configurations on the benchmarking datasets. Section V concludes this paper and proposes future work.

This research is supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative. This research is also supported in part by the Science & Technology Development Foundation of Jilin Province under grant No. 20160101259JC and the National Science Fund Project of China No. 61772227.

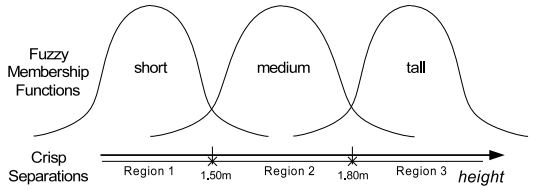

Fig. 1. Illustration of transferring crisp membership functions to fuzzy ones.

II. GARSC: Genetic Algorithm based Rough Set Clustering

GARSC [17] incorporates the advantages of both genetic algorithm [18] and rough set theory [19]. Specifically, we employ genetic algorithm to determine optimal or suboptimal solutions and use rough set theory to perform knowledge reduction [20]. Based on rough set knowledge reduction, any categorical inference rule set can be greatly reduced without losing any essential knowledge. This great property of rough set theory can be really helpful in improving the comprehensiveness of a set of inference rules [21], i.e., reducing the number of retained features, the number of employed rules, and the number of arguments kept in each inference rule. Please note that the crisp rules reducted by rough set approximations are transformed into fuzzy ones by deriving Gaussian fuzzy membership functions accordingly (See Fig. 1). Specifically, assume in dimension x, we define $n-1$ number of separation boundaries, then x is discretized into n regions. Therefore, the determination of a Gaussian type fuzzy membership function $f_{G_i}(x) = \exp(-\frac{\|x-c_i\|^2}{2\sigma_i^2})$ only requires the computation of mean c_i and standard deviation σ_i of all the data points in the ith region x_i. Then, the generated fuzzy rules are employed by GARSINFIS (see Section III) for performance evaluation of the underlying solution.

The step of transforming the crisp membership functions to fuzzy ones is necessary to better deal with the non-overlapping in crisp separations adopted by rough set theory. Instead, we employ fuzzy membership functions to tolerate imprecise information and better deal with unforeseen circumstances. This particular step of knowledge transfer naturally prevents the resulting fuzzy membership functions from separating or overlapping too much with their neighbours, which makes the fuzzy rules more interpretable. Moreover, as the membership functions are generated in each individual dimension without normalization and transformation, the semantic meanings of the associated fuzzy linguistic labels are greatly reserved. The necessity of transformation from crisp membership functions to fuzzy ones is also empirically shown in Section IV-A.

A. Predefined Discretization Constraints

Before introducing GARSC in detail, we first define a couple of constraints being applied on data discretization. The first constraint is the maximum number of separation boundaries allowed in each dimension. It is easy to infer that this constraint subsequently defines the maximum number of fuzzy membership functions might be formulated in each dimension. Nonetheless, the actual number of fuzzy membership functions derived is also affected by the knowledge reduction process. In any dimension, the minimum number of separation boundaries actually in use is zero, which denotes that the corresponding dimension is not included in the reducted inference rule base. Furthermore, this constraint should not be set to a large value so as to avoid the employment of a large number of fuzzy membership functions, which degrade the interpretability of the overall model.

The second constraint is on the minimum distance has to be fulfilled between any neighbouring separation boundaries in the same dimension. This constraint ensures the relatively high level of generation possessed by the derived fuzzy membership functions. As such, any neighbouring membership functions are well separated. We use *mindis* to denote this minimum distance requirement and present its definition as follows:

$$\textit{mindis}_i = \frac{ub_i - lb_i}{\max(\textit{nop}_i, M)}, \tag{1}$$

where i denotes the ith dimension; ub_i and lb_i denote the upper and lower boundaries of the value range in the ith dimension, respectively; $\textit{nop}_i$ denotes the total number of different values in the ith dimension that for each corresponding conditional attribute value, its associated decision attribute has more than one values; and M denotes the predefined minimum number of separation boundaries in each dimension, which is assigned to 10 unless specified otherwise.

B. Attribute and Rule Removal

In rough set theory, a decision table is independent when all its dispensable attributes have been removed. Therefore, we can obtain an independent decision table by performing attribute reduction to find the reduct of the original decision table with the minimum cardinality. If in all rules, some attributes are always dispensable, they shall be removed from the reasoning process. As such, we actually perform feature selection along the knowledge reduction process.

The reduction of decision rules is similar to attribute reduction. Besides the merging of duplicate rules, an inference rule is dispensable if and only if the performance of the resulting rule base does not decline after the rule being removed. This removal procedure is often denoted as the pruning of redundant rules. Furthermore, rules share the same conditional attribute values but differ in the decision attribute are named inconsistent rules. The removal of these rules is required to preserve the integrity of the inference rule base. The confidence of the kth rule is computed as follows:

$$\textit{conf}(k) = \min\left(\frac{\textit{card}(U_i(k_i) \cap d_k)}{\textit{card}(U_i(k_i))}\right), \forall i \in C, \tag{2}$$

where *card* computes cardinality; U_i denotes the union function of decision attributes of each individual rule in the underlying decision table that shares the same value on the ith attribute; k_i denotes the value of the ith attribute of the kth rule; and d_k denotes the decision value of the kth rule.

Within each inconsistent rule set, only one rule should be kept by following three selection criteria: (i) preserve the rule

that has the maximum confidence value; (ii) if confidence value ties, preserve the rule that covers the most number of data samples; and (iii) if the number of data samples still ties, preserve a random selected rule with equal probability.

C. Commonly Adopted Strategies in Genetic Algorithms

In genetic algorithms, the number of chromosomes exist in one generation is defined by the population size. Therefore, to evaluate more number of solution candidates, we can set a larger population size.

GARSC uses real numbers to compose chromosomes. Specifically, each gene of a chromosome represents a separation boundary in the corresponding dimension. Please note that although GARSC confines the maximum number of separation boundaries allowed in each dimension, the actual number of partitions in use varies, i.e., chromosomes (comprising of separation boundaries in all dimensions) in GARSC do not have a fixed length.

When producing a new generation of chromosomes, GARSC applies the elitism replacement strategy. Specifically, the elitism ratio $\mu \in [0, 1)$ determines how many highly fit chromosomes in the current generation $P(t)$ shall be kept in the next generation $P(t+1)$. Generally speaking, to avoid domination of certain species especially in the early generations, μ is normally set to relatively small values.

The stopping criterion of GARSC is defined as when GA reaches the pre-determined number of generations. This generation number should be set carefully to allow GA converge.

D. Evaluation of the Fitness of Chromosome

The fitness evaluation function examines the performance of the corresponding chromosome. Because the fitness function characterizes the optimal solution that GA tries to search for, it is often considered as the most important module in GA. Because the aim of GARSC is to derive comprehensive inference rules without degrading accuracy, we integrate both interpretability and accuracy terms in its fitness function (see (3)). Specifically, capital letters are used to denote constant values and small letters are used to denote variables. Terms 1 and 5 in (3) relate to accuracy and the remaining terms relate to interpretability. A chromosome with smaller fitness value is a better solution candidate to the underlying problem.

$$\begin{aligned} f(x) \quad = \quad & \underbrace{\tau_1(1-a)\frac{\mathit{NOD}}{\mathit{NOF}}}_{1} + \underbrace{\tau_2\frac{\mathit{nof}}{\mathit{NOF}}}_{2} + \underbrace{\tau_3\frac{\mathit{nor}}{\mathit{NOD}}}_{3} \\ & + \underbrace{\tau_4\frac{\mathit{noa}}{\mathit{NOF}\cdot\mathit{NOD}}}_{4} + \underbrace{\tau_5\frac{\mathit{mse}}{\mathit{NOF}}}_{5}, \end{aligned} \tag{3}$$

where x denotes the chromosome under evaluation; $\tau_1, \ldots, \tau_5$ denote the pre-determined coefficient values; a denotes the accuracy of solution x on the underlying data set; *NOD* denotes the number of data elements; *NOF* denotes the number of dimensions; *nof* denotes the number of features (dimensions) included in the inference rule base; *nor* denotes the number of rules in the inference rule base; *noa* denotes the aggregated number of arguments in the antecedent part of all rules; and mse denotes mean squared error:

$$\mathit{mse} = \frac{1}{\mathit{NOD}}\sum_{i=1}^{\mathit{NOD}}(y_i - \hat{y}_i)^2, \tag{4}$$

where y_i denotes the value of prediction and $\hat{y}_i$ denotes the value of ground truth.

E. Selection of Parents to Produce Offspring

During the production of offspring to be evaluated in the next generation, each pair of parents are selected from the current generation based on their fitness values. Generally speaking, parents normally have relatively higher fitness values than those not being selected. Among all parent selection strategies, we adopt tournament selection [22], in which the competition among candidates can be easily regulated by both the tournament size m and the selection probability s.

Prior to the selection of two parents to produce offspring by means of the crossover operator, m number of candidates are first randomly chosen for consideration. These candidates are then sorted in descending order according to their fitness values. Subsequently, the selection starts with the first candidate in the sorted list until one fulfils the selection criterion and is select as one parent. Specifically, the selection probability of the ith candidate $s(i)$ is defined as follows:

$$s(i) = s(1-s)^{i-1},\ 0.5 < s \leq 1. \tag{5}$$

The tournament size m determines the stressfulness of comparatively lesser fit chromosomes being selected as parents. Specifically, for a relatively less fit chromosome, its chance of getting selected as parent will increase with a smaller m value, but decrease with a larger m value. Therefore, to prevent early domination of certain highly fit chromosomes or often formally known as premature convergence in the early generations of GA, s should be set to a smaller value so that less fit chromosomes still have relatively higher chances of being selected. On the other hand, to fine-tune the highly fit chromosomes with more in-depth exploitation in the late generations, s should be set to a larger value. As such, we define the tournament selection probability s as follows:

$$s = 0.5\left(1 + \frac{\mathit{icg}}{\mathit{NOG}}\right), \tag{6}$$

where *icg* denotes the index of the current generation and *NOG* denotes the predefined number of generations to terminate GA.

Because $\mathit{icg} \in [1, \mathit{NOG}]$, s for each generation in GA forms an arithmetic progression series in the interval $[0.5 + \frac{0.5}{\mathit{NOG}}, 1]$, which precisely fulfils the constraining requirement of (5).

F. Modified Crossover Operator for Varying Length

When a pair a parents have been selected, they produce offspring that partially inherit their genes through a crossover operation. Nonetheless, the crossover rate determines whether the selected pair of parents will eventually exchange their genes so that only their offspring are kept in the next generation or themselves shall be kept alternatively. Due to

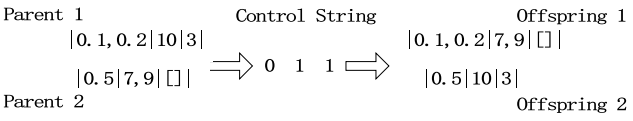

Fig. 2. An example of applying the modified uniform crossover operator.

the adoption of elitism replacement strategy, in GARSC, we always set the crossover rate to one.

To deal with the varying length of different chromosomes that comprise of different numbers of separation boundaries across all the input dimensions, we propose a modified uniform crossover operator and illustrate its usage in Fig. 2. Akin to conventional uniform crossover operators, a binary control string of length equals to NOF is first randomly generated. In each position of this string, the corresponding binary value determines a child should inherit the gene from which parent. As such, there shall be no misunderstanding in the dimensionality and length of the corresponding genes when producing the offspring.

Please recall that GARSC performs feature selection (see Section II-B), therefore, it is common for chromosomes have empty gene in the respective input dimension as represented by the square brackets "[ ]" in Fig. 2. As such, it is possible that a produced offspring consists of only empty genes in every dimension. To deal with this exception, the "empty" offspring shall be reinitialized to a random "non-empty" chromosome.

G. Modified Mutation Operators for Gene Replacement

To deal with GARSC's chromosomes, which comprise of separation boundaries across all dimensions, we propose three modified mutation operators. Specifically, one of the following three operators shall be applied on the select gene for mutation based on equal probability: (i) add one randomly selected separation boundary if it does not violate any constraint; (ii) if the gene is non-empty, remove one separation boundary from it; and (iii) vary the value of a randomly selected separation boundary if the new value does not violate any constraint.

Akin to tournament selection probability s, the mutation rate *mrate* determining the probability of mutating each gene should increase from smaller values in the early generations to larger values in the late generations. As such, we define *mrate* as follows:

$$mrate = \frac{1}{NOF} + \frac{(NOF - 1) \cdot icg}{NOF \cdot NOG}. \quad (7)$$

III. GARSINFIS: Genetic Algorithm and Rough Set Incorporated Neural Fuzzy Inference System

GARSINFIS [23] is a six-layer, feed-forward, and partially connected architecture. For the antecedent and consequent parts of the derived fuzzy rules (by GARSC), we use rectangular boxes (see Fig. 3) to represent their corresponding neurons in the condition and consequence layers, respectively.

Altogether, GARSINFIS comprises of six layers, where each layer performs the corresponding non-fuzzy or fuzzy

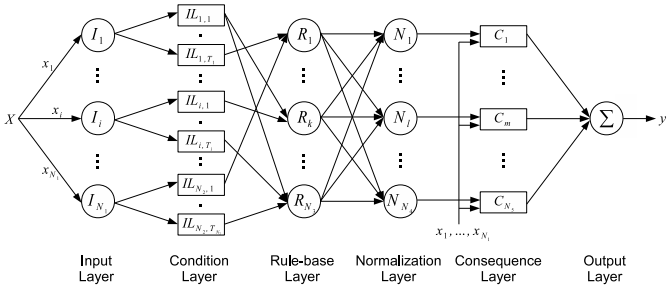

Fig. 3. The network architecture of GARSINFIS.

operation. Specifically, the input layer designates vectored input data to the corresponding linguistic variables. As GARSC performs feature selection, not all the linguistic variables are going to be used in this layer. Condition layer provides fuzzy membership functions used for each of the linguistic variables employed. Rule-base layer fires the antecedent part of fuzzy rules and passes the firing strengths to all the nodes in the next layer. Normalization layer normalizes the rule firing strengths and passes them to the respective nodes in the next layer. Consequence layer computes the consequence part of fuzzy rules using the normalized rule firing strengths and passes the results to the single neuron in the following layer. Output layer computes the final non-fuzzy output of the network. Readers may refer to [24] for all the mathematical details of the non-fuzzy and fuzzy operations used in GARSINFIS.

Because the knowledge space is reduced during the iterative clustering process, the network structure will not be fully connected and the number of nodes created in each layer will be minimized. By applying the most essential information, the network size is expected to be smaller than the other Takagi-Sugeno-Kang (TSK) [25], [26] type of neural fuzzy systems.

IV. Experimental Results

Different configurations of GARSINFIS used in this paper are summarized in Table I. Please note that the coefficient values presented in Table I are simply selected for demonstration purposes, the trade-off between accuracy and interpretability may be easily tuned by assigning the corresponding coefficient parameters (see (3)) to any combinations of real numbers.

All the datasets used in this paper (see Table II) are downloaded from UCI [27]. In each experiment, two adjacent configurations from Table I are applied for comparisons to show performance improvement. Furthermore, in each experiment run, two thirds of randomly selected data are used to train GARSINFIS and the remaining one third are used to test. The same pairs of the training and testing datasets are then applied by the benchmarking models to ensure all of them are compared on equal basis. Performance of all models is averaged from ten runs to remove randomness.

The commonly adopted GARSINFIS's control parameters in all experiments are introduced as follows: (i) in any input dimension, we only allow a maximum of two separation boundaries; (ii) we set the elitism ratio to 0.1; and (iii) we

TABLE I
DIFFERENT GARSINFIS CONFIGURATIONS EVALUATED

Id	Configuration	Details
1	GARSINFIS-crisp	employs crisp inference rules; its identified separation boundaries are different from those of the fuzzy configuration
2	GARSINFIS-a&i	focuses on both accuracy and interpretability: $\tau_{1,\ldots,5}=1$, i.e., $f(x)=(1-a)\frac{NOD}{K}+\frac{nof}{NOF}+\frac{nor}{NOD}+\frac{noa}{NOF\cdot NOD}+\frac{mse}{NOF}$
3	GARSINFIS-a	focuses on accuracy only (rules are still simplified): $\tau_{1,5}=1$, $\tau_{2,3,4}=0$, i.e., $f(x)=(1-a)\frac{NOD}{K}+\frac{mse}{NOF}$
4	GARSINFIS-1	extends the zero-order TSK fuzzy rules derived by GARSINFIS-a into first-order ones (see details in [24])

TABLE II
SUMMARY OF GARSINFIS PARAMETERS USED FOR VARIOUS DATASETS

Dataset	NOF	NOD	population size.	NOG
wine	13	178	100	60
ionosphere	32	351	200	10
material	60	208	200	20

TABLE III
RESULTS ON UCI WINE DATA SET

Model		Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-a&i	mean	**100.0**	94.31	98.09	**3.5**	5.6
	std	**0.00**	2.10	0.71	0.71	0.97
GARSINFIS-crisp	mean	96.13	93.66	95.30	3.8	5.9
	std	1.97	2.19	1.53	0.63	0.88
C4.5	mean	90.88	94.13	91.97	4.0	5.6
	std	2.29	3.49	1.79	1.25	**0.84**
Naive Bayes	mean	97.13	97.48	97.25	13	N.A.
	std	1.14	1.80	0.89		
SVM	mean	98.82	**97.82**	98.49	13	N.A.
	std	0.91	**1.37**	0.75		
MLP	mean	97.21	97.48	97.30	13	11
	std	1.06	1.63	0.78		
RBF	mean	**100.0**	**97.82**	**99.27**	13	6
	std	**0.00**	1.96	0.65		
ANFIS	mean	**100.0**	96.98	98.99	13	**3.7**
	std	**0.00**	2.05	0.69		1.16
DENFIS	mean	**100.0**	96.98	98.99	13	42.8
	std	**0.00**	1.73	**0.58**		1.99
RS-POPFNN	mean	99.83	92.12	97.26	12.5	90.5
	std	0.35	4.02	1.34	0.71	8.40
RS-HeRR	mean	**100**	94.63	98.21	4.8	73.0
	std	**0.00**	2.15	0.63	**0.32**	2.21

set the tournament size to two. The other parameter values used in each experiment are listed in Table II.

For benchmarking models, we select the following ones: C4.5 decision tree [28], Naive Bayes classifier [29], Support Vector Machine (SVM) [30], Multi-Layer Perceptron (MLP) network [31], Radial Basis Function (RBF) network [31], ANFIS [32] (in this paper, ANFIS employs the Fuzzy C-Means (FCM) clustering algorithm [33]), DENFIS [34] (employs Evolving Clustering Method (ECM) [35]), RS-POPFNN [8] and RS-HeRR [9].

A. Performance Improvement by Employing Fuzzy Rules

In this subsection, GARSINFIS-a&i and GARSINFIS-crisp (the two models are optimized separately, not directly transformed) are applied to the wine recognition dataset. As shown in Table III, GARSINFIS-a&i achieves higher accuracy and employs more compact inference rule bases than GARSINFIS-crisp. These results illustrate the necessity of representing the crisp clustering results using fuzzy membership functions to better deal with imprecise information and unforeseen circumstances. Among all models, although GARSINFIS-a&i only achieves the best accuracy on the training dataset, the rest measures are still competitive to the respective winners with small difference.

TABLE IV
RESULTS ON UCI IONOSPHERE DATA SET

Model		Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-a	mean	94.70	**90.77**	93.39	8.3	19.2
	std	1.96	2.16	1.92	1.25	2.86
GARSINFIS-a&i	mean	93.59	90.51	92.56	7.9	18.9
	std	0.92	1.87	1.05	1.37	**2.08**
C4.5	mean	97.99	90.60	95.53	7.6	**11.5**
	std	0.78	3.32	1.15	**1.17**	2.12
Naive Bayes	mean	82.88	83.68	83.14	32	N.A.
	std	1.52	3.28	1.67		
SVM	mean	90.34	85.56	88.75	32	N.A.
	std	0.88	**1.69**	**0.45**		
MLP	mean	99.36	88.80	**95.84**	32	19
	std	0.22	2.92	0.93		
RBF	mean	93.63	90.60	92.62	32	6
	std	1.09	2.67	0.79		
ANFIS	mean	**100.0**	84.24	94.75	32	24.6
	std	**0.00**	1.85	0.62		3.84
DENFIS	mean	99.91	80.17	93.33	32	93.6
	std	0.18	3.32	1.08		2.17
RS-POPFNN	mean	98.21	82.39	92.94	27.1	151.3
	std	0.53	2.62	0.99	1.20	5.50
RS-HeRR	mean	**100.0**	87.26	95.75	**6.7**	146.2
	std	**0.00**	2.56	0.85	2.16	15.0

B. Accuracy Increase without Sacrificing Interpretability

In this subsection, GARSINFIS-a and GARSINFIS-a&i are applied to the ionosphere detection dataset. As shown in Table IV, when comparing to GARSINFIS-a&i, GARSINFIS-a achieves higher accuracy but worse interpretability by employing only accuracy focused fitness function (see Table I). This finding illustrates how the trade-off between accuracy and interpretability may be effortlessly tuned by assigning different values to the respective coefficients. Among all models, although GARSINFIS-a only achieves the best accuracy in the testing dataset, the rest measures are still competitive to the respective winners with acceptable difference.

C. Further Accuracy Increase Using More Complex Rules

In this subsection, GARSINFIS-1 and GARSINFIS-a are applied to the material discrimination (sonar) dataset. As shown in Table V, when comparing to GARSINFIS-a, GARSINFIS-1 achieves higher accuracy by extending the zero-order TSK fuzzy rules into first-order ones [24]. Please note that the decrease in interpretability is not represented in the number of selected features and employed rules. It is the consequent parts of the rules become less comprehensible but fine-tune the model to achieve higher accuracy. This finding demonstrates a way to increase accuracy by sacrificing interpretability. Among all models, GARSINFIS-1 achieves satisfactory accuracy.

V. CONCLUSION

In this paper, we illustrate how we leverage the trade-off between accuracy and interpretability in GARSINFIS. In a nutshell, GARSINFIS self-organizes its network structure with

TABLE V
RESULTS ON UCI SONAR DATA SET

Model		Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-1	mean	92.72	72.59	86.01	8.6	21.1
	std	1.30	**2.94**	1.44	**0.70**	2.96
GARSINFIS-a	mean	90.70	69.38	83.53	8.6	21.1
	std	1.39	5.36	2.08	**0.70**	2.96
C4.5	mean	97.84	70.82	88.83	11.6	13.6
	std	0.59	6.25	2.23	1.58	1.17
Naive Bayes	mean	73.78	70.26	72.60	60	N.A.
	std	2.88	6.27	2.60		
SVM	mean	88.40	77.31	84.70	60	N.A.
	std	2.47	4.11	1.54		
MLP	mean	99.21	**81.21**	**93.21**	60	33
	std	3.20	4.27	2.15		
RBF	mean	96.47	81.20	91.38	60	10
	std	2.24	3.13	2.09		
ANFIS	mean	**100.0**	73.84	91.28	60	**4.4**
	std	**0.00**	3.73	**1.24**		**0.88**
DENFIS	mean	98.92	77.03	91.62	60	71.7
	std	0.91	4.43	1.43		1.64
RS-POPFNN	mean	**100.0**	70.01	90.00	24.2	137.3
	std	**0.00**	6.01	2.00	6.43	2.41
RS-HeRR	mean	**100.0**	72.85	90.95	**6.1**	120.7
	std	**0.00**	6.24	2.08	1.20	5.10

a small set of control parameters and constraints. Moreover, its autonomously generated inference rule base tries to achieve higher interpretability without sacrificing accuracy. We conduct experiments on applying difference configuration options of GARSINFIS using well-known benchmarking datasets. The performance of GARSINFIS on both accuracy and interpretability is shown to be encouraging when compared against other competitive benchmarking models.

Going forward, we plan to investigate other options to further improve the accuracy of GARSINFIS without sacrificing interpretability. Moreover, we will apply GARSINFIS on challenging real-world applications to serve as a comprehensive and reliable decision support system.

REFERENCES

[1] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems*. Prentice-Hall, 1996.

[2] S.-M. Zhou and J. Q. Gan, "Low-level interpretability and high-level interpretability: A unified view of data-driven interpretable fuzzy system modelling," *Fuzzy Sets and Systems*, vol. 159, no. 23, pp. 3091–3131, 2008.

[3] M. Setnes, R. Babuska, and H. B. Verbruggen, "Complexity reduction in fuzzy modeling," *Mathematics and Computers in Simulation*, vol. 46, no. 5-6, pp. 507–516, 1998.

[4] M. J. Gacto, R. Alcala, and F. Herrera, "Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 515–531, 2010.

[5] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 1–11, 1998.

[6] S. Guillaume and B. Charnomordic, "A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data," in *Interpretability Issues in Fuzzy Modeling*, J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, Eds. Springer-Verlag, 2003, pp. 148–175.

[7] ——, "Generating an interpretable family of fuzzy partitions from data," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 3, pp. 324–335, 2004.

[8] K. K. Ang and C. Quek, "RSPOP: Rough set-based pseudo outer-product fuzzy rule identification algorithm," *Neural Computation*, vol. 17, no. 1, pp. 205–243, 2005.

[9] F. Liu, C. Quek, and G. S. Ng, "A novel generic Hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system," *Neural Computation*, vol. 19, no. 6, pp. 1656–1680, 2007.

[10] L. Chen, C. L. P. Chen, and W. Pedrycz, "A gradient-descent-based approach for transparent linguistic interface generation in fuzzy models," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 40, no. 5, pp. 1219–1230, 2010.

[11] E. Y. Cheu, C. Quek, and S. K. Ng, "ARPOP: An appetitive reward-based pseudo-outer-product neural fuzzy inference system inspired from the operant conditioning of feeding behavior in Aplysia," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 317–329, 2012.

[12] T. Z. Tan, G. S. Ng, and C. Quek, "A novel biologically and psychologically inspired fuzzy decision support system: Hierarchical complementary learning." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 1, pp. 67–79, 2008.

[13] W. L. Tung and C. Quek, "eFSM–a novel online neural-fuzzy semantic memory model," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 136–157, 2010.

[14] M. Cococcioni, P. Ducange, B. Lazzerini, and F. Marcelloni, "A Pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems," *Soft Computing*, vol. 11, no. 11, pp. 1013–1031, 2007.

[15] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 4–31, 2007.

[16] A. Botta, B. Lazzerini, F. Marcelloni, and D. C. Stefanescu, "Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index," *Soft Computing*, vol. 13, no. 5, pp. 437–449, 2008.

[17] D. Wang, G. S. Ng, and C. Quek, "Ovarian cancer diagnosis using fuzzy neural networks empowered by evolutionary clustering technique," in *Proceedings of IEEE Congress on Evolutionary Computation*. IEEE, 2006, pp. 9314–9320.

[18] J. Holland, *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.

[19] Z. Pawlak, "Rough sets," *International Journal of Information and Computer Science*, vol. 11, no. 5, pp. 341–356, 1982.

[20] I. Eccles and M. Su, "Illustrating the curse of dimensionality numerically through different data distribution models," in *Proceedings of International Symposium on Information and Communication Technologies*, 2004, pp. 232–237.

[21] R. P. Paiva and A. Dourado, "Interpretability and learning in neuro-fuzzy systems," *Fuzzy Sets and Systems*, vol. 147, no. 1, pp. 17–38, 2004.

[22] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[23] D. Wang, G. S. Ng, and C. Quek, "A novel hybrid intelligent system: Genetic algorithm and rough set incorporated neural fuzzy inference system," in *Proceedings of IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 2546–2553.

[24] D. Wang, C. Quek, and G. S. Ng, "Ovarian cancer diagnosis using a hybrid intelligent system with simple yet convincing rules," *Applied Soft Computing*, vol. 20, pp. 25–39, 2014.

[25] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 15, no. 1, pp. 116–132, 1985.

[26] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 13–33, 1988.

[27] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[28] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[29] S. M. Ross, *Introduction to Probability Models*. Academic Press, 1985.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, p. 273297, 1995.

[31] S. Haykin, *Neural networks: A Comprehensive Foundation*. Prentice Hall, 1998.

[32] J. S. R. Jang, "ANFIS: Adaptive network-based fuzzy inference systems," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 23, pp. 650–684, 1993.

[33] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, 1981.

[34] N. K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[35] Q. Song and N. Kasabov, "ECM: A novel on-line, evolving clustering method and its applications," in *Proceedings of Conference on Artificial Neural Networks and Expert Systems*, 2001, pp. 87–92.