An Interpretable Neural Fuzzy Inference System for Predictions of Underpricing in Initial Public Offerings

Di Wang[a], Xiaolin Qian[b], Chai Quek[c], Ah-Hwee Tan[a,c], Chunyan Miao[a,c], Xiaofeng Zhang[d], Geok See Ng[e], You Zhou[f,*]

[a] *Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore*
[b] *Department of Risk Assessment, Standard Chartered Bank, Singapore*
[c] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[d] *Department of Computer Science, Harbin Institute of Technology, Shenzhen, China*
[e] *School of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore*
[f] *College of Computer Science and Technology, Jilin University, Changchun, China*

Abstract

Due to their aptitude in both accurate data processing and human comprehensible reasoning, neural fuzzy inference systems have been widely adopted in various application domains as decision support systems. Especially in real-world scenarios such as decision making in financial transactions, the human experts may be more interested in knowing the comprehensive reasons of certain advices provided by a decision support system in addition to how confident the system is on such advices. In this paper, we apply an integrated autonomous computational model termed genetic algorithm and rough set incorporated neural fuzzy inference system (GARSINFIS) to predict underpricing in initial public offerings (IPOs). The difference between a stock's potentially high value and its actual IPO price is referred as money-left-on-the-table, which has been extensively studied in the literature of corporate finance on its theoretical foundations, but surprisingly under-investigated in the field of computational decision support systems. Specifically, we use GARSINFIS to derive interpretable rules in determining whether there is money-left-on-the-table in IPOs to assist the investors in their decision making. For performance evaluations, we first demonstrate

*Corresponding author
Email address: zyou@jlu.edu.cn (You Zhou)

how to balance between accuracy and interpretability in GARSINFIS by simply altering the values of several coefficient parameters using well-known datasets. We then use GARSINFIS to investigate the IPO underpricing problem. The encouraging experimental results show that we may yield higher initial returns of IPOs by following the advices provided by GARSINFIS than any other benchmarking model. Therefore, our autonomous computational model is shown to be capable of offering the investors highly interpretable and reliable decision supports to grab the money-left-on-the-table in IPOs.

Keywords: neural fuzzy inference system, interpretable rules, initial public offering, financial decision support system, IPO underpricing

---

1. 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 integrates 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]. Because accuracy and interpretability are two contradicting objectives, an ideal system (see Figure 1) is usually not available. In most cases, a satisfactory balance between the aforementioned two contradictory objectives is made based on the complexity and purpose of the underlying application.

In this paper, we illustrate how we leverage the trade-off between accuracy and interpretability in an NFIS termed 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

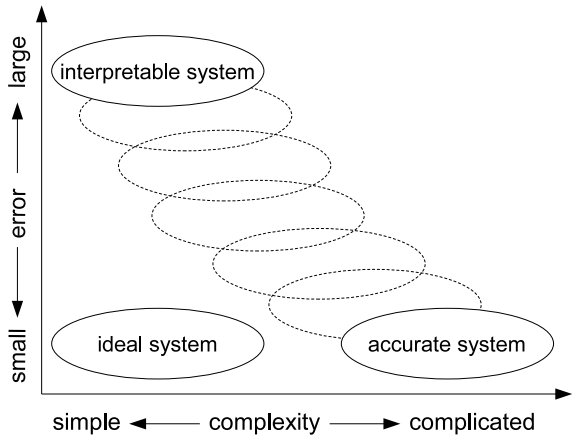


Figure 1: Illustration of the trade-off between accuracy and interpretability.

is autonomously derived by an iterative clustering algorithm termed 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 first conduct experiments on well-known datasets of different complexity to demonstrate the different configuration options of GARSINFIS. We then investigate the underpricing problem in initial public offerings (IPOs) by applying GARSINFIS to predict whether there is money-left-on-the-table.

IPO refers to a type of public offering in which shares of a company are sold to the general public through a securities exchange for the first time. During IPO, many companies choose to purposely lower their stocks' price to create more incentives for the potential investors. The difference between the stock's potentially high value and its purposely lowered IPO price is referred as money-left-on-the-table. However, due to various reasons, there is not always money-left-on-the-table in IPOs, which means the investors may lose money, or on the other hand, miss the opportunity to grab the money-left-on-the-table. Although this problem has been extensively studied in the literature of corporate finance on its theoretical foundations, surprisingly, there are only few studies

carried out to investigate this problem using computational models for practical decision supports. Moreover, to the best of our knowledge, there is only one computational model proposed in the literature, which derives interpretable rules to assist the decision making of the investors. In this paper, we collect a dataset comprising 28 years of IPOs in U.S. and follow the experts' prior study [3] to generate fifteen robust determinants for investigations.

Our main contributions in this paper are summarized as follows:

1. We introduce the dynamics of an autonomous clustering algorithm and a neural fuzzy inference system with detailed mathematical formulations.
2. We analyse and highlight both the low-level and high-level interpretability properties of the overall system.
3. We demonstrate how the balance between accuracy and interpretability may be straightforwardly altered by assigning different values to a few coefficient parameters using well-known datasets.
4. We show that our hybrid intelligent system is capable of offering the investors highly interpretable and reliable decision support to grab the money-left-on-the-table in IPOs.

The rest of this paper is organized as follows. Section 2 reviews relevant literature. Sections 3 and 4 present the preliminaries of rough set theory and genetic algorithm, respectively. Section 5 introduces the GARSC clustering algorithm. Section 6 presents the system architecture of GARSINFIS, which employs and fine-tunes the fuzzy rules derived by GARSC. Sections 7 and 8 report the experimental results of applying GARSINFIS in different configurations on well-known datasets and own-collected IPO dataset, respectively. Section 9 concludes this paper and proposes future work.

2. Related Work

In this section, we review the literature in two relevant research fields, namely interpretability improvement in NFISs and financial decision support systems.

2.1. Improvement of Interpretability in NFISs

There are two major approaches proposed in the literature to improve the interpretability of an NFIS, namely reducing the complexity after the construction of the model and defining constraints before the construction process. In terms of model complexity, the latter approach is generally more complex due to the varies constraints considered during the learning process, however, as a form of compensation, the resulting models are usually more interpretable. For the first approach, many methods have been proposed, such as rule aggregation [4], rule removal [5], rule transformation [6], feature selection [7], and knowledge reduction (on both rules and features) [8–10]. The second approach mainly focuses on controlling the quality and quantity of the derived membership functions [11, 12] and also focuses on defining constraints on both membership functions and rules [13, 14]. Only a few prior studies [15–17] provide options or device parameters to leverage the trade-off between accuracy and interpretability. In this paper, we show that our model can also be easily configured to leverage such trade-off by simply altering several coefficient parameter values.

2.2. Existing Financial Decision Support Systems

Due to their desirable high-level accuracy, NFISs or neural networks in general have been adopted in various financial application domains as decision support systems. To predict the likelihood of bank failures, Wang et al. extracted nine fiancial covariates [18]. To study bank risk contagion, Cerchiello et al. further expanded the data sources, i.e., from both financial markets and financial tweets [19]. Along the same line of research, Ronnqvist and Sarlin relied on the text analysis of public news on bank distress and government interventions [20]. In another well-known financial application domain, i.e., stock price prediction, a recent study [21] reported the influence of varying input window length on the prediction of stock price movement directions. Moreover, decision support systems for stock exchanges generally favour accuracy (normally in terms of RMSE, e.g., [22]) much more than interpretability. In this paper, we strive for

a better leverage of the trade-off between both accuracy and interpretability in an NFIS model for IPO underpricing prediction.

Although computational models have been widely adopted to assist in various financial applications, surprisingly, there are only a limited number of prior studies on the prediction of IPO underpricing. The first well-known IPO pricing study using neural networks was conducted back in 1995 [23], wherein Jain and Nag closely approximated the pricing of IPOs according to IPO first day closing price. Similarly, Robertson et al. [24] and Reber et al. [25] also constructed neural network models to predict the post-IPO market price. Alternatively, Yao and Zhou employed rough set theory and support vector machine to identify the most influential factors in Chinese IPOs [26]. Only recently, a comprehensive study on using machine learning models to predict the initial returns of IPOs was published [27]. However, in [27], none of the investigated models may easily generate human interpretable rules, which means the decision support systems would be all "black-boxes". To the best of our knowledge, the only interpretable rule-based model used to predict IPO underpricing was proposed in [28]. Although Quintana et al. reported in [28] that their rule-based decision model optimized by genetic algorithm (GA) outperforms the regression approach, they only include seven financial covariates in their study and the overall data sample size is 840. In this paper, we include fifteen input variables following the experts' suggestions [3] and our data sample size is 5,203. We select Quintana's GA model [28] as one of the benchmarking models when we investigate the IPO underpricing problem.

3. Rough Set Theory for Knowledge Reduction

Rough set theory was first proposed by Pawlak [29] to investigate the intrinsic relations or knowledge embedded in a given dataset, which can be in turn used to reduce the dimensionality of the underlying dataset [30]. Rough sets are often compared to fuzzy sets [31], which use membership functions to define the degree of the belongingness. In comparison, rough set uses approximations

Table 1: A Simple Illustration of a Decision Table

U	A		
	C		D
	weight	height	body size
1	light	short	small
2	heavy	tall	big

to model the relationships between subsets of data. It has been suggested to integrate both theories in one system to exploit their complements [32].

3.1. Construction of Decision Tables

To perform knowledge reduction, rough set theory employs decision logic language to model the knowledge representation system (KRS). Such a system S comprises a non-empty and finite set U, which denotes the universe of discourse, and a non-empty and finite set A, which denotes the primitive attributes, i.e., $S = (U, A)$.

A decision table can then be defined based on a KRS. Assume in $S = (U, A)$, we know the condition attributes C and decision attributes D, i.e., $C, D \subset A$, then a decision table $T = \{U, C, D\}$ can be constructed as S with distinguished (C, D) pairs (see Table 1). Moreover, in the context of clustering, each row in T may represent a cluster of data samples [33].

3.2. Relationship Approximations in Rough Set Theory

In rough set theory, the indiscernible relation $IND(G)$ over knowledge G is defined as follows:

$$IND(G) = \{(p, q) \in U^2 \mid \forall r \in G, r(p) = r(q)\}. \tag{1}$$

The set of all correspondence relations $IND(G)$ is denoted as $U/IND(G)$.

Furthermore, relationships in rough set theory are approximated by the *lower and upper approximations* [29]. When $IND(G)$ is provided, these approximations are defined as follows:

$$\underline{G}Q = \bigcup \{P : P \in U/IND(G),\ P \subseteq Q\}, \tag{2a}$$

$$\overline{G}Q = \bigcup \{P : P \in U/IND(G),\ P \cap Q \neq \phi,\ P \subseteq Q\}. \tag{2b}$$

Specifically, the lower approximation $\underline{G}Q$ denotes the set of elements that can be definitely distinguished by G and Q, and upper approximation $\overline{G}Q$ denotes the set of elements that can be probably distinguished by G and Q.

3.3. Attribute Removal and Feature Selection in Rough Set Theory

In rough set theory, knowledge reduction is performed based on two fundamental definitions called *reduct* and *core* [29]. Reduct denotes a subset of knowledge that necessarily defines all the essential relations and core denotes the subset of primary knowledge that only comprises the commonly shared knowledge among all reducts.

In a decision table $T = \{U, C, D\}$, an attribute k is dispensable if and only if $IND(C) = IND(C - \{k\})$. Otherwise, k is indispensable. Moreover, C is independent if $\forall k \in C$ is indispensable. Attribute removal is carried out during the procedure of identifying independent C with minimal cardinality. If an attribute k is dispensable in all relations, it may be excluded from C. As such, feature selection is performed during the process of excluding all dispensable attributes.

Based on the afore-defined indispensable relations, $Y \subseteq X$ is a reduct of X, if Y is independent and $IND(X) = IND(Y)$. The core of X is defined as the overlapping portions of all reducts, i.e. $CORE(X) = \bigcap REDUCT(X)$.

3.4. Knowledge Reduction in Rough Set Theory

Knowledge reduction is carried out during the procedure of identifying a reduct of the decision table. Specifically, when given a dataset, if the data are continuous and the separation boundaries in every dimension are known, the continuous data can be discretized into categorical values. As such, by removing all the dispensable attributes based on rough set approximations and further merging the duplicates, we obtain a simplified set of decision rules.

4. Genetic Algorithm to Optimize Boundary Separations

Genetic algorithm (GA) [34] is designed based on the dynamics of natural selection and mechanics of natural genetics [35]. In the beginning of a typical GA procedure, a population of artificial creatures referred as chromosomes are randomly initialized. Then in each generation, certain highly fit chromosomes are selected in pairs as parents to produce offspring. This reproduction process is regulated by the crossover operations. Moreover, certain chromosomes are then selected for mutation, wherein their genes are varied. Over the iterative productions of new generations of chromosomes, better solutions are obtained. The production process ends when any termination criterion is met. Although GA is regulated in a random manner, it efficiently "exploits historical information to speculate on new search points with expected improvements" [35].

Because rough set theory only deals with categorical values, if the underlying dataset is continuous, discretization is required first. As such, we employ GA to optimize the selections of separation boundaries in each input dimension. The relevant GA strategies adopted by our clustering algorithm are introduced in the following section.

5. GARSC: Genetic Algorithm based Rough Set Clustering

GARSC [36] incorporates the advantages of both genetic algorithm [34] and rough set theory [29]. Specifically, we employ genetic algorithm to look for desirable feature segmentations and use rough set theory to perform knowledge reduction [37]. Based on rough set knowledge reduction, any categorical inference rule set can be prominently reduced without discarding any indispensable knowledge. This great property of rough set theory can be really beneficial in improving the legibility of a set of inference rules [38], 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 Figure 2). Specifically, suppose in

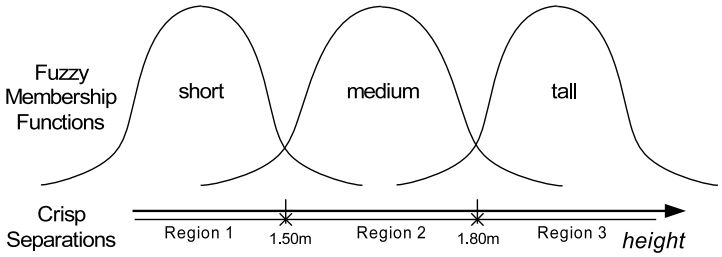


Figure 2: Illustration of transferring crisp membership functions to fuzzy ones.

dimension x, we select $(n-1)$ number of separation boundaries, then x can be 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. Subsequently, the generated fuzzy rules are employed by GARSINFIS (see Section 6) for performance evaluation as a solution candidate.

The procedure 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 (MFs) to tolerate imprecise information for better performance in unforeseen circumstances. This particular step of knowledge transfer naturally prevents the resulting fuzzy MFs from separating or overlapping too much with their neighbours, which makes the fuzzy rules more interpretable. Moreover, as the MFs are generated in each individual dimension without normalization and transformation, the representations of the associated fuzzy semantic labels are deemed highly interpretable. The necessity of transformation from crisp membership functions to fuzzy ones is also empirically shown in Section 7.1.

5.1. 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 maximal number of separation boundaries allowed in each dimension. It is easy to infer that this restriction subsequently defines the maximal number of fuzzy MFs that might be

devised in each dimension. Nonetheless, the actual number of fuzzy MFs derived is also affected by the knowledge reduction procedure. In any dimension, the minimal number of separation boundaries really in use is zero, which denotes that the corresponding dimension is not included in the reducted inference rule base. Furthermore, this number of maximal number of separation boundaries allowed in each dimension should not be large so as to avoid the employment of a large number of fuzzy MFs, which degrades the interpretability of the overall model.

The second constraint is on the minimal distance to be guaranteed between any neighbouring separation boundaries in the same dimension. This restriction 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 minimal distance requirement and present its definition as follows:

$$mindis_j = \frac{ub_j - lb_j}{\max(nop_j, M)}, \tag{3}$$

where j denotes the jth dimension, ub_j and lb_j denote the maximal value and the minimal value seen in the jth dimension, respectively, nop_j denotes the count of all different values seen in the jth dimension that for each corresponding conditional attribute value, its associated decision attribute has more than one values, and M denotes the predefined minimal number of separation bins in each dimension, which is assigned to 10 unless specified otherwise.

5.2. 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 removal to find the reduct of the original decision table with the minimal 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 [39]. The confidence of the ath rule $conf(a)$ is computed as follows:

$$conf(a) = \min\left(\frac{card(U_j(a_j) \cap d_a)}{card(U_j(a_j))}\right), \forall j \in C, \tag{4}$$

where $card$ computes cardinality, U_j denotes the union function of decision attributes of each individual decision rule that has the same categorical value on the jth dimension, a_j denotes the categorical value of the ath rule on the jth dimension, and d_a denotes the decision attribute value of the ath rule.

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

5.3. Commonly Adopted Strategies in Genetic Algorithms

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

GARSC uses real numbers to construct chromosomes. Specifically, each gene of a chromosome represents a separation boundary in the corresponding dimension. Please note that although GARSC confines the maximal number of separation boundaries allowed in each dimension, the actual number of partitions in use varies, i.e., chromosomes (consisting 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 [35]. Specifically, when producing chromosomes in the next generation $G(t+1)$, a certain number of chromosomes in the current generation $G(t)$ shall be directly kept in $G(t+1)$. The number is determined by the elitism ratio $\mu \in [0, 1)$ and the population size. 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 predetermined number of generations. This generation number should be set carefully to allow GA to converge.

5.4. Fitness Evaluation of Chromosomes

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 component 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 $f(x)$ as follows:

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

where x denotes the chromosome under evaluation, $\tau_1, \ldots, \tau_5$ denote the predetermined coefficient values, a denotes the accuracy of solution x on the underlying dataset, *NOD* denotes the number of data samples, *NOF* denotes the total number of dimensions exist in the underlying dataset, *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 the mean squared error that

$$mse = \frac{1}{NOD}\sum_{i=1}^{NOD}(y_i - \hat{y}_i)^2, \quad (6)$$

where y_i denotes the value of prediction and $\hat{y}_i$ denotes the value of ground truth. Please note that capital letters are used to denote constant values and small letters are used to denote variables. Terms 1 and 5 in (5) 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.

5.5. 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 better fitness values than those not being selected. Among all parent selection strategies, we adopt tournament selection [40], in which the competition among candidates can be easily regulated by tournament size m and selection probability s.

Prior to the selection of two parents to produce offspring by applying the crossover operator, m number of candidates are first randomly chosen for consideration. These candidates are then sorted in the ascending order (because the fitness function is to be minimized) based on their fitness values. Subsequently, the selection starts from the beginning of the sorted list until one fulfils the selection criterion. Specifically, the selection probability of the ith candidate $s(i)$ is defined as follows:

$$s(i) = s(1-s)^{i-1},\ 0.5 < s \leq 1. \tag{7}$$

The tournament size m determines the stressfulness of less fit chromosomes being selected as parents. Specifically, for a relatively less fit chromosome, its chance of getting selected as a parent will increase with a smaller m value, but decrease with a larger m value. Moreover, 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:

$$s = 0.5\left(1 + \frac{icg}{NOG}\right), \tag{8}$$

where icg denotes the index of the current generation and NOG denotes the predefined total number of generations until GA terminates. Because $icg \in [1, NOG]$, s for each generation in GA forms an arithmetic progression series in the $[0.5 + \frac{0.5}{NOG}, 1]$ interval, which precisely fulfils the constraining requirement defined in (7).

5.6. 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 the adoption of elitism replacement strategy, in GARSC, we always set the crossover rate to 100%.

To deal with the varying length of different chromosomes that comprise different numbers of separation boundaries across all the input dimensions, we propose a modified uniform crossover operator and illustrate its usage in Figure 3. Akin to normal uniform crossover operators, a binary control string of length equals to NOF is 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 5.2), therefore, it is common for a chromosome has empty gene in the respective input dimension as represented by the square brackets "[ ]" in Figure 3. 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.

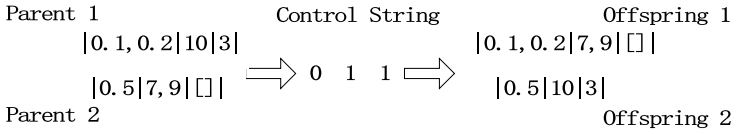


Figure 3: An example of applying the modified uniform crossover operator.

5.7. Modified Mutation Operators for Gene Replacement

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

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

$$mr = \frac{1}{NOF} + \frac{(NOF-1)\cdot icg}{NOF \cdot NOG}. \tag{9}$$

5.8. Computational Complexity Analysis

Because GARSC iteratively optimizes the inference rule base, in this subsection, we further analyze its computational complexity. First of all, when dealing with each individual solution candidate, the computationally heavy procedures of GARSC are identified as decision table construction, rule transformation (from crisp rules to fuzzy ones), and knowledge reduction. Furthermore, the complexity of knowledge reduction is determined by three major procedures, namely attribute reduction, conflict rule removal, and redundant rule removal. The complexity of each major procedure is reported in Table 2. As shown, the

Table 2: Computational Complexity of Individual Solution Candidate

Procedure	Complexity
Decision table construction	$O(NOF \cdot NOD)$
Rule transformation	$O(NOF \cdot NOD)$
Knowledge reduction (attribute)	$O(nor^2 \cdot nof)$
Knowledge reduction (conflict rule)	$O(nor^2)$
Knowledge reduction (redundant rule)	$O(nor \cdot NOF \cdot NOD^2)$
Overall	$O(nor \cdot NOF \cdot NOD^2)$

overall complexity is mainly determined by redundant rule removal, wherein the performance of the fuzzy inference rule set is iteratively evaluated.

At the end of each generation of GA, GARSC produces a new population of solution candidates for performance evaluations in the subsequent generation. The complexity of this generation procedure is determined by the population size P, tournament size m, and the number of features in use *nof*, i.e., $O(P \cdot m \cdot nof)$. Because in this work, P is always smaller than NOD^2 (see Table 4 and Section 8.1), m is set to a small value, and $nof \leq NOF$, the complexity of this procedure is always smaller than the overall complexity of dealing with an individual solution candidate (see Table 2). Therefore, the complexity of GARSC in one GA generation is $O(P \cdot nor \cdot NOF \cdot NOD^2)$. As such, the overall complexity of GARSC is determined as $O(NOG \cdot P \cdot nor \cdot NOF \cdot NOD^2)$.

Furthermore, because GARSINFIS simply uses the fuzzy inference rules derived by GARSC to organize its network structure and set the weight vectors accordingly (if zero-order TSK rules are employed, see more technical details in the subsequent section), the computational complexity of whole model is the same as that of GARSC.

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

GARSINFIS (see Figure 4) is a six-layer, feed-forward, and partially connected architecture [36]. In each layer, neurons are not connected to each other

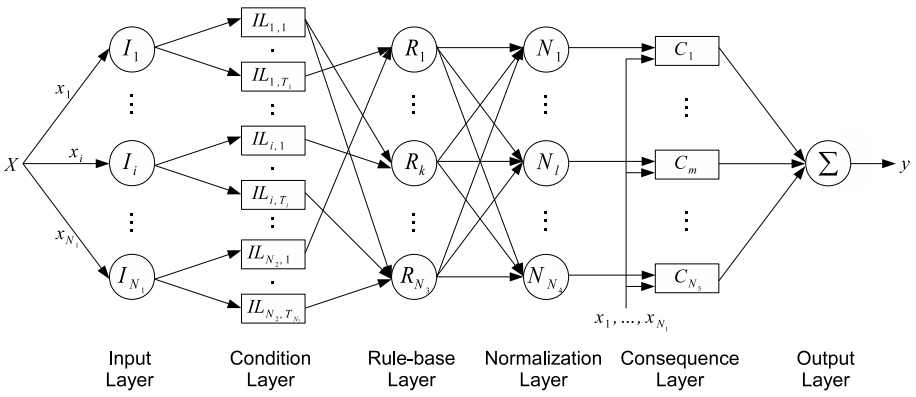


Figure 4: The network architecture of GARSINFIS.

but only connected to neurons in the adjacent layers. For the antecedent and consequent parts of the fuzzy rules (derived by GARSC), we use rectangular boxes to represent their corresponding neurons in the condition and consequence layers, respectively. GARSINFIS employs TSK type of fuzzy rules R [41, 42] as follows:

$$\begin{aligned} R_i: \quad & \text{IF } x_1 \text{ is } A_{(1,i)} \wedge \cdots \wedge x_N \text{ is } A_{(N,i)} \\ & \text{THEN } y_i = \alpha_{(0,i)} + \alpha_{(1,i)} x_1 + \cdots + \alpha_{(N,i)} x_N, \end{aligned} \tag{10}$$

where x_n denotes the nth input attribute, $A_{(n,i)}$ denotes the fuzzy linguistic label of the ith rule on the nth input attribute, N denotes the total number of attributes, y_i denotes the output of the ith rule, and $\alpha_{(n,i)}$ denotes the coefficient associated to x_n in the ith rule.

TSK type of fuzzy rules have been widely adopted in the literature [43–47] due to their high precision in function approximation. To improve the legibility of TSK rules, the authors of [48] adopt sparse regularization such that more consequent coefficients can be approximated to zero. Moreover, if the consequent part of the rules are simplified to $y_i = \alpha_{(0,i)}$, they are named zero-order TSK rules (e.g., [49]). In this paper, GARSINFIS simply employs the zero-order

TSK rules derived by GARSC, if higher level of legibility is preferred.

6.1. Input layer

Neurons in the input layer, termed as linguistic neurons, receive and transform the input vector into fuzzy singletons in the respective dimension. Because feature selection has been performed during clustering, not all input features in the given dataset are actually in use. The input function f_i^I and output function o_i^I of linguistic neurons are defined as follows:

$$f_i^I = x_i \ ; \ o_i^I = f_i^I, \tag{11}$$

where x_i denotes the ith element of input vector X and $i \in \{1, ..., N_1\}$, $N_1 = nof \leq NOF$.

6.2. Condition Layer

Neurons in the condition layer, termed as input label neurons, compute the activations of the corresponding fuzzy membership functions (MFs). The number of input label neurons in the ith dimension T_i equals to the number of fuzzy MFs formulated in the corresponding dimension. Moreover, the maximal value of T_i is restricted by the maximal number of separation boundaries allowed in GARSC. The input function f_{ij}^{II} and output function o_{ij}^{II} of input label neurons are defined as follows:

$$f_{ij}^{II} = -\frac{(o_i^I - c_{ij}^{II})^2}{2\sigma_{ij}^{II^2}} \ ; \ o_{ij}^{II} = \exp(f_{ij}^{II}), \tag{12}$$

where c_{ij}^{II} denotes the mean of the jth cluster in the ith dimension, σ_{ij}^{II} denotes the corresponding standard deviation, and $j \in \{1, ..., T_{N_2}\}, N_2 = N_1$.

6.3. Rule-base Layer

Neurons in the rule-base layer, termed as rule-base neurons, perform fuzzy reasoning based on the activation values received from the condition layer. The number of inputs received by each rule-base neuron equals to the number of

arguments exist in the corresponding fuzzy rule. Since attribute reduction has been performed on the antecedent parts of the rules during clustering, rule-base neurons may not be connected to condition neurons in every dimension. The input function f_k^{III} and output function o_k^{III} of rule-base neurons are defined as follows:

$$f_k^{III} = \min(o_{ij}^{II}) \ ; \ o_k^{III} = f_k^{III}, \tag{13}$$

where $k \in \{1, ..., N_3\}$, $N_3 = nor$.

6.4. Normalization Layer

Neurons in the normalization layer, termed as normalization neurons, normalize the activation values of all fuzzy rules. The normalization of rule firing strength should not be omitted and its necessity has been analysed in [50]. The input function f_l^{IV} and output function o_l^{IV} of normalization neurons are defined as follows:

$$f_l^{IV} = \frac{o_l^{III}}{\sum o_k^{III}} \ ; \ o_l^{IV} = f_l^{IV}, \tag{14}$$

where $l \in \{1, ..., N_4\}$, $N_4 = N_3$.

6.5. Consequence Layer

Neurons in the consequence layer, termed as consequence neurons, compute the prediction of each rule according to the normalized rule firing strength received from the normalization layer. The input function f_m^V and output function o_m^V of consequence neurons are defined as follows:

$$f_m^V = c_0 + c_1 x_1 + \cdots + c_{N_1} x_{N_1} \ ; \ o_m^V = o_m^{IV} f_m^V, \tag{15}$$

where c_0 denotes a constant, c_i denotes the coefficient associated to the ith attribute, and $m \in \{1, ..., N_5\}$, $N_5 = N_4$.

The rules produced by GARSC is zero-order, i.e., $c_i = 0, \forall i \in \{1, ..., N_1\}$. In complex applications or by user requirements, zero-order rules may be extended

into first-order ones to further increase accuracy with the price of decreasing legibility. Specifically, GARSINFIS employs the recursive least squares (RLS) algorithm to estimate the optimal coefficient matrix W^*. Assume there are P training data samples and let a_p denotes the pth row of the weighted input matrix A, then W^* can be recursively estimated as follows:

$$\begin{aligned} S_0 &= \gamma I, \\ W_0 &= [c_{10} \cdots c_{M0} \ \overbrace{0 \cdots 0}^{MN}]^T, \\ S_p &= S_{p-1} - \frac{S_{p-1} a_p^T a_p S_{p-1}}{1 + a_p S_{p-1} a_p^T}, \quad p = 1, \cdots, P, \\ W_p &= W_{p-1} + S_p a_p^T \underbrace{(D_p - a_p W_{p-1})}_{\text{prediction error}}, \\ W^* &= W_P, \end{aligned} \tag{16}$$

where γ denotes a large positive value, I denotes the identity matrix, c_{m0} denotes the consequent part of the mth zero-order TSK rule, $M = nor$, $N = nof$, S_p denotes the error covariance matrix of the pth input vector, and D denotes the matrix of ground truth.

6.6. Output Layer

The only neuron in the output layer, termed as the output neuron, accumulates the inputs received from the consequence layer and output the prediction value. The input function f^{VI} and output function o^{VI} of the output neuron are defined as follows:

$$f^{VI} = \sum o_m^V \ ; \ o^{VI} = f^{VI}. \tag{17}$$

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

6.7. Interpretability Properties of GARSINFIS

Based on the detailed introductions on GARSC and GARSINFIS, we list the desirable interpretability properties of the overall NFIS model in this section. These properties are summarized from prior studies [2, 51–53]. Moreover, the first six properties represent low-level interpretability, i.e., optimization of MFs on fuzzy set level, and the last four properties represent high-level interpretability, i.e., derivation of compact and consistent fuzzy rule base [2].

i) **Completeness**: The entire universe of discourse U of any input dimension should be covered by the derived MFs, i.e., every datum should belong to at least one fuzzy membership function (MF) $\mu_i(x)$:

$$\forall x \in U, \exists \mu_i(x) \in F : \mu_i(x) > 0, \tag{18}$$

where F denotes the set of all MFs. Because GARSINFIS employs Gaussian type of MF, U is covered by each MF.

ii) **Convexity**: The membership value of a datum belonging to any interval should not be lower than the lower membership value at the boundaries of the interval:

$$\forall a, b, x \in U : a \leq x \leq b \rightarrow \mu_i(x) \geq \min(\mu_i(a), \mu_i(b)). \tag{19}$$

Gaussian type of MF is convex because it monotonically decreases along either direction starting from the centroid.

iii) **Distinguishability**: Each MF should represent a clear semantic meaning, which is distinguishable from the others in the same input dimension. In

other words, each MF should not overlap too much with its neighbors:

$$\forall \mu_i(x), \mu_{i+1}(x) \in F : \max_{x \in U}(\min(\mu_i(x), \mu_{i+1}(x))) \leq t, \tag{20}$$

where t denotes the desired overlap threshold. This property is intuitively realized by the discretization of each input dimension based on the selected separation boundaries (see Figure 2 and Section 5.1).

iv) **Normality**: An MF is normal if there is at least one datum has full membership value:

$$\exists x \in U : \mu_i(x) = 1. \tag{21}$$

The centroid of every Gaussian type of MF has full membership value of one.

v) **Small number of MFs**: The number of MFs in each dimension should be kept within the maximal number (7 ± 2) of conceptual entities that human can efficiently handle [54]. This property is realized by setting the maximal number of partitions allowed in each dimension (see Section 5.1).

vi) **Unimodality**: An MF is unimodal if there is only one datum has full membership value:

$$\exists p, q \in U : \mu_i(p) = \mu_i(q) = \max_{x \in U} \mu_i(x) \Rightarrow p = q. \tag{22}$$

Gaussian type of MF is unimodal because only the centroid has full membership value of one.

vii) **Consistency**: The inference rule base is consistent if there are no contradictory rules. In GARSC, only one rule from every inconsistent rule set is retained (see Section 5.2).

viii) **Readability of single rule**: The number of arguments in the antecedent part of each rule should not exceed 7 ± 2 [54]. Moreover, fewer words may be recalled if they have longer spoken duration [55] or they have similar speech sounds [56]. Therefore, the antecedent part of each rule should employ a small number of arguments with short and distinctive linguistic labels. This property is realized by performing attribute reduction and the discretization of each input dimension based on the maximal number of partitions allowed

(see Figure 2, Sections 5.1 and 5.2) and its score is incorporated in the fitness function (term-4 of (5)).

ix) **Small number of features**: The employment of only a subset of the original features decreases the dimensionality of the problem and increases the readability of the rule base. This property is realized by performing feature selection (see Section 5.2) and its score is incorporated in the fitness function (term-2 of (5)).

x) **Small number of rules**: The employment of a smaller number of rules increases the legibility of the rule base if the model's accuracy is retained. This property is realized by performing rule removal (see Section 5.2) and its score is incorporated in the fitness function (term-3 of (5)).

Although it is widely accepted that Mamdani type of fuzzy rules [57] are more comprehensible than TSK ones because the consequent parts of Mamdani rules are fuzzy sets and those of TSK ones are linear functions, it is stated in [2] that interpretability of TSK fuzzy model should be evaluated based on how well the local linear models fit the non-linear global model in the respective local regions. In GARSINFIS, zero-order TSK rules are derived first to maximize the level of interpretability. Based on the complexity of the application or by user requirements, zero-order rules may be extended into first-order ones (see Section 6.5) to increase accuracy.

7. Experimental Results on Well-Known Datasets

Different configurations of GARSINFIS used in this paper are summarized in Table 3. Please note that the coefficient values presented in Table 3 are simply selected for demonstration purposes, the balanced between accuracy and interpretability may be easily adjusted by assigning the corresponding coefficient parameters (see (5)) to any combinations of real numbers.

All the datasets used in this section (see Table 4) are downloaded from UCI [58]. In each experiment scenario (see Sections 7.1 to 7.3), two adjacent configurations from Table 3 are applied for comparisons to show performance

Table 3: 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 Section 6.5)

Table 4: Summary of UCI Datasets in Use

Dataset	NOF	NOD	Population size	NOG
iris	4	150	100	30
wine	13	178	100	60
thyroid	5	215	200	80
ionosphere	32	351	200	10
glass	9	214	200	20
material	60	208	200	20

improvement. Furthermore, in each experimental run, two thirds of randomly selected data are used to train GARSINFIS (half of the training dataset are used for model construction and the remaining half are used for validations for rule removal, see Section 5.2) and the remaining are used for testing. The same pairs of the training and testing datasets are then used by the benchmarking models to ensure all of them are compared on equal basis. Performance of all models is averaged over 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 maximal of two separation boundaries (i.e., each dimension comprises at most three fuzzy membership functions), ii) we set the elitism ratio to 0.1, and iii) we set the tournament size to two. The other two parameter values (i.e., population size and number of iterations) used in each experiment are listed in Table 4.

For benchmarking models, we select the following ones: C4.5 [59], Naive Bayes [60], SVM [61], MLP [62], RBF [62], ANFIS [63] (in this paper, ANFIS employs the fuzzy c-means (FCM) clustering algorithm [64]), DENFIS [65] (employs evolving clustering method (ECM) [66]), RS-POPFNN [9] and RS-HeRR [10]. Among these benchmarking models, only C4.5, RS-POPFNN and RS-HeRR performs feature selection and all the other models use all input features. Moreover, in terms of the number of derived rules, we report the number of tree leaves in C4.5, the number of hidden neurons employed by MLP, and the number of radial basis neurons used by RBF. Because MLP and RBF do not produce interpretable rules, the corresponding number of neurons (determined by trial-and-error) listed in the number of rules column (see Tables 5 to 10 and 14) are presented in parentheses and are not used for comparison.

7.1. 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 iris classification and wine recognition datasets. As shown in Tables 5 and 6, 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 in wine recognition, the rest measures are still competitive to the respective winners with small difference.

7.2. Accuracy Increase without Sacrificing Interpretability

In this subsection, GARSINFIS-a and GARSINFIS-a&i are applied to the thyroid diagnosis and ionosphere detection datasets. As shown in Tables 7 and 8, when comparing to GARSINFIS-a&i, GARSINFIS-a achieves higher accuracy but worse interpretability by employing only accuracy focused fitness

Table 5: Results on UCI Iris Dataset

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-a&i	98.90	96.60	98.13	2.3	**3.4**
GARSINFIS-crisp	96.10	95.20	95.80	2.2	3.7
C4.5	94.90	94.60	94.80	**2.1**	4.1
Naive Bayes	95.50	96.20	95.73	4	N.A.
SVM	96.10	**97.20**	96.47	4	N.A.
MLP	97.00	95.60	96.53	4	(6)
RBF	98.70	94.80	97.40	4	(6)
ANFIS	99.70	96.40	**98.60**	4	4.6
DENFIS	**100.0**	95.20	98.40	4	13.4
RS-POPFNN	97.90	94.80	96.87	3.9	13.4
RS-HeRR	97.90	95.60	97.13	2.3	8.5

Table 6: Results on UCI Wine Dataset

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-a&i	**100.0**	94.31	98.09	**3.5**	5.6
GARSINFIS-crisp	96.13	93.66	95.30	3.8	5.9
C4.5	90.88	94.13	91.97	4.0	5.6
Naive Bayes	97.13	97.48	97.25	13	N.A.
SVM	98.82	**97.82**	98.49	13	N.A.
MLP	97.21	97.48	97.30	13	(11)
RBF	**100.0**	**97.82**	**99.27**	13	(6)
ANFIS	**100.0**	96.98	98.99	13	3.7
DENFIS	**100.0**	96.98	98.99	13	42.8
RS-POPFNN	99.83	92.12	97.26	12.5	90.5
RS-HeRR	**100**	94.63	98.21	4.8	73.0

function (see Table 3). This finding illustrates how the balance between accuracy and interpretability may be effortlessly adjusted by assigning different values to the respective coefficients. Among all models, although GARSINFIS-a only achieves the best accuracy in the testing dataset in ionosphere detection, the rest measures are still competitive to the respective winners with acceptable difference.

Table 7: Results on UCI Thyroid Dataset

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-a	98.40	95.40	97.40	3.2	4.3
GARSINFIS-a&i	97.77	94.83	96.79	**2.8**	**4.0**
C4.5	91.70	93.49	92.28	3.2	5.9
Naive Bayes	96.30	**97.91**	96.84	5	N.A.
SVM	87.72	89.96	88.47	5	N.A.
MLP	95.32	96.79	95.81	5	(7)
RBF	98.60	97.07	98.09	5	(6)
ANFIS	97.84	91.21	95.63	5	11.6
DENFIS	**100.0**	95.96	**98.65**	5	13.2
RS-POPFNN	93.18	91.67	93.10	5	25.6
RS-HeRR	95.53	92.88	94.65	3.8	29.5

Table 8: Results on UCI Ionosphere Dataset

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-a	94.70	**90.77**	93.39	8.3	19.2
GARSINFIS-a&i	93.59	90.51	92.56	7.9	18.9
C4.5	97.99	90.60	95.53	7.6	**11.5**
Naive Bayes	82.88	83.68	83.14	31	N.A.
SVM	90.34	85.56	88.75	32	N.A.
MLP	99.36	88.80	**95.84**	32	(19)
RBF	93.63	90.60	92.62	32	(6)
ANFIS	**100.0**	84.24	94.75	32	24.6
DENFIS	99.91	80.17	93.33	32	93.6
RS-POPFNN	98.21	82.39	92.94	27.1	151.3
RS-HeRR	**100.0**	87.26	95.75	**6.7**	146.2

7.3. Further Accuracy Increase Using More Complex Rules

In this subsection, GARSINFIS-1 and GARSINFIS-a are applied to the glass identification and material discrimination (sonar) datasets. As shown in Tables 9 and 10, when comparing to GARSINFIS-a, GARSINFIS-1 achieves higher accuracy by extending the zero-order TSK fuzzy rules into first-order ones. Please note that the decrease in interpretability is not represented in

Table 9: Results on UCI Glass Dataset

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-1	80.50	**68.03**	76.34	**7.1**	19.5
GARSINFIS-a	78.29	66.63	74.40	**7.1**	19.5
C4.5	62.44	66.48	63.79	8.4	20.4
Naive Bayes	46.32	49.09	47.24	9	N.A.
SVM	61.53	57.93	60.33	9	N.A.
MLP	65.62	67.18	66.07	9	(13)
RBF	80.24	66.75	75.74	9	(12)
ANFIS	83.04	60.17	75.42	9	**7.1**
DENFIS	84.37	63.53	77.43	9	21.6
RS-POPFNN	78.68	62.03	73.13	9	102
RS-HeRR	**93.56**	64.80	**83.97**	**7.1**	93.7

Table 10: Results on UCI Sonar Dataset

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS-1	92.72	72.59	86.01	8.6	21.1
GARSINFIS-a	90.70	69.38	83.53	8.6	21.1
C4.5	97.84	70.82	88.83	11.6	13.6
Naive Bayes	73.78	70.26	72.60	60	N.A.
SVM	88.40	77.31	84.70	60	N.A.
MLP	99.21	**81.21**	**93.21**	60	(33)
RBF	96.47	81.20	91.38	60	(10)
ANFIS	**100.0**	73.84	91.28	60	**4.4**
DENFIS	98.92	77.03	91.62	60	71.7
RS-POPFNN	**100.0**	70.01	90.00	24.2	137.3
RS-HeRR	**100.0**	72.85	90.95	**6.1**	120.7

the number of selected features and employed rules. It is the consequent parts of the rules become less legible 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 the best accuracy in the testing dataset in glass identification and satisfactory accuracy in the rest measures.

It is worth mentioning that in both Tables 9 and 10, RS-HeRR selects the least number of features and ANFIS employs the least number of rules. However, RS-HeRR employs significantly more number of rules than other models except RS-POPFNN, and RS-HeRR suffers much more from the over-fitting problem (manifested as the accuracy difference between training and testing datasets, e.g., in glass identification, the difference of GARSINFIS-1 is $80.5\% - 68.03\% = 12.47\%$ and that of RS-HeRR is $93.56\% - 64.8\% = 28.76\%$). Moreover, the determination of the number of clusters in FCM employed by ANFIS demands extra effort through trial-and-error. To better compare the performance of all models in various aspects, we provide a set of comprehensive comparisons in the following subsection.

7.4. Performance Benchmarks on All Datasets

The computational time taken by GARSINFIS on each dataset (we use the model shown in the first row of Tables 5 to 10) is reported in Table 11. Please note that the computational time was recorded using a notebook equipped with Intel(R) Core(TM)2 DUO CPU at 2.53GHz each and 3G physical RAM and GARSINFIS was implemented using MATLAB. Moreover, for the derivation of the theoretical computational complexity $O(NOG \cdot P \cdot nor \cdot NOF \cdot NOD^2)$, please refer to Section 5.8. As shown in Table 11, the actual computational time and the theoretical computational complexity are highly consistent with the correlation computed as 0.82. Due to the reason that although the other benchmarking models were all run using the same computer, they were implemented using different programming languages, their actual computational time was not reported in this paper for comparisons. Nonetheless, because the other benchmarking models do not employ iterative algorithms such as GA, their computational time is significantly smaller than that of GARSINFIS. However, we deem that GARSINFIS does not require excessive computational resources, because for all the UCI datasets used in this paper, GARSINFIS managed to obtain competitive accuracy with a great level of interpretability (see the latter part of this subsection) within a maximum of two hours (see Table 11),

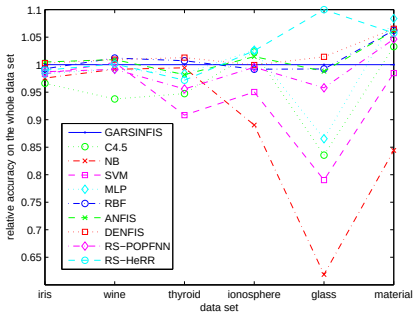
Table 11: Computational Time Taken by GARSINFIS (in second)

Dataset	Iris	Wine	Thyroid
Computational time	50.40 ± 6.37	1434.17 ± 188.33	599.65 ± 46.43
Theoretical complexity	102e+6	154e+7	177e+7
Dataset	Ionosphere	Glass	Sonar
Computational time	5123.92 ± 450.55	5439.76 ± 252.17	6957.30 ± 227.78
Theoretical complexity	168e+8	357e+7	243e+8

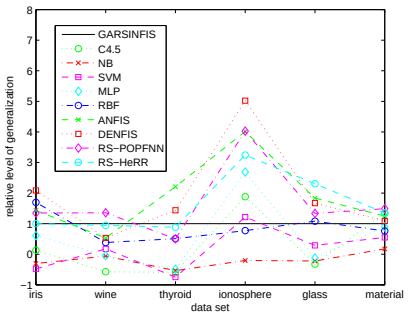
For performance comparisons between the two GARSINFIS models applied on each dataset (the models shown in the first two rows of Tables 5 to 10), we run single-factor ANOVA tests to validate whether there is a real difference in performance. The ANOVA test results suggest that all the difference between the two models in terms of accuracy (i.e., train, test and whole) is statistically significant with P-Value of 0.05.

To better benchmark the performance of GARSINFIS against other models in terms of accuracy, relative comparisons of the following measures are defined: i) Relative accuracy on the whole dataset $a^i_{\text{relative}} = a^i_{\text{whole}}/a^G_{\text{whole}}$, where i denotes the ith model and G denotes GARSINFIS (the model shown in the first row of Tables 5 to 10). This measure roughly shows the correctness and completeness of each derived model in capturing the characteristics of the whole dataset. ii) Relative level of generalization $g^i_{\text{relative}} = (a^i_{\text{train}} - a^i_{\text{test}})/(a^G_{\text{train}} - a^G_{\text{test}})$, which evaluates whether each model suffers from the over-fitting problem. iii) Relative accuracy on the testing dataset $t^i_{\text{relative}} = a^i_{\text{test}}/a^G_{\text{test}}$, which evaluates the ability of each model to predict unforeseen data. iv) Relative number of employed rules $r^i_{\text{relative}} = r^i/r^G$. v) Relative number of selected features $f^i_{\text{relative}} = f^i/f^G$. The results of these measures are shown in Figure 5.

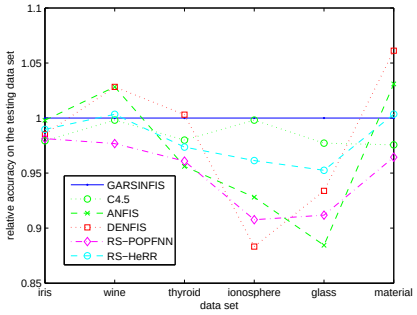
Although GARSINFIS achieves only a middle level of accuracy on the whole dataset (see Figure 5(a)) and generalization (see Figure 5(b)) among all models, it performs better than most NFISs for most datasets (except for wine and material when comparing to ANFIS and DENFIS) on the testing accuracy (see Figure 5(c)). However, ANFIS does not self-organize its network



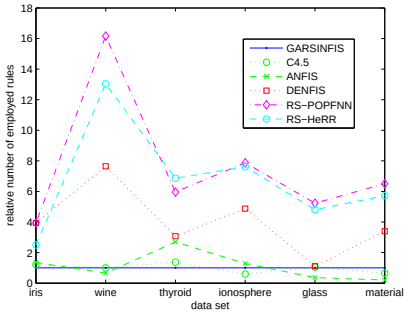
(a) Relative comparison on a^i_{relative}



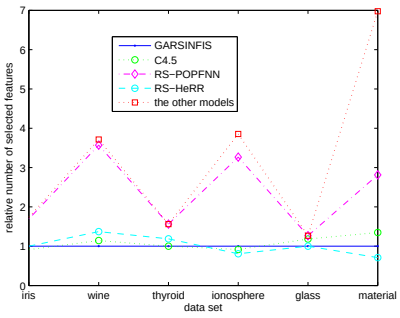
(b) Relative comparison on g^i_{relative}



(c) Relative comparison on t^i_{relative}



(d) Relative comparison on r^i_{relative}



(e) Relative comparison on f^i_{relative}

Figure 5: Visualization on the defined relative comparisons.

structure and DENFIS employs much more number of rules than GARSINFIS does (see Figure 5(d)). It is also encouraging to learn that among all NFISs

that perform knowledge reduction (i.e., RS-POPFNN, RS-HeRR and GARSINFIS), GARSINFIS always obtains comparable or better testing accuracy (see Figure 5(c)). It is clearly shown in Figure 5(d) that C4.5, GARSINFIS, and ANFIS employ lesser number of rules than the other models do. However, C4.5 uses crisp rules that highly likely lead to its inferior testing accuracy and ANFIS requires extra effort in identifying the optimal number of clusters through trial-and-error (due to the employment of FCM). GARSINFIS and RS-HeRR perform best in terms of feature selection (see Figure 5(e)). However, RS-HeRR employs significantly more number of rules. In summary, GARSINFIS produces competitive accuracy with a great level of interpretability.

The fact that GARSINFIS only achieves a competitive level of accuracy well demonstrates the trade-off between the two contradicting objectives, i.e., accuracy and interpretability. As aforementioned in the context of Figure 1, an ideal system is usually not available. GARSINFIS strives for better interpretability (instead of excellence in accuracy) without sacrificing accuracy. Nonetheless, in the following section, we show that GARSINFIS achieves high level of both accuracy and interpretability in a real-world financial application.

8. Decision Making in IPO Investments

Investing in IPO may be profitable on average, but as many other investments, it is risky and the return is subject to many determinants. In the literature of corporate finance, there are numerous empirical studies on the influencing factors of the first day or initial returns of IPOs (e.g., [67, 68]). However, existing IPO studies in the financial aspect usually focus on each individual variable's incremental effect in explaining IPO's initial return, e.g., whether certain strategy adopted by IPO issuer is a significant explanatory variable. However, to our surprise, only a couple of studies, including those in the computer science research field, have documented the corresponding decision support strategies for investments in IPOs to grab the money-left-on-the-table. Therefore, in this paper, we apply GARSINFIS on real-world IPO data and further investigate

whether it can provide interpretable and reliable decision supports in IPO investments.

8.1. Design of Experiments

Recently, the financial researchers summarised and identified the most robust determinants of IPO underpricing [3] among all that had been investigated in the literature. Based on the findings reported in [3], we select fifteen financial covariates in this study and present them in Table 12. These variables span across determinants of the intrinsic value of the stock, the sentiment of market participants, and the strategies of IPO issuers and underwriters. Furthermore, we select earning ratio of the first day return as the dependant variable, which quantifies the return of the investment.

Because all relevant IPO data are publicly available, we collected the public information (by merging multiple databases and deriving the variables listed in Table 12) of all IPOs in U.S. from 1986 to 2013 (28 years). Specifically, we followed the convention of financial studies on IPO underpricing to exclude American depositary receipts (ADRs), closed-end funds, real estate investment trusts, financial institutions (SIC codes 6000-6999), unit offerings, and IPOs with an offer price below five dollars per share. Moreover, after removing missing values, in the end, the size of the IPO dataset is 5,203. Furthermore, based on the dependant variable, i.e., earning ratio of the first day return $r_i^{fdr} = \frac{P_i^{fdc} - P_i^{IPO}}{P_i^{IPO}} \times 100\%$, where P_i^{fdc} denotes the first day closing price of the ith IPO and P_i^{IPO} denotes the offering price of the ith IPO, we categorize all data samples into three intuitive categories as listed in Table 13.

In the literature of finance, researchers mostly use regression models to test the significance of individual variables. In these regression models, every variable is assigned with a corresponding coefficient, i.e., every input feature takes into account. Therefore, in financial decision support systems, if we assume the investors have primitive financial background knowledge, they would not mind the decision rules employing many input features and each rule consisting of many arguments. What they really would mind are the unnecessarily large

Table 12: List of Financial Covariates Used in IPO Underpricing Prediction

ID	Variable	Description
1	Ln of firm sale	Ln of annual firm sales (REVT) reported within one year prior to IPO issue date
2	Offer price revision	$100(\frac{\text{Offer Price - Original Middle Filling Price Range}}{\text{Original Middle Filling Price Range}})$, where Original Middle Filling Price Range = $\frac{1}{2}$(Original Low Filling Price + Original High Filling Price)
3	Ln of news stories	Ln(1 + News Stories) where News Stories = Fulltext search hits of the IPO company name in the 6 months prior to the IPO issue date.
4	Total liab to asset ratio	The ratio of Total Liabilities (LT) to Total Assets (AT) reported within one year prior to the IPO issue date.
5	Investment bank market share	IB Mkt $\text{Share}_{i,t} = 100(\frac{\text{IB Proceeds}_{i,t}}{\text{Total IPO Proceeds}_t})$ for investment bank i and year t
6	Avg undprcg in prv 30 days	Average IPO first trading day return in the 30 days prior to the IPO issue date
7	Avg prc rvs in prv 30 days	Average Offer Price Revision of IPOs in 30 days prior to the IPO issue date
8	Prior 30 day CRSP EW index	$\widehat{\mu}_t^{CRSP} = \frac{1}{30}\sum_{i=t-31}^{t-1}$CRSP Equal Weighted Index Return_i, where t is the IPO issue date
9	Ln(1 + shrs rtnd/shrs ofrd)	$\text{Ln}(1+\frac{\text{Secondary Shares Retained}}{\text{Shares Offered}})$, where Secondary Shares Retained = Shares Outstanding Total Shares Sold (includes overallotment shares)
10	Offer revision from orgnl flng	Equals Offer Price Revision if Offer Price Revision < 0, otherwise = 0.
11	Ln inds mkt value to sales	Rolling 12 month average of the industry market value to sales ratio
12	Ln price to sales ratio	$\text{Ln}(\frac{\text{Offer Price + Shares Outstanding}}{\text{Annual Firm Sales}})$, where Annual Firm Sales (REVT) are reported within one year prior to IPO issue date
13	Prior 30 days industry rtrn	$\widehat{\mu}_{j,t}^{FFlnd} = \frac{1}{30}\sum_{i=t-31}^{t-1}$Fama French Industry $\text{Return}_{i,j}$, where t is IPO issue date and j is one of 49 Industry Groups
14	Prior 30 days SD of industry rtrn	Standard deviation of $\widehat{\mu}_{j,t}^{FFlnd}$
15	Prior 30 days NASDAQ rtrn	$\widehat{\mu}_t^{NASDAQ} = \frac{1}{30}\sum_{i=t-31}^{t-1}$NASDAQ composite return_i, where t is the IPO issue date

Table 13: IPO Categorization Based on Their First Day Return

ID	Characteristic	# samples	Percentage	Categorization criterion
1	Not worth buying	1,425	27.39%	$r_i^{fdr} \leq 0$
2	Worth buying	2,463	47.34%	$0 < r_i^{fdr} < 25\%$
3	Definitely worth buying	1,315	25.27%	$25\% \leq r_i^{fdr}$

number of rules, which might be overwhelming in a negative sense. Therefore, based on these preferences, we assign the coefficient parameter values in (5) accordingly that $\tau_1 = \tau_3 = \tau_5 = 1$ and $\tau_2 = \tau_4 = 0$, i.e., the fitness function in this IPO underpricing study is set as $f(x) = (1-a)\frac{NOD}{K} + \frac{nor}{NOD} + \frac{mse}{NOF}$. Moreover, to avoid degrading the interpretability, GARSINFIS employs zero-order rules.

We use the same set of constraints and parameter values as those reported in Section 7, except we set population size to 100 and the number of generations to 10. Furthermore, all the benchmarking models used in Section 7 are also used in this study. In addition, we include another two benchmarking models, namely linear regression and Quintana's model [28]. Because Quintana's model actually employs a pool of rules, which consists of more number of rules than the population size (accumulated across all generations), in this paper, we set its number of rules to its population size and assign the same population size and number of generations as GARSINFIS for comparison purposes. Furthermore, to demonstrate the capability of GARSINFIS in terms of discovering the most essential knowledge to perform accurate predictions on unseen data, we randomly select 20% of the IPO data for training and the remaining for testing.

8.2. Experimental Results

The averaged results of ten independent runs are shown in Table 14. It is encouraging to find that GARSINFIS achieves the best accuracy on the testing datasets with the least number of selected features (even though we did not specifically construct the fitness evaluation function to minimize the number of selected features). Furthermore, although GARSINFIS employs more number of rules than ANFIS, both of them employs significantly much lesser number

Table 14: Experimental Results on IPO Underpricing Prediction

Model	Train%	Test%	Whole%	# Fea.	# Rule
GARSINFIS	53.80	**51.21**	51.63	**9.5**	15.5
C4.5	84.44	46.94	54.44	15	112
Naive Bayes	47.76	48.97	48.73	15	N.A.
SVM	53.58	50.40	51.04	15	N.A.
MLP	69.91	49.61	53.67	15	(26)
RBF	78.82	47.64	53.88	15	(9)
ANFIS	82.51	47.50	54.50	15	**11.2**
DENFIS	84.67	48.47	55.71	15	64.3
RS-POPFNN	85.15	48.85	56.11	13.9	170.3
RS-HeRR	**87.49**	49.71	**57.27**	10.2	155.6
Regression	56.31	50.77	51.88	15	N.A.
Quintana [28]	49.44	41.53	43.11	14.4	100

of rules than the other rule-based models. However, please recall that in this study, ANFIS employs FCM clustering method, whose cluster number needs to be predetermined by trial-and-error. In addition, ANFIS is fully connected and GARSINFIS is partially connected (see Figure 4). Therefore, ANFIS may not be more interpretable than GASINFIS in this IPO underpricing prediction. However, ANFIS certainly achieves lower prediction accuracy on unseen data samples. Comparing GARSINFIS to Quintana's model [28], which is the only rule-based model that had been applied to predict IPO underpricing, GARSINFIS outperforms Quintana's model in every aspect. This finding is not surprising mainly due to the following two reasons: i) Quintana's model employs interval-based crisp rules, which may not perform well on unseen data and ii) Quintana's model may require a significantly larger pool of rules (contradictory to the interpretability requirement of financial decision support systems) to improve its accuracy.

The averaged computational time taken by GARSINFIS is less than four hours, i.e., 14039.95±2788.55 seconds (using the same computer as reported in Section 7.4). In IPO underpricing prediction, GARSINFIS spent significantly

more amount of computational time than the other benchmarking models did, including Quintana's model [28], which also employs the same iterative algorithm, i.e., GA. However, the time consuming optimization procedures employed by GASINFIS are highly effective because it achieves the best accuracy on the testing dataset and subsequently yields the highest possible profits in IPO first-day returns (see Section 8.4). Thus, for low frequency financial transactions, such as investments in IPOs, GARSINFIS is definitely a trustworthy decision support system that fulfils the response time requirement, especially the processing time of GARSINFIS is significantly much smaller (due to the employment of a reducted inference rule set) than its training time (due to the employment of an iterative optimization algorithm).

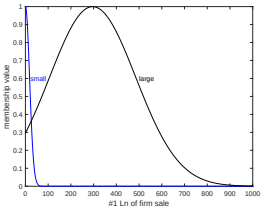
8.3. Presentation of Derived Interpretable Rules

To demonstrate the interpretability of the rules derived by GARSINFIS, we select one set of rules derived in one of the ten runs as examples. Specifically, we visualize the generated fuzzy membership functions in each of the nine selected features in Figure 6 and list all the ten derived rules in Table 15. This set of derived rules achieve 51.64% accuracy on the testing dataset. As clearly shown in Table 15 that this set of rules possess high-level interpretability, not only because they use less number of features and employ less number of rules (see Table 14), but also because each rule is more legible that on average, each rule only consists of 3.2 arguments in the antecedent part.

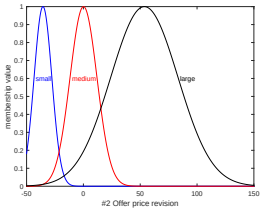
8.4. Further Investigations on Financial Returns

To intuitively show the performance of each model in terms of financial returns in IPO investments, we simply implement the following strategies:

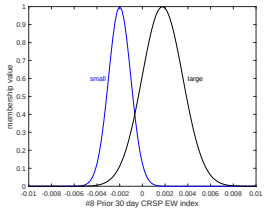
1. If the decision support system suggests an IPO is definitely worth buying, an investor will invest $x\%$ of available money.
2. If the decision support system suggests an IPO is worth buying, an investor will invest $\frac{x}{2}\%$ of available money.



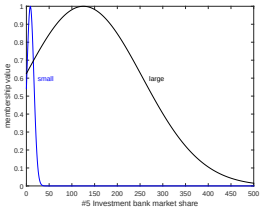
(a) #1 Ln of firm sale



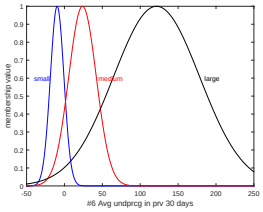
(b) #2 Offer price revision



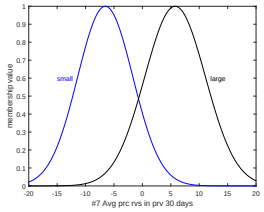
(c) #3 Ln of news stories



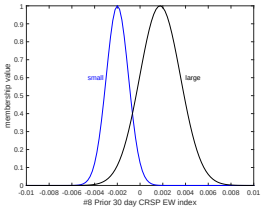
(d) #5 Investment bank market share



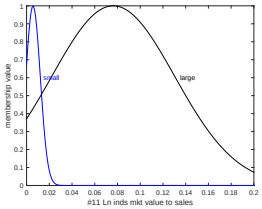
(e) #6 Avg undprcg in prv 30 days



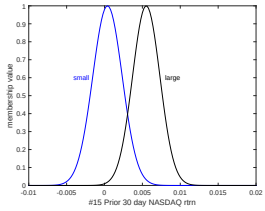
(f) #7 Avg prc rvs in prv 30 days



(g) #8 Prior 30 day CRSP EW index



(h) #11 Ln inds mkt value to sales



(i) #15 Prior 30 day NASDAQ rtrn

Figure 6: Visualization on one set of the generated fuzzy membership functions.

765 3. If the decision support system suggests an IPO is not worth buying, an investor will not invest.

Moreover, to distinguish each model's prediction performance for different types of investors, we further set different values of x to the following three investment behavioral patterns: i) for progressive investors, $x = 80$, ii) for normal investors, 770 $x = 40$, and iii) for conservative investors, $x = 20$. Based on these strategies, we compute how much returns (accumulated initial returns) each type of the investor may yield on IPO investments by following the decision support sys-

Table 15: One Set of the Derived Interpretable Fuzzy Rules

ID	Rule interpretation
1	**IF** firm sales (#1) is large ∧ offer price revision (#2) is medium ∧ number of news (#3) is small ∧ IB market share (#5) is small ∧ prior underpricing (#6) is small ∧ NASDAQ return (#15) is small, **THEN** it is not worth buying (i.e., do not invest)
2	**IF** offer price revision (#2) is small ∧ number of news (#3) is small ∧ NASDAQ return (#15) is small, **THEN** it is not worth buying (i.e., do not invest)
3	**IF** number of news (#3) is small ∧ IB market share (#5) is large ∧ prior offer price revision (#7) is large ∧ industry v2s ratio (#11) is large, **THEN** it is not worth buying (i.e., do not invest)
4	**IF** number of news (#3) is small ∧ prior underpricing (#6) is medium ∧ prior offer price revision (#7) is small ∧ CRSP EW index (#8) is small, **THEN** it is not worth buying (i.e., do not invest)
5	**IF** number of news (#3) is small ∧ CRSP EW index (#8) is small ∧ industry v2s ratio (#11) is large, **THEN** it is not worth buying (i.e., do not invest)
6	**IF** offer price revision (#2) is small ∧ number of news (#3) is large ∧ IB market share (#5) is large ∧ prior offer price revision (#7) is small, **THEN** it is worth buying (i.e., invest moderately)
7	**IF** offer price revision (#2) is large ∧ number of news (#3) is large, **THEN** it is definitely worth buying (i.e., invest progressively)
8	**IF** offer price revision (#2) is large ∧ IB market share (#5) is large, **THEN** it is definitely worth buying (i.e., invest progressively)
9	**IF** offer price revision (#2) is large ∧ prior offer price revision (#7) is large, **THEN** it is definitely worth buying (i.e., invest progressively)
10	**IF** prior underpricing (#6) is large ∧ NASDAQ return (#15) is large, **THEN** it is definitely worth buying (i.e., invest progressively)

tem's suggestions only. The final financial returns are listed in Table 16. In the beginning of each simulation, an investor gets 100 dollars as the seed money.

We were surprised on the first sight of the seemingly overly large amount of returns listed in Table 16. To further validate the correctness of these results, we implement a purely random investment strategy that each time an investor will not invest, invest $\frac{x}{2}\%$ of available money, or invest $x\%$ of available money with equal probability. It turns out that even this random investment

Table 16: Theoretical Financial Returns on IPO Investments

Model	Prospective	Normal	Conservative
GARSINFIS	**1.12e+56**	**4.03e+30**	**4.65e+16**
C4.5	6.57e+52	8.24e+28	3.36e+15
Naive Bayes	9.22e+54	5.21e+29	7.66e+15
SVM	5.88e+55	8.50e+29	1.35e+16
MLP	3.81e+55	7.55e+29	1.30e+16
RBF	8.72e+51	5.79e+28	1.63e+15
ANFIS	9.53e+52	9.65e+28	3.82e+15
DENFIS	9.74e+50	1.89e+28	1.00e+15
RS-POPFNN	8.00e+53	2.97e+29	5.50e+15
RS-HeRR	1.55e+53	1.18e+29	4.03e+15
Regression	7.28e+55	2.70e+30	2.79e+16
Quintana [28]	2.95e+49	3.18e+27	9.13e+14

strategy may yield financial returns on the magnitude of e+47, e+26 and e+14 according to the respective investment behaviors. This large amount of financial returns based on random investment decisions on some level shows the huge amount of money-left-on-the-table in U.S. IPOs over 28 years. Nonetheless, all the decision support systems listed in Table 16 perform better than the random strategies. It is encouraging to see that GARSINFIS may help an investor to yield the largest amount of financial returns. It is also worth noting that the small amount of difference in accuracy on the testing datasets (see Table 14) may transform into a huge amount of difference in financial returns, e.g., comparing GARSINFIS with regression, the averaged difference in testing accuracy is $51.21\% - 50.77\% = 0.44\%$, however, the averaged difference in financial returns is 1.12e+56−7.28e+55= 3.92e+55. To further investigate whether the potential financial returns yielded by each model are statistically different, for each type of investors (different columns in Table 16), we sort the financial returns and apply single-factor ANOVA tests on the adjacent values. The ANOVA tests suggest that all the financial returns yielded by various models are significantly different with P-Value of 0.05.

Please note that the potential investment profits listed in Table 16 are computed based on the theoretical basis. IPO investments in the real world may be affected by many factors such as over-subscriptions, IPO volume, information asymmetry, transaction fees, etc.

9. Conclusion

In this paper, we introduce a hybrid intelligent system termed genetic algorithm and rough set incorporated neural fuzzy inference system (GARSINFIS), which may function as a data-driven decision support system. We first illustrate how the trade-off between accuracy and interpretability may be easily leveraged in GARSINFIS using well-known benchmarking datasets. We then focus on applying GARSINFIS to grab money-left-on-the-table in IPOs. Empirical studies show that GARSINFIS outperforms the other benchmarking models in the prediction of IPO underpricing and may yield the most amount of financial returns in IPO investments. The highly interpretable yet highly reliable rules derived by GARSINFIS may be well accepted by interested investors.

To further improve the accuracy of GARSINFIS in financial applications, we will look into the employment of asymmetric membership functions to better characterize financial covariates, which are normally positively skewed [69].

Acknowledgement

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.

References

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

[2] S.-M. Zhou, J. Q. Gan, Low-level interpretability and high-level interpretability: A unified view of data-driven interpretable fuzzy system modelling, Fuzzy Sets and Systems 159 (23) (2008) 3091–3131.

[3] A. W. Butler, M. O. Keefe, R. Kieschnick, Robust determinants of IPO underpricing and their implications for IPO research, Journal of Corporate Finance 27 (2014) 367–383.

[4] M. Setnes, R. Babuska, H. B. Verbruggen, Complexity reduction in fuzzy modeling, Mathematics and Computers in Simulation 46 (5-6) (1998) 507–516.

[5] M. J. Gacto, R. Alcala, 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 18 (3) (2010) 515–531.

[6] W. E. Combs, J. E. Andrews, Combinatorial rule explosion eliminated by a fuzzy rule configuration, IEEE Transactions on Fuzzy Systems 6 (1) (1998) 1–11.

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

[8] S. Guillaume, B. Charnomordic, Generating an interpretable family of fuzzy partitions from data, IEEE Transactions on Fuzzy Systems 12 (3) (2004) 324–335.

[9] K. K. Ang, C. Quek, RSPOP: Rough set-based pseudo outer-product fuzzy rule identification algorithm, Neural Computation 17 (1) (2005) 205–243.

[10] F. Liu, C. Quek, G. S. Ng, A novel generic Hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system, Neural Computation 19 (6) (2007) 1656–1680.

[11] L. Chen, C. L. P. Chen, W. Pedrycz, A gradient-descent-based approach for transparent linguistic interface generation in fuzzy models, IEEE Transactions on Systems, Man and Cybernetics, Part B 40 (5) (2010) 1219–1230.

[12] E. Y. Cheu, C. Quek, 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 23 (2) (2012) 317–329.

[13] T. Z. Tan, G. S. Ng, C. Quek, A novel biologically and psychologically inspired fuzzy decision support system: Hierarchical complementary learning., IEEE/ACM Transactions on Computational Biology and Bioinformatics 5 (1) (2008) 67–79.

[14] W. L. Tung, C. Quek, eFSM–a novel online neural-fuzzy semantic memory model, IEEE Transactions on Neural Networks 21 (1) (2010) 136–157.

[15] M. Cococcioni, P. Ducange, B. Lazzerini, F. Marcelloni, A Pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems, Soft Computing 11 (11) (2007) 1013–1031.

[16] H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning, International Journal of Approximate Reasoning 44 (1) (2007) 4–31.

[17] A. Botta, B. Lazzerini, F. Marcelloni, D. C. Stefanescu, Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index, Soft Computing 13 (5) (2008) 437–449.

[18] D. Wang, C. Quek, G. S. Ng, Bank failure prediction using an accurate and interpretable neural fuzzy inference system, AI Communications 29 (4) (2016) 477–495.

[19] P. Cerchiello, P. Giudici, G. Nicola, Twitter data models for bank risk contagion, Neurocomputing 264 (2017) 50–56.

[20] S. Ronnqvist, P. Sarlin, Bank distress in the news: Describing events through deep learning, Neurocomputing 264 (2017) 57–70.

[21] Y. Shynkevich, T. M. McGinnity, S. A. Coleman, A. Belatreche, Y. Li, Forecasting price movements using technical indicators: Investigating the impact of varying input window length, Neurocomputing 264 (2017) 71–88.

[22] S. Wang, F.-L. Chung, J. Wu, J. Wang, Least learning machine and its experimental studies on regression capability, Applied Soft Computing 21 (8) (2014) 677–684.

[23] B. A. Jain, B. N. Nag, Artificial neural network models for pricing initial public offerings, Decision Sciences 26 (3) (1995) 283–302.

[24] S. J. Robertson, B. L. Golden, G. C. Runger, E. A. Wasil, Neural network models for initial public offerings, Neurocomputing 18 (1-3) (1998) 165–182.

[25] B. Reber, B. Berry, S. Toms, Predicting mispricing of initial public offerings, Intelligent Systems in Accounting, Finance and Management 13 (2005) 41–59.

[26] K. Yao, L. Zhou, Analysis of influencing factors of IPO underpricing based on rough set and support vector machine, in: Proceedings of International Conference on Information Management, Innovation Management and Industrial Engineering, 2012, pp. 244–248.

[27] D. Quintana, Y. Saez, P. Isasi, Random forest prediction of IPO underpricing, Applied Sciences 7 (6) (2017) 636.

[28] D. Quintana, C. Luque, P. Isasi, Evolutionary rule-based system for IPO underpricing prediction, in: Proceedings of Annual Conference on Genetic and Evolutionary Computation, 2005.

[29] Z. Pawlak, Rough sets, International Journal of Information and Computer Science 11 (5) (1982) 341–356.

[30] Q. Shen, A. Chouchoulas, Rough set-based dimensionality reduction for supervised and unsupervised learning, International Journal of Applied Mathematics and Computer Science 11 (3) (2001) 583–601.

[31] L. A. Zadeh, Fuzzy sets, Information Control 8 (1965) 338–353.

[32] Z. Pawlak, Rough Sets, Kluwer Academic Publishers, 1991.

[33] D. Wang, A.-H. Tan, Self-regulated incremental clustering with focused preferences, in: Proceedings of International Joint Conference on Neural Networks, IEEE, 2016, pp. 1297–1304.

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

[35] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[36] D. Wang, C. Quek, A.-H. Tan, C. Miao, G. S. Ng, Y. Zhou, Leveraging the trade-off between accuracy and interpretability in a hybrid intelligent system, in: Proceedings of International Conference on Security, Pattern Analysis, and Cybernetics, 2017, pp. 55–60.

[37] I. Eccles, 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.

[38] R. P. Paiva, A. Dourado, Interpretability and learning in neuro-fuzzy systems, Fuzzy Sets and Systems 147 (1) (2004) 17–38.

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

[40] B. L. Miller, D. E. Goldberg, Genetic algorithms, tournament selection, and the effects of noise, Complex Systems 9 (1995) 193–212.

[41] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modelling and control, IEEE Transactions on Systems, Man and Cybernetics, Part B 15 (1) (1985) 116–132.

[42] M. Sugeno, G. T. Kang, Structure identification of fuzzy model, Fuzzy Sets and Systems 28 (1988) 13–33.

[43] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Scalable TSK fuzzy modeling for very large datasets using minimal-enclosing-ball approximation, IEEE Transactions on Fuzzy Systems 19 (2) (2011) 210–226.

[44] Y. Jiang, F.-L. Chung, H. Ishibuchi, Z. Deng, S. Wang, Multitask TSK fuzzy system modeling by mining intertask common hidden structure, IEEE Transactions on Cybernetics 45 (3) (2015) 534–547.

[45] Z. Deng, L. Cao, Y. Jiang, S. Wang, Minimax probability TSK fuzzy system classifier: A more transparent and highly interpretable classification model, IEEE Transactions on Fuzzy Systems 23 (4) (2015) 813–826.

[46] Y. Jiang, Z. Deng, F.-L. Chung, S. Wang, Realizing two-view TSK fuzzy classification system by using collaborative learning, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (1) (2017) 145–160.

[47] Y. Jiang, Z. Deng, F.-L. Chung, G. Wang, P. Qian, K.-S. Choi, S. Wang, Recognition of epileptic EEG signals using a novel multiview TSK fuzzy system, IEEE Transactions on Fuzzy Systems 25 (1) (2017) 3–20.

[48] M. Luo, F. Sun, H. Liu, Hierarchical structured sparse representation for T-S fuzzy systems identification, IEEE Transactions on Fuzzy Systems 21 (6) (2013) 1032–1043.

[49] D. Wang, C. Quek, G. S. Ng, Novel self-organizing Takagi Sugeno Kang fuzzy neural networks based on ART-like clustering, Neural Processing Letters 20 (1) (2004) 39–51.

[50] M. F. Azeem, M. Hanmandlu, N. Ahmad, Generalization of adaptive neuro-fuzzy inference systems, IEEE Transactions on Neural Networks 11 (6) (2000) 1332–1346.

[51] C. Mencar, A. M. Fanelli, Interpretability constraints for fuzzy information granulation, Information Sciences 178 (24) (2008) 4585–4618.

[52] J. M. Alonso, L. Magdalena, G. Gonzalez-Rodriguez, Looking for a good fuzzy system interpretability index: An experimental approach, International Journal of Approximate Reasoning 51 (1) (2009) 115–134.

[53] M. J. Gacto, R. Alcala, F. Herrera, Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, Information Sciences 181 (20) (2011) 4340–4360.

[54] G. A. Miller, The magical number seven, plus or minus two: Some limits on our capacity for processing information, Psychological Review 63 (2) (1956) 81–97.

[55] A. Baddeley, N. Thomson, M. Buchanan, Word length and the structure of short-term memory, Journal of Verbal Learning and Verbal Behavior 14 (6) (1975) 575–589.

[56] R. Conrad, A. J. Hull, Information, acoustic confusion and memory span, British Journal of Psychology 55 (4) (1964) 429–432.

[57] E. H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic synthesis, IEEE Transactions on Computers 26 (12) (1977) 1182–1191.

[58] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

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

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

[61] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273297.

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

[63] J. S. R. Jang, ANFIS: Adaptive network-based fuzzy inference systems, IEEE Transactions on Systems, Man and Cybernetics, Part B 23 (1993) 650–684.

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

[65] N. K. Kasabov, Q. Song, DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144–154.

[66] Q. Song, 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.

[67] J. R. Ritter, I. Welch, A review of IPO activity, pricing, and allocations, The Journal of Finance 57 (4) (2002) 1795–1828.

[68] T. Loughran, J. R. Ritter, Why has IPO underpricing changed over time?, Financial Management 33 (3) (2004) 5–37.

[69] E. B. Deakin, Distributions of financial accounting ratios: Some empirical evidence, Accounting Review 51 (1976) 90–96.