Bank failure prediction using an accurate and interpretable neural fuzzy inference system

Wang, Di; Ng, Geok See; Quek, Chai

2016

Wang, D., Quek, C., & Ng, G. S. (2016). Bank failure prediction using an accurate and interpretable neural fuzzy inference system. *AI Communications*, 29(4), 477-495.
doi:10.3233/AIC-160702

<https://hdl.handle.net/10356/89642>

<https://doi.org/10.3233/AIC-160702>

© 2016 IOS Press. This is the author created version of a work that has been peer reviewed and accepted for publication by *AI Communications*, IOS Press. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [<http://dx.doi.org/10.3233/AIC-160702>].

Downloaded on 26 Apr 2023 17:59:56 SGT

Bank Failure Prediction Using an Accurate and Interpretable Neural Fuzzy Inference System

Di Wang^a, Chai Quek^{a,*} and Geok See Ng^b

^a *School of Computer Engineering, Nanyang Technological University, Singapore 639798*

^b *School of Information Systems Technology & Design, Singapore University of Technology & Design, Singapore 138682*

Abstract. Bank failure prediction is an important study for regulators in the banking industry because the failure of a bank leads to devastating consequences. If bank failures are correctly predicted, early warnings can be sent to the responsible authorities for precaution purposes. Therefore, a reliable bank failure prediction or early warning system is invaluable to avoid adverse repercussion effects on other banks and to prevent drastic confidence losses in the society. In this paper, we propose a novel self-organizing neural fuzzy inference system, which functions as an early warning system of bank failures. The system performs accurately based on the auto-generated fuzzy inference rule base. More importantly, the simplified rule base possesses a high level of interpretability, which makes it much easier for human users to comprehend. Three sets of experiments are conducted on a publicly available database, which consists of 3635 United States banks observed over a 21-year period. The experimental results of our proposed model are encouraging in terms of both accuracy and interpretability when benchmarked against other prediction models.

Keywords: Automatic forecasting, bank failure prediction, early warning system, interpretability, fuzzy neural networks

1. Introduction

Bank failure prediction is of great importance to a bank's policy-makers, regulators, and clients. There is no doubt that the failure or collapse of a bank leads to devastating consequences and widespread repercussions on other banks and fi-

ancial institutions. Some of the negative impacts are the massive bailout cost for a failing bank and the negative sentiments and loss of confidence developed by investors and depositors. Generally speaking, bank failures are due to financial distress. In the past several years, many countries have experienced significant banking sector problems with the United States sub-prime crisis and weak economy being the most possible causes. According to the Federal Deposit Insurance Corporation (FDIC) [9], from 2008 to 2011, there are 414 banks closed in United States only. The overall deposits of the 414 failed banks are over 481 million US dollars and the overall assets are over 668 million. The estimated overall financial losses of the 414 failed banks are over 89 million. There are 157 bank failures in 2010 only, which is the highest number since 1992. The increase in bank failures has rekindled the interest in the prediction or early warning of bank failures.

A reliable early warning system of bank failures requires an accurate prediction model. Most prediction models in the literature are based on statistical studies. The pioneers of the statistical approach are proposed by Beaver [2] and Altman [1]. Beaver is one of the first researchers to use financial statements to predict bankruptcy and his study is based on one financial ratio (univariate) at a time. Altman, on the other hand, uses a number of financial ratios (multivariate) as inputs to predict bankruptcy. The Multivariate Discriminant Analysis (MDA) method [1] employs a discriminant function to classify the firms into their respective groups. The discriminant function is essen-

*Corresponding author: E-mail: ashcquek@ntu.edu.sg

tially a linear combination of independent financial ratios of the firm to minimize misclassifications. The MDA method is widely adopted in the literature and it is based on the assumption that any two different classes have Gaussian distributions with equal covariance matrices. The Recursive Partitioning Algorithm (RPA) [11], which is a computerized and non-parametric technique to construct a classification tree (precedent to C4.5 [31]) for the prediction of firm insolvencies, is found to outperform MDA in most cases. However, the authors of [11] also observe that additional information can be derived by assessing both RPA and MDA. Other well-known statistical models proposed in the literature include the logistic regression approach [26], which is essentially a linear sigmoid model that functions like a single-neuron network, and the Cox's proportional hazards model [19,5], which employs an estimated proportional hazard function based on the hazard rate of an average-performing bank. All the above introduced models have their own deficiencies and a "satisfactory model has yet to be developed" [38].

Unlike statistical models, the neural network approach [20] does not make assumptions on data distributions. Furthermore, it does not require any rigid restrictions on the use of input and output functions other than being continuous and differentiable [38]. Although the neural network approach provides superior results to the traditional statistical models [38], it offers little explanatory capability. The trained weight values associated with respective linkages of the neural network are simply numbers used for computations rather than meaningful indicators for human users to comprehend. This is the reason why majority neural networks function as black boxes [3]. A reliable model, which is as robust as a neural network and offers better interpretability, is thus desired.

Neural fuzzy inference system [18,22] (fuzzy neural network is used interchangeably in the literature) combines the learning capabilities of neural networks and the transparent properties of fuzzy systems together by performing respective fuzzy or non-fuzzy operations in each layer of the network. The objective of such soft computing approaches [16] is to synthesize the human ability to tolerate and handle uncertain, imprecise, and ambiguous information in the decision-making process. Some Neural Fuzzy Inference Systems (NFISs) self-organize their structures, which are normally

realized by utilizing the clustering results obtained from the training data sets [42]. In this way, NFIS saves a great amount of efforts on the determination of the system structure and the tuning of fuzzy rules, which are the two major overheads of traditional fuzzy systems. Furthermore, NFIS provides semantically meaningful linguistic inference rule base rather than the "black box" offered by most neural networks. The high-level linguistic fuzzy rules are represented in the IF-THEN form, which makes NFIS extremely intuitive and effortlessly comprehensive to the human users. In the bank failure prediction application, NFIS can be applied to identify the inherent characteristics of the failed banks and thus allows us to semantically and numerically understand the financial distress that leads to a bank failure. In the literature, there are a series of promising NFISs proposed to forecast bank failures [41,39,40,25,30]. Although these models are accurate in prediction, none of them focuses on the improvement of interpretability. Some of them employ unnecessarily larger number of fuzzy rules and all of them utilize all financial covariates given in the data set.

On the other hand, our novel proposed model automatically constructs a simplified inference rule base and at the same time obtains high-level prediction accuracy, i.e. our proposed model is intuitively more comprehensive and is still highly competitive in terms of accuracy. Furthermore, only a limited number of high-level control parameters and necessary constraints are required before our proposed system iteratively optimizes the self-generated inference rule base without any expert guidance or human intervention. After adequate exploration (facilitated by the genetic algorithm), our proposed system obtains a concise yet highly reliable inference rule base to function as an early warning system for bank failure predictions. It is more encouraging to show in the later sections of this paper that the auto-generated rules are consistent with expert knowledge and the performance of our proposed system gets more reliable with the increase of the prediction time of bank failures.

The rest of this paper is organized as follows. Sections 2 and 3 respectively describe two techniques, which are the foundations to introduce our proposed clustering method. Section 4 provides the details of our proposed clustering method. Section 5 briefly defines the system architecture of our proposed model, which employs the fuzzy rules

automatically obtained by the proposed clustering method. Section 6 presents the experimental results on bank failure predictions and analytically compares the results of our proposed model against other benchmarking models. Section 7 concludes this paper and recommends possible further improvements.

2. Rough Set Theory for Knowledge Reduction

Rough sets [28] are always compared to fuzzy sets [45]. To describe belongingness, fuzzy sets use straightforwardly defined membership functions, while rough sets use relative relations denoted as the lower and upper approximations. Both theories aim to achieve the same type of goal [44], however, it is always better to have them both in one system to take advantages of their complements [29].

2.1. Knowledge Representation System and Decision Table

Rough set theory is initiated to model relations in a given data set or knowledge base [28]. It is a formally defined methodology that can be applied to reduce the dimensionality of a given data set [34], preceding the training of an inference system.

To express mathematically how rough set theory is applied in knowledge reduction, decision logic language is used to model the Knowledge Representation System (KRS). Such a system is represented in a pair $S = (U, A)$, where U is a nonempty and finite set denoted as the universe of discourse and A is a nonempty and finite set of primitive attributes.

Decision tables can be defined in terms of KRS. If we have $S = (U, A)$ and $C, D \subset A$, which denote the condition and decision attributes respectively, then S with distinguished (C, D) pairs is essentially a decision table, i.e. $T = \{U, C, D\}$ or T is a CD-based decision table. Moreover, each element in U is a CD-based decision rule and it represents a cluster of data. Subsequently, U is the union of all clusters found in the data set. A simple example of decision table is given in Table 1 to illustrate the concept and a more informative numerical example of decision table is given in Table 2.

Table 1

An example to illustrate the concept of a decision table.

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

Table 2

A numerical example of a decision table.

U	a	b	c	d	e
1	1	0	0	1	$\rightarrow$ 1
2	1	0	0	0	$\rightarrow$ 1
3	0	0	0	0	$\rightarrow$ 0
4	1	1	0	1	$\rightarrow$ 0
5	1	1	0	2	$\rightarrow$ 2
6	2	1	0	2	$\rightarrow$ 2
7	2	2	2	2	$\rightarrow$ 2

2.2. Indiscernible Relation and Approximations

Indiscernible relation over knowledge K , denoted as $IND(K)$, is defined in Eq. (1). The family of all equivalence classes of the equivalence relation $IND(K)$ is denoted as $U/IND(K)$.

$$IND(K) = \{(x, y) \in U^2 \mid \forall r \in K, r(x) = r(y)\}. \quad (1)$$

Rough set theory approximates knowledge using a pair of relational approximations. The lower and upper approximations of a set, when given an equivalence relation $IND(K)$, are defined in Eqs. 2 and 3 respectively. By presenting knowledge K , lower approximation $\underline{K}Y$ is the set of elements that can be certainly classified by K and Y . Upper approximation $\overline{K}Y$ is the set of elements that can be possibly classified by K and Y .

$$\underline{K}Y = \bigcup \{X : X \in U/IND(K), X \subseteq Y, Y \subseteq U\}, \quad (2)$$

$$\overline{K}Y = \bigcup \{X : X \in U/IND(K), X \cap Y \neq \emptyset, X \subseteq Y, Y \subseteq U\}. \quad (3)$$

According to Table 2, we can define three sets based on the decision attribute e : $E_0 = \{x_3, x_4\}$, $E_1 = \{x_1, x_2\}$, and $E_3 = \{x_5, x_6, x_7\}$. Therefore, if the relation $R_1 = \{x_1, x_4\}$ (based on the condition attribute d) is given, then $\underline{K}R_1 = \emptyset$ and $\overline{K}R_1 = E_0 \cup E_1 = \{x_1, x_2, x_3, x_4\}$. Similarly, if $R_2 = \{x_5, x_6, x_7\}$, then $\underline{K}R_2 = \overline{K}R_2 = E_3 = \{x_5, x_6, x_7\}$.

2.3. Attribute Reduction and Feature Selection

Rough set theory performs knowledge reduction with two fundamental concepts, reduct and core. Intuitively, a reduct of knowledge is an essential subset of knowledge that suffices to define all basic relations, whereas a core is the most fundamental subset of knowledge that consists of the common attributes of all reducts.

Given a decision table $T = \{U, C, D\}$, an attribute a is dispensable if and only if $IND(C) = IND(C - \{a\})$. Otherwise, a is indispensable. The family C is independent if $\forall a \in C$ is indispensable in C . Attribute reduction is performed during the process of finding independent C with minimum cardinality. If for all elements in U , an attribute b is dispensable, then b can be removed from the training data set. This process of eliminating attributes that do not contribute any information for inferences is considered as feature selection, which is extremely useful to handle problems with high dimensionality.

Based on indispensable relations, reduct and core are defined as follows. $Q \subseteq P$ is a reduct of P , if Q is independent and $IND(P) = IND(Q)$. The core of P is defined as the common parts of all reducts, i.e. $CORE(P) = \bigcap REDUCT(P)$. It is easy to infer that P can have many reducts, but only has one core.

The decision table, which is derived by removing all dispensable attributes from Table 2, is shown in Table 3, where “x” denotes a do-not-care value. Please note that attribute c in Table 3 is removed because that column consists of do-not-care values only. Therefore, there are only three features selected to represent the original decision table without losing any essential information.

Table 3

The decision table after attribute reduction.

U	a	b	d	e
1	1	0	x	→ 1
2	1	0	x	→ 1
3	0	x	x	→ 0
4	x	1	1	→ 0
5	x	x	2	→ 2
6	x	x	2	→ 2
7	x	x	2	→ 2

2.4. Knowledge Reduction Process

When a training data set is given, continuous data can be represented with categorical values if separation boundaries in every input dimension are determined. After removing all dispensable attributes and merging all duplicates, a simplified decision rule base is obtained. This knowledge reduction process is actually the process of finding a reduct of the knowledge representation system constructed from the given data set. The final decision table after knowledge reduction (based on Table 2) is shown in Table 4.

Table 4

The final decision table after knowledge reduction.

U	a	b	d	e
1	1	0	x	→ 1
2	0	x	x	→ 0
3	x	1	1	→ 0
4	x	x	2	→ 2

3. Genetic Algorithm for Optimal Separation

Genetic Algorithm (GA) [14] is in the field of evolutionary algorithms. The idea of GA is inspired by the survival of the fittest theory proposed by Charles Darwin. GAs are searching algorithms based on the mechanics of natural selection and natural genetics [12]. GA normally starts with a randomly initialized population, which consists of artificial creatures denoted as chromosomes. Based on their fitness values, some of them are selected in pairs as parents and granted the opportunities to produce offspring by means of crossover operators. Subsequently, some survived chromosomes are randomly selected for mutation, which means their genes are to be varied. The chromosomes in the next generation are expected to perform better and the process goes on iteratively until any termination criterion is met. Although randomized, GA is not a random walk. It efficiently exploits historical information to speculate on new search points with expected improvements [12].

Because rough set theory only applies to categorical values, discretization of the training data set is required. GA is employed by our proposed model to search for optimal or satisfactory sub-optimal separation boundaries in every input di-

mension. Different strategies that can be applied to each step of GA are not covered in this paper. However, all strategies applied to our proposed model are introduced with details in the following section.

4. Proposed Clustering Technique

Genetic Algorithm based Rough Set Clustering (GARSC) is a clustering technique, which integrates genetic algorithm and rough set theory together. Genetic algorithm is applied to determine optimal or at least satisfactory suboptimal solutions. Rough set theory is incorporated to alleviate the curse of dimensionality problem [8], which leads to unnecessarily large network sizes of many established inference systems. By applying rough set approximations, the original knowledge base is greatly reduced without losing essential information. This wonderful characteristic of rough set theory is extremely helpful to improve the interpretability [27] of an existing inference rule base, i.e. reduce the number of features used for reasoning, reduce the number of rules in the rule base, and reduce the number of arguments stated in each inference rule. Therefore, the overall proposed system achieves a high level of interpretability without sacrificing accuracy. The overall GARSC process is illustrated in Fig. 1.

Fig. 1. Flowchart of genetic algorithm based rough set clustering process.

Please note that up to now the inference rules are crisp decision rules and we need to transform

them into fuzzy rules by generating Gaussian type fuzzy membership functions based on the clustering results and assigning corresponding linguistic terms. Subsequently, the transformed fuzzy rules are used to evaluate the performance of the current solution. This knowledge transfer concept is illustrated in Fig. 2.

Fig. 2. Illustration of knowledge transfer from crisp rules to fuzzy ones.

The reason why Gaussian type fuzzy membership functions are utilized in our proposed system rather than other commonly adopted ones (such as triangular and trapezoidal) is that Gaussian function is continuous, which better represents the density of the cluster (see Fig. 5), and it only has two parameters (mean and standard deviation), which are easier to derive. Assume in dimension x , we define $n - 1$ number of separation boundaries, then x is discretized into n regions $\{x_1, \dots, x_i, \dots, x_n\}$. Therefore, determining Gaussian type fuzzy mem-

bership function $f_{G_i}(x) = e^{-\frac{\|x - c_i\|^2}{2\sigma_i^2}}$ is actually computing the center c_i and the standard deviation σ_i of all the data points in the i th region x_i .

This particular process of knowledge transfer from crisp to fuzzy cannot be omitted because the crispness of separation adopted in rough set theory does not tolerate overlapping. Fuzzy membership functions are employed to represent the derived clusters to deal with inexact information and unforeseen circumstances. This kind of knowledge transfer has a great advantage because it naturally prevents the fuzzy membership functions from overlapping or separating too much with adjacent ones, which is another important aspect of interpretability in fuzzy modeling. Furthermore, because clustering is performed in each individual feature, no transformation or normalization is required and more importantly, semantic meanings of the assigned linguistic labels are preserved.

4.1. Constraints on Data Discretization

Before the detailed introduction of GARSC, there are certain constraints on data discretiza-

tion to be defined. One is the maximum number of separation boundaries allowed in each dimension. This constraint straightforwardly determines the maximum number of fuzzy membership functions allowed in each dimension. However, the actual number of fuzzy membership functions formulated in each dimension is also determined by the knowledge reduction process. The smallest number of separation boundaries actually in use is zero, which means that particular feature is not considered in the simplified knowledge base. This constraint should not be set to a large value because employing a huge number of fuzzy membership functions degrades interpretability.

The other constraint is on the minimum distance between any pair of adjacent separation boundaries in the same dimension, which is termed *mindis*. This constraint is imposed to make sure that the actual constructed fuzzy membership functions have high-level generalization such that any adjacent pair of them should not be merged into one. The minimum distance constraint *mindis* is defined in Eq. (4). The *max* function in the denominator defines the level of generalization.

$$mindis_i = \frac{ub_i - lb_i}{\max(nop_i, M)}, \quad (4)$$

where i denotes the i th dimension; ub_i and lb_i denote the upper and lower boundaries of values in the i th dimension respectively; nop_i denotes the total number of possible separation boundaries in the i th dimension, which if not specifically stated otherwise, is assigned to the total number of different values in the i th dimension where corresponding values in the decision attribute are changed across; and M denotes the user specified minimum number of separation boundaries in every dimension, which has a default value of 10.

4.2. Knowledge Reduction and Removal of Inconsistent Rules

It has been discussed earlier in this paper that after the removal of all dispensable attributes, the finalized decision table is independent, which is actually one of the many possible reducts of the originally constructed decision table. Therefore, attribute reduction is performed during the process of constructing the independent decision table with minimum cardinality. If an attribute is dispensable in all rules, then that attribute can

be removed from the given data set. This process of removing attributes that do not contribute to the essential knowledge base is referred as feature selection.

Decision rule reduction is conceptually similar to attribute reduction. Other than merging each set of duplicate rules into one single rule, a decision rule in the rule set is dispensable if and only if the performance of the rule base does not decrease with the removal of that rule. This is also referred as the pruning process of decision rules. Moreover, the removal of inconsistent rules is necessary to maintain the integrity of the knowledge base.

Inconsistent rules, which are the rules with the same conditional attributes but different decision attribute, do often exist in real world applications. Only one rule from each inconsistent rule set should be preserved to remove ambiguities. To determine which rule(s) should be preserved in the simplified decision table, we propose Eq. (5) (in rough set theory terms) to compute the confidence of the k th rule. The *min* function is applied to penalize information incompatibility.

$$conf(k) = \min \left(\frac{card(U_i(k_i) \cap d_k)}{card(U_i(k_i))} \right), \forall i \in C, \quad (5)$$

where *card* denotes the function to compute cardinality; U_i denotes the function to generate a union of decision attributes of every rule in the decision table that shares the same value on the i th attribute; k_i denotes the value of the i th attribute of the k th rule; and d_k denotes the decision attribute of the k th rule.

Based on the confidence evaluation function, the criteria to remove inconsistent rules from the decision table are defined as follows. The rule with the maximum confidence value will be preserved in the decision table while all the other rules from the same inconsistent rule set will be removed. If multiple rules are tied at the maximum confidence value in the same inconsistent rule set, the rule covers the most number of data elements is preserved. If multiple rules still tie in this situation, a random rule will be selected as the winner with equal probability.

4.3. Proposed Genetic Algorithm Strategies

As mentioned earlier in this paper that all strategies of genetic algorithm applied to our proposed clustering method is introduced with details in this subsection.

4.3.1. Commonly Adopted Strategies

In genetic algorithm, the population size defines the total number of chromosomes existing in each generation. Increasing the population size is equivalent to increasing the number of possible solutions. Therefore, more candidates will be examined, and a wider range of area in the universe of discourse will be explored.

In GARSC, real number coding strategy is employed to construct chromosomes. Each gene used in the chromosome represents a separation boundary in its respective dimension. Because GARSC technique only constrains the maximum number of partitions allowable in each dimension, hence, the actual number of partitions varies, i.e. chromosomes in GARSC have different lengths.

In GARSC, elitism replacement strategy is applied to exploit solutions that have been evaluated before. The elitism ratio μ , which is in the $[0, 1)$ interval, defines what percentage of highly fit individuals in the current generation $P(t)$ will be directly brought into the next generation $P(t + 1)$. Normally, μ is assigned to a relatively small number to prevent the domination of species on local optimums.

The overall clustering process stops when genetic algorithm reaches the predefined number of generations, which should sufficiently ensure the convergence of the genetic algorithm.

4.3.2. Fitness Evaluation Function

Fitness function evaluates the quality of each chromosome. It is probably the most important component in genetic algorithm, because it directly defines the performance of each chromosome and intimately characterizes the ideal solution that the user attempts to search for. Based on the nature of the bank failure prediction problem, we propose our fitness function f in Eq. (6). We use capital letters to represent constants and small letters to represent variables. Term-1 and term-3 in Eq. (6) represent the accuracy of the model and term-2 represents the interpretability because it is the score on the number of derived fuzzy rules. This fitness function is to be minimized by the genetic algorithm.

$$f(x) = \underbrace{(1-a)\frac{NOD}{NOF}}_1 + \underbrace{\frac{nor}{NOD}}_2 + \underbrace{\frac{mse}{NOF}}_3, \quad (6)$$

where x denotes the chromosome to be evaluated; a denotes the accuracy of applying the data set to

the model derived using x ; NOD is the total number of data elements in the data set; NOF denotes the total number of input features in the data set; nor denotes the number of rules actually used to construct the model; and mse denotes the mean squared error computation, which is defined in Eq. (7).

$$mse = \frac{1}{NOD} \sum_{i=1}^{NOD} (y_i - \hat{y}_i)^2, \quad (7)$$

where y_i denotes the predicted value and $\hat{y}_i$ denotes the expected value.

We propose three terms that evaluate two different types of performance in the fitness function. The employment of this type of fitness function is one way to implement a multi-objective genetic algorithm [7]. The three terms in Eq. (6) are placed in descending order of importance and the proof is given as follows.

Term-1 of Eq. (6) can be expanded into Eq. (8).

$$(1-a)\frac{NOD}{NOF} = (1-\frac{noc}{NOD})\frac{NOD}{NOF} = \frac{NOD - noc}{NOF}, \quad (8)$$

where noc denotes the number of correctly classified data elements.

If the number of correctly classified data elements decreases by one, i.e. $noc \rightarrow noc - 1$, then we can rewrite Eq. (8) into Eq. (9).

$$\frac{NOD - (noc-1)}{NOF} = \frac{NOD - noc}{NOF} + \frac{1}{NOF}. \quad (9)$$

Comparing Eq. (8) to Eq. (9), if all the other terms in Eq. (6) remain unchanged, then the fitness value is increased by $\frac{1}{NOF}$, which denotes a lesser fit of the chromosome.

Similarly, we can find that if one more rule is employed in the inference knowledge base, i.e. $nor \rightarrow nor + 1$, then the fitness value is increased by $\frac{1}{NOD}$. The number of dimensions in the given data set is normally smaller than the number of data elements (except for gene data), i.e. $NOF < NOD \Rightarrow \frac{1}{NOF} > \frac{1}{NOD}$, which implies that the effect of the slightest change in term-1 of Eq. (6) is greater than that of term-2.

By substituting Eq. (7), we can rewrite term-3 of Eq. (6) into Eq. (10).

$$\frac{mse}{NOF} = \sum \frac{(y_i - \hat{y}_i)^2}{NOF \cdot NOD}. \quad (10)$$

Because we are evaluating the effect of the slightest change of Eq. (10) while the other two

terms of Eq. (6) remain unchanged, we can say that the classification accuracy and interpretability of the constructed model do not vary, but there is a small increase in the mean squared error. The amount of the slightest change is assumed to be smaller than one (if only one predicted value among all the data elements is computed differently without changing the classification accuracy). It is also obvious that $\frac{1}{NOF \cdot NOD} > \frac{1}{NOF \cdot NOD}$. Therefore, term-2 of Eq. (6) takes on a greater effect than that of term-3 in a rather stable solution where predictions do not vary drastically.

Thus, the complete proof has been given that the amount of effect decreases along the three terms of Eq. (6). Term-1 ensures the correctness of the constructed model and term-2 encourages the constructed model to achieve the same level of accuracy by using lesser number of rules. A higher level of interpretability with the same level of accuracy constructs better model because a higher level of generalization is achieved to prevent the model from over-fitting. Term-3 refines the constructed model to achieve a better fitting on the given data set as long as accuracy and interpretability are kept unchanged. Please note that although evaluations on feature selection and attribute reduction are not introduced in the fitness function (the number of features selected and the number of arguments employed in each inference rule are less concerned by the financial experts because they run regressions on all available features), GARSC still performs those two processes to optimize the inference rule base for a higher level of interpretability.

4.3.3. Tournament Selection Strategy

To produce new chromosomes, parents with high fitness values are selected from the current generation to produce offspring in the next generation. Tournament selection strategy [24] is employed in GARSC because we can easily control the selection stress by adjusting only the tournament size m and the selection probability p .

For each chromosome to be selected for crossover, m candidates are randomly selected from the current generation for consideration and they are sorted in descending order based on their fitness values. The selection process starts with the first candidate and the probability of selecting the n th candidate $p(n)$ is defined by Eq. (11).

$$p(n) = p(1 - p)^{n-1}, \quad 0.5 < p \leq 1. \quad (11)$$

If m is large, it is more stressful for lesser fit candidates to be selected. On the other hand, a small tournament size increases the probability of lesser fit candidates being selected as they are competing with a lesser number of the others. In general, a large value of m is used in simple and unimodal application domains to accelerate the converging process and a small value of m is used in complex and multimodal application domains to better explore the universe of discourse [24].

In the early generations of GA, p should be set to a small value to give more chances to those lesser fit candidates to get selected. In this way, the search is prevented from premature convergence because more possible candidates are considered even if they have smaller fitness values. However, in the late generations, p should set to a large value, because only those highly fit candidates are expected to lead the selection towards the best solution in more in-depth exploitation. Tournament selection probability p in GARSC is defined by Eq. (12).

$$p = 0.5 \left(1 + \frac{cgi}{NOG} \right), \quad (12)$$

where cgi denotes the current generation index and NOG denotes the maximum number of generations of GA. Because cgi is in the $[1, NOG]$ interval, p is equally distributed in the $[0.5 + \frac{0.5}{NOG}, 1]$ interval, which exactly fulfills the constraint defined in Eq. (11).

4.3.4. Modified Uniform Crossover Operator

When sufficient number of parents has been selected, the crossover operator is applied to each pair of parents to produce offspring. The probability for a pair of selected parents eventually mate is defined as the crossover rate. However, because elitism replacement strategy is applied in GARSC, the crossover rate is set to one, i.e. every pair of parents is crossed over to produce offspring. Because chromosomes in GARSC consist of real number coded genes to represent sets of separation boundaries, which are different in length, there is no simple crossover operator feasible to perform the proposed information exchange between the selected parents.

A modified uniform crossover operator is proposed to deal with chromosomes of different lengths. Similar to conventional uniform crossover

operator, a binary string is randomly created to control in each position, from which parent the child should inherit the gene. The length of the control string is assigned to the number of conditional attributes in the given data set. Therefore, there will be no risk taken on the misunderstanding of the dimensionality during the creation process of offspring. The modified uniform crossover operator is illustrated in Fig. 3.

Fig. 3. Illustration of the proposed modified uniform crossover operator.

Because feature selection is applied in GARSC, many chromosomes have no gene in certain input dimension as indicated with the empty brackets “[]” in Fig. 3. Therefore, when a pair of chromosomes is uniformly crossed over, there is a chance that one of the offspring has no gene in every input dimension. In such case, the “empty” chromosome will be replaced with a randomly constructed non-empty one.

4.3.5. Modified Mutation Operators

Unlike conventional mutation operators, which simply vary the values of the selected genes, a set of three different mutation operators is proposed in GARSC. Whenever a gene is selected to be mutated, one of the following three operators are performed with equal probability:

1. Add one separation boundary to the gene if possible;
2. Remove one separation boundary from the gene if possible;
3. Vary the value of a randomly selected separation boundary in the gene if possible.

Similar to tournament selection probability p , the mutation rate $mrate$, which defines the probability for each gene to be mutated, should increase from a small value in the early generations to a large value in the late generations. Based on this policy, $mrate$ is defined in Eq. (13).

$$mrate = \frac{1}{NOF} + \frac{(NOF - 1) \cdot cgi}{NOF \cdot NOG}. \quad (13)$$

5. Proposed System Architecture

The architecture of Genetic Algorithm and Rough Set Incorporated Neural Fuzzy Inference System (GARSINFIS) is illustrated in Fig. 4. GARSINFIS is a six-layered, feed-forward, and partially connected architecture. In each layer, neurons are not connected to each other, but only connected to neurons in the adjacent layer(s).

Fig. 4. The network architecture of our proposed GARSINFIS model.

Each layer of GARSINFIS performs respective fuzzy or non-fuzzy operations, which are described as follows. Input layer receives the input data vector and translates it into fuzzy singletons in each dimension. Because feature selection is applied, not all given linguistic variables are presented to the connected neurons in the next layer. Condition layer stores respective fuzzy membership functions with respect to each selected linguistic variable and subsequently presents the activation values to the connected neurons in the next layer. Rule-base layer performs fuzzy reasoning and subsequently presents the activation value of the corresponding fuzzy rule to all the neurons in the next layer. Normalization layer scales the activation values of all fuzzy rules to the same level of reference and subsequently presents the normalized values to the connected neuron in the next layer. Consequence layer computes the prediction of each rule and subsequently present it to the only neuron in the next layer. Output layer consists of only one neuron, which aggregates all the inputs received and presents the result as the overall inference output.

In Fig. 4, the rectangular boxes used to represent neurons in condition layer and consequence layer denote the antecedent and consequent parts of the employed fuzzy rules respectively. Because this paper focuses on bank failure prediction, which has a relatively high accuracy rate, GARSINFIS em-

employs zero-order TSK type of fuzzy rules [37,36] to achieve the maximum level of interpretability by minimizing the consequent part of each fuzzy rule. Zero-order TSK type of fuzzy rule is functionally equivalent to Mamdani type of fuzzy rules [23] if fuzzy singletons are used for simplicity [43]. Both types of simple fuzzy rules can be defined by Eq. (14).

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{(1,i)}, \dots, x_n \text{ is } A_{(n,i)}, \dots, x_N \text{ is } A_{(N,i)}; \\ &\text{THEN } y_i \text{ is } B_i, \end{aligned} \quad (14)$$

where x_n denotes the n th input linguistic variable; $A_{(n,i)}$ denotes the fuzzy label defined in the i th rule on the n th input linguistic variable; N denotes the total number of selected linguistic variables; y_i denotes the output of the i th rule; and B_i denotes the singleton defined in the i th rule.

Because zero-order TSK type of fuzzy rule is employed for a higher level of interpretability, the general function of mapping input $X = (x_1, \dots, x_i, \dots, x_I)$ into output y using M number of rules can be defined by Eq. (15).

$$y = \frac{\sum_{m=1}^M \alpha_m \cdot B_m}{\sum_{m=1}^M \alpha_m}, \quad (15)$$

where α_m is the firing strength of the m th rule.

When the implication operator is selected as the *min* operator, then α_m can be defined by Eq. (16).

$$\alpha_m = \min(\mu_{A_{m1}}(x_1), \dots, \mu_{A_{mi}}(x_i), \dots, \mu_{A_{mI}}(x_I)), \quad (16)$$

where $\mu_{A_{mi}}(x_i)$ denotes the membership value of x_i according to the membership function A_{mi} of the m th rule in the i th dimension.

Because GARSINFIS directly employs the fuzzy rules derived by Genetic Algorithm based Rough Set Clustering (GARSC) technique, GARSINFIS self-organizes its network structure. Moreover, because GARSC systematically derives simplified fuzzy inference rules, the network size of GARSINFIS is smaller than that of the other similar models such as ANFIS [15]. Unlike ANFIS, GARSINFIS is not fully connected. Because feature selection is performed, not all input dimensions in the given data set are utilized. Because the maximum number of fuzzy membership functions in each dimension is constrained by GARSC, the number of neurons employed to represent the derived membership functions is also constrained. Because attribute selection in the antecedent part of fuzzy rules is performed, hence the derived fuzzy rules

may not necessarily employ all the fuzzy membership functions in every selected dimension. Because rule pruning is performed during the clustering process, the number of neurons used to represent the fuzzy rule set is also minimized. The compact GARSINFIS architecture is now ready for performance evaluations to predict bank failures.

6. Empirical Studies on Bank Failure Prediction

6.1. Bank Failure Prediction Data Set

The bank failure prediction data set used in this paper is extracted from the financial statements of 3635 banks in United States, which are publicly available [10]. Based on the annual financial statements, nine financial covariates are extracted to categorize whether a bank has failed or survived. The selected features are inspired by the works of financial experts [19,5]. All the nine selected financial covariates are listed in Table 5 with their expected impacts on bank failures. Most financial experts in banking finance run regressions on these covariates. However, it is stated in [41] that soft computing models overcome the deficiencies of traditional statistical models whose results do not possess semantic meanings. The data set used in this paper has been analyzed by different NFISs in the literature [41,39,40,25,30] for different purposes with different configurations. In this paper, we focus on the balance between accuracy and interpretability (not studied in the previous works) of the constructed model and the effectiveness to generate early warnings for potentially failing banks.

6.2. Design of Experiments

The objective of the experiments is to evaluate the capability of GARSINFIS to systematically derive a small number of rules from the training data set and subsequently perform accurate inferences on a large number of unforeseen instances. Sensitivity and specificity, which are defined in Eqs. (17) and (18) respectively, are reported besides accuracy on the testing data set. In bank failure prediction, sensitivity is critical because successful early warnings of poor performing banks buy more time to prevent unnecessary consequences and speci-

Table 5
Input features of bank failure prediction data set.

Category	Financial covariate	Expected impact on bank failure
Capital Adequacy	CAPADE: $\frac{\text{average total equity capital}}{\text{average total assets}}$	if ratio is high, then capital is relatively large to absorb losses, i.e. small probability to fail
Asset (Loan) Quality	OLAQLY: $\frac{\text{average (accumulated) loan loss allowance}}{\text{average total loans \& leases}}$	if ratio is small, then loan quality is relatively good, i.e. small probability to fail
	PROBLO: $\frac{\text{average (accumulated) loans 90+ days late}}{\text{average total loans \& leases}}$	
	PLAQLY: $\frac{(\text{annual}) \text{ loan loss provision}}{\text{average total loans \& leases}}$	
Management	NIEOIN: $\frac{\text{non interest expense}}{\text{operating income}}$	if ratio is small, then bank operates relatively healthy, i.e. small probability to fail
Earning	NINMAR: $\frac{\text{total interest income} - \text{interest expense}}{\text{average total asset}}$	if ratio is high, then bank is relatively profitable, i.e. small probability to fail
	ROE: $\frac{\text{net income (after tax)} + \text{applicable income tax}}{\text{average total equity capital}}$	
Liquidity	LIQUID: $\frac{(\text{all taken the average values}) \text{ cash} + \text{federal funds sold}}{\text{total deposit} + \text{fed funds bought} + \text{liability}}$	if ratio is small, then utilization of resources is relatively efficient, i.e. small probability to fail
Miscellaneous	GROWLA: $\frac{(\text{total loans \& leases})_t - (\text{total loans \& leases})_{t-1}}{(\text{total loans \& leases})_{t-1}}$	if ratio is high, then loan growth is relatively high, i.e. small probability to fail

ficity is critical because false alarms to good performing banks create unnecessary panics.

$$\text{Sensitivity} = \frac{\text{No. of failed banks correctly predicted}}{\text{Total no. of failed banks}}, \quad (17)$$

$$\text{Specificity} = \frac{\text{No. of survived banks correctly predicted}}{\text{Total no. of survived banks}}. \quad (18)$$

There are three scenarios studied in this bank failure prediction application. Numbers of all available data samples in each scenario are listed in Table 6. In scenario-1, each model actually identifies whether a bank is failed. While in scenario-2 and scenario-3, each model predicts whether a bank will fail in one and two years respectively.

Table 6

Data distributions in the bank failure prediction data set.

Scenario	Survived (negative)	Failed (positive)
1: Last year available	2555 (82.34%)	548 (17.66%)
2: One year prior	2572 (84.44%)	474 (15.56%)
3: Two years prior	2585 (87.84%)	358 (12.16%)

To demonstrate that GARSINFIS utilizes only the most essential knowledge to perform accurate inferences, in each scenario and in each experiment, 100 instances from each class are randomly selected for training and all the other instances consist of the testing data set. In such a way, the ability to derive a correct inference knowledge base from a relatively small training data set is ultimately assessed. Please note that the fuzzy rules derived by GARSC are of zero-order TSK type without adaptively tuning the parameters of Gaus-

sian membership functions. Instead, only numerous sets of membership functions and fuzzy rules are iteratively evaluated and the set with the most promising performance is used to construct the system when GA terminates.

For all scenarios and all experiments (10 experiments are conducted in each scenario to remove randomness and to assess the stability), the population size of GA is set to 300 and the number of generations is set to 20. Elitism ratio is set to 0.1, which means in every generation, 30 chromosomes with the highest fitness values are directly brought into the next generation. Tournament size is set to 2 to ease the stress during selection. The maximum number of separation boundaries allowed in any input dimension is set to 2. Therefore, the maximum number of fuzzy membership functions in any input dimension is 3. This guarantees a high level of interpretability because there are at most three linguistic labels, small, medium, and large, assigned in any input dimension. The number of control parameters or constraints required by GARSINFIS to automatically obtain optimal or satisfactory suboptimal solutions is limited. GARSINFIS systematically construct simple yet accurate inference rule base without human intervention and expert guidance.

6.3. Benchmarking Models

When presenting the experimental results, other than accuracies (including sensitivity and speci-

ficity) on the testing data sets and training times spent, other measures such as the number of employed features, rules, and arguments are also recorded for comparison purposes. The number of features utilized by GARSINFIS is the actual number of features selected from the given data set to perform inferences. The number of rules employed by GARSINFIS is the actual number of rules in the simplified rule set after the removals of redundancies and inconsistencies. The number of arguments is defined by the total number of arguments in the antecedent part of all the employed rules after attribute reduction.

When benchmarking the experimental results, some well-established models such as C4.5 decision tree [31], Naive Bayes classifier [32], Multi-Layer Perceptron (MLP) network [13], Radial Basis Function (RBF) network [13], Adaptive Network-based Fuzzy Inference System (ANFIS) [15] and Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) [17] are used for comparisons. All these models are applied to the same pairs of training and testing data sets, which are applied to GARSINFIS. In this way, their performances are fairly compared on the same references.

In terms of comparisons on the selected features, only GARSINFIS and C4.5 decision tree are discussed, because all the other models utilize all given features without preferences. Even if the associated weights of certain input dimensions can be significantly small in some models, it is still not considered as feature selection.

In terms of the size of the employed rule set, the actual number of employed rules is recorded for GARSINFIS, ANFIS, and DENFIS. For C4.5 decision tree, the number of tree leaves is recorded because it is equivalent to the number of crisp decision rules. For Naive Bayes classifier, the number of rules is not applicable. For MLP network, the number of hidden neurons, which is systematically determined by the total number of input dimensions and the number of output classes, is recorded. For RBF network, the number of radial basis neurons, which is pre-determined by the number of clusters in every output class, is recorded.

In the comparison tables, the winner of any particular column is highlighted in bold. The winner is either the largest accuracy value or the least amount of information employed. Because multiple experiments are conducted on each model for each scenario, stability (standard deviations) of every

model is also recorded and compared. Winners in stability are also highlighted in bold.

In this paper, ANFIS utilizes clustering results derived by the employed Fuzzy C-Means (FCM) clustering technique [4]. In this way, ANFIS is more comparable to GARSINFIS because the number of rules employed by ANFIS is greatly reduced. However, extra efforts are taken to carefully select the optimal number of pre-defined clusters through trial-and-error. This is a great disadvantage of ANFIS, because its network structure is not self-organized, but has to be pre-defined by employing additional knowledge. The Evolving Clustering Method (ECM) [35] employed by DENFIS constructs clusters based on the distances between data samples in the high-dimensional space. ECM often derives a large number of clusters and as a result DENFIS employs a large number of rules. Generally speaking, DENFIS often fails to alleviate the curse of dimensionality problem [8].

6.4. Scenario-1: Last Year Available

Table 7 reports the performance of GARSINFIS on bank failure prediction scenario-1 data set. On average, GARSINFIS utilizes 4 input features and employs 4.9 rules (3 negative ones and 1.9 positive ones) to achieve 96.49% accuracy on the testing data sets. Each rule employs averagely $10/4.9 = 2.04$ arguments in the antecedent part. The average training time is less than 11 minutes. Experiment-5 utilizes the least number of features and the least number of rules to achieve the highest accuracy. The derived fuzzy membership functions in experiment-5 are illustrated in Fig. 5 and the employed fuzzy rules in the same experiment are shown in Table 8.

Experiment-5 utilizes three input features and two rules for each class to achieve the accuracy of 97.21% on the testing data set. It is more encouraging to learn that the rules shown in Table 8 are consistent with the expert knowledge presented in Table 5 that if CAPADE and GROWLA are large and PLAQLY is small, then the probability of bank failure is small. Therefore, Fig. 5 and Table 8 present an excellent example of how GARSINFIS constructs a highly comprehensive yet accurate fuzzy inference rule base.

Table 9 reports the performance comparisons of GARSINFIS against other benchmarking models on bank failure scenario-1 data set. In terms of ac-

Table 7

Summary of applying GARSINFIS on bank failure prediction scenario-1 data set.

Experiment Index	Accuracy (%)	Sensitivity (%)	Specificity (%)	nof	nor	noa	Time (seconds)
1	96.76	96.65	96.78	4	4	9	618.80
2	96.42	95.09	96.66	4	5	9	700.74
3	95.73	94.87	95.89	4	6	13	664.22
4	96.45	96.21	96.50	5	7	15	709.54
5	97.21	95.76	97.47	3	4	8	663.12
6	96.80	92.63	97.56	4	4	7	583.02
7	96.66	91.96	97.52	4	4	7	545.96
8	96.18	95.31	96.33	4	4	8	663.97
9	96.56	93.30	97.15	4	5	11	705.57
10	96.14	95.54	96.25	4	6	13	598.07
mean	96.49	94.73	96.81	4	4.9	10.0	645.30
std	0.41	1.57	0.59	0.47	1.10	2.83	56.31

nof: number of features; nor: number of rules; noa: number of (total) arguments

Table 8

One set of the derived fuzzy rules of bank failure prediction scenario-1 data set.

IF PLAQLY is small and GROWLA is large	THEN the bank is survived
IF CAPADE is large and PLAQLY is medium	THEN the bank is survived
IF CAPADE is small and GROWLA is small	THEN the bank is failed
IF CAPADE is medium and PLAQLY is large	THEN the bank is failed

Table 9

Benchmarks on bank failure prediction scenario-1 data set.

Model		Accuracy (%)	Sensitivity (%)	Specificity (%)	Features	Rules
GARSINFIS	mean	96.49	94.73	96.81	4	4.9
	std	0.41	1.57	0.59	0.47	1.10
C4.5	mean	95.76	95.42	95.82	2.8	5.2
	std	0.68	1.85	0.95	0.63	1.40
Naive Bayes	mean	96.26	96.25	96.26	9	N.A.
	std	0.42	0.87	0.63		
MLP	mean	97.31	95.31	97.67	9	7 (nodes)
	std	0.60	1.84	0.68		
RBF	mean	96.06	95.27	96.20	9	2 (nodes)
	std	0.47	1.07	0.71		
ANFIS	mean	97.37	91.25	98.48	9	4.0
	std	0.55	3.56	0.38		
DENFIS	mean	96.13	95.36	96.27	9	28.2
	std	1.16	1.12	1.33		

curacy, ANFIS performs the best, but it has the lowest sensitivity. In bank failure prediction application, sensitivity is more important than specificity because the consequence of no early warnings is more serious than that of false alarms. Therefore, although ANFIS achieves the highest accuracy on the testing data set, it is not the most reliable model in this particular application. MLP also performs better than GARSINFIS. However, trained MLP network does not possess semantic meanings of the inference process. In terms of interpretability, C4.5 decision tree utilizes the least

number of input features. However, its accuracy is lower than that of GARSINFIS and it employs more number of rules than GARSINFIS does. ANFIS employs the least number of rules. However, it utilizes all the input features and requires extra efforts to determine the number of clusters for the employed FCM clustering technique through trial-and-error. Based on the benchmarks on bank failure prediction scenario-1 data set, GARSINFIS does not lead in any performance measure yet. We can only state that the performance of GARSINFIS is satisfactory and it needs to be further eval-

Fig. 5. One set of the derived fuzzy membership functions of bank failure prediction scenario-1 data set.

uated on more challenging scenarios.

6.5. Scenario-2: One Year Prior

Table 10 reports the performance of GARSINFIS on bank failure prediction scenario-2 data set. Comparing Table 10 to Table 7, performance of GARSINFIS decreases because accuracy decreases with more number of features utilized, more number of rules employed, and more computational time spent. However, this is within expectation because prediction of bank failures is naturally more difficult based on one year prior data. On average, GARSINFIS utilizes 5.1 input features and employs 7.9 rules (5 negative ones and 2.9 positive ones) to achieve 92.39% accuracy on the testing data sets. Each rule employs averagely $18.4/7.9 = 2.33$ arguments in the antecedent part. The average training time is less than 19 minutes.

Table 11 reports the performance comparisons of GARSINFIS against other benchmarking models on bank failure scenario-2 data set. In terms of accuracy on the testing data set, ANFIS and MLP again perform better than GARSINFIS, but their sensitivities are lower than that of GARSINFIS, which achieves the highest value among all models. Although Naive Bayes classifier also performs better than GARSINFIS on accuracy, Naive Bayes classifier is not reliable to generate early warnings to potentially failing banks because its sensitivity is the lowest among all models. In terms of interpretability, C4.5 decision tree utilizes a slightly lesser number of input features and the same number of rules when comparing to GARSINFIS. How-

ever, C4.5 decision tree performs worse on every accuracy measures. Again, ANFIS employs the least number of rules. However, it utilizes all the input features, requires extra efforts through trial-and-error and is not reliable because its sensitivity is low. By employing a compact inference knowledge base, GARSINFIS achieves the highest sensitivity and satisfactory specificity on bank failure prediction scenario-3 data set.

6.6. Scenario-3: Two Years Prior

Table 12 reports the performance of GARSINFIS on bank failure prediction scenario-3 data set. Comparing Table 12 to Table 10, performance of GARSINFIS further decreases because prediction of bank failures is naturally even more difficult based on two years prior data. On average, GARSINFIS utilizes 6.1 input features and employs 10.9 rules (6.8 negative ones and 4.1 positive ones) to achieve 89.91% accuracy on the testing data sets. Each rule employs averagely $28.9/10.9 = 2.65$ arguments in the antecedent part. The average training time is less than 34 minutes.

Table 13 reports the performance comparisons of GARSINFIS against other benchmarking models on bank failure scenario-3 data set. The performance comparisons of different models are similar to that of Table 11, except that DENFIS performs better than GARSINFIS in terms of accuracy and specificity. However, DENFIS is less reliable than GARSINFIS because DENFIS has lower sensitivity. Moreover, DENFIS utilizes all input features and employs $(26.6 - 10.9)/10.9 = 144\%$ more times of rules. In terms of interpretability, although less amount of knowledge is utilized, C4.5 decision tree again performs worse on every accuracy measures comparing to GARSINFIS. By employing a compact inference knowledge base, GARSINFIS achieves the highest sensitivity and satisfactory specificity on bank failure prediction scenario-3 data set.

6.7. Summary on Empirical Results

In the bank failure prediction application, although only number of rules is defined in the fitness function besides accuracy terms, GARSINFIS still utilizes a relatively small number of features and a small number of arguments (no more than 3 on average) in the antecedent part of each rule.

Table 10

Summary of applying GARSINFIS on bank failure prediction scenario-2 data set.

Experiment Index	Accuracy (%)	Sensitivity (%)	Specificity (%)	nof	nor	noa	Time (seconds)
1	92.90	91.98	93.04	5	6	15	943.37
2	92.59	90.37	92.92	4	9	20	1145.53
3	92.27	93.32	92.11	6	11	26	1292.22
4	91.53	90.11	91.75	5	6	13	1038.10
5	94.59	92.25	94.94	6	8	18	1166.04
6	92.13	91.18	92.27	5	9	21	1308.21
7	92.45	93.85	92.23	5	5	12	979.59
8	91.74	89.84	92.03	4	7	14	1010.98
9	91.53	92.51	91.38	5	8	20	1195.62
10	92.13	89.04	92.60	6	10	25	1290.32
mean	92.39	91.44	92.53	5.1	7.9	18.4	1137.00
std	0.89	1.59	0.99	0.74	1.91	4.88	137.11

nof: number of features; nor: number of rules; noa: number of (total) arguments

Table 11

Benchmarks on bank failure prediction scenario-2 data set.

Model		Accuracy (%)	Sensitivity (%)	Specificity (%)	Features	Rules
GARSINFIS	mean	92.39	91.44	92.53	5.1	7.9
	std	0.89	1.59	0.99	0.74	1.91
C4.5	mean	90.59	91.04	90.52	4.7	7.9
	std	1.70	1.75	2.10	1.34	1.97
Naive Bayes	mean	92.70	85.83	93.74	9	N.A.
	std	1.10	3.29	1.71		
MLP	mean	93.15	91.42	93.41	9	7 (nodes)
	std	1.41	2.44	1.83		
RBF	mean	92.05	90.64	92.26	9	6 (nodes)
	std	1.24	1.57	1.55		
ANFIS	mean	94.36	88.72	95.21	9	6.3
	std	1.80	2.54	1.98		1.49
DENFIS	mean	92.14	89.68	92.51	9	29.8
	std	1.25	2.07	1.52		5.83

Table 12

Summary of applying GARSINFIS on bank failure prediction scenario-3 data set.

Experiment Index	Accuracy (%)	Sensitivity (%)	Specificity (%)	nof	nor	noa	Time (seconds)
1	91.36	91.09	91.39	7	13	33	2106.18
2	90.30	87.60	90.58	7	10	25	1819.11
3	89.17	88.76	89.22	6	8	22	2006.57
4	87.64	91.86	87.20	5	8	20	1750.96
5	91.62	88.37	91.95	7	15	39	1942.09
6	89.46	87.21	89.70	7	9	26	2118.63
7	91.32	89.15	91.55	6	13	37	2044.56
8	88.84	87.98	88.93	5	10	27	2175.83
9	91.18	86.43	91.67	5	11	29	1934.32
10	88.23	89.53	88.09	6	12	31	2033.17
mean	89.91	88.80	90.03	6.1	10.9	28.9	1993.14
std	1.44	1.69	1.65	0.88	2.33	6.17	133.89

nof: number of features; nor: number of rules; noa: number of (total) arguments

High level of interpretability is achieved. Although it employs a slightly larger amount of knowledge, GARSINFIS outperforms C4.5 decision tree in almost every accuracy measures (only lower sensi-

tivity in scenario-1). This implies that GARSINFIS performs better than C4.5 decision tree on the inference process of unforeseen data. Moreover, this is a possible piece of evidence that fuzzy

Table 13
Benchmarks on bank failure prediction scenario-3 data set.

Model		Accuracy (%)	Sensitivity (%)	Specificity (%)	Features	Rules
GARSINFIS	mean	89.91	88.80	90.03	6.1	10.9
	std	1.44	1.69	1.65	0.88	2.33
C4.5	mean	85.46	87.25	85.28	5.5	9.2
	std	2.69	3.26	3.21	1.35	2.20
Naive Bayes	mean	89.86	75.47	91.35	9	N.A.
	std	2.43	5.99	3.21		
MLP	mean	92.55	87.21	93.10	9	7 (nodes)
	std	1.48	2.65	1.75		
RBF	mean	88.67	86.74	88.87	9	6 (nodes)
	std	2.02	3.75	2.51		
ANFIS	mean	91.91	82.17	92.93	9	7.4 2.76
	std	1.91	3.42	2.28		
DENFIS	mean	91.26	87.75	91.62	9	26.6 2.91
	std	1.46	1.61	1.64		

Table 14
Summary of selected features on bank failure prediction data sets.

Feature	GARSINFIS			C4.5 Decision Tree		
	Scenario-1	Scenario-2	Scenario-3	Scenario-1	Scenario-2	Scenario-3
CAPADE	10	10	10	10	10	10
OLAQLY	4	2	3	0	3	1
PROBLO	3	4	7	1	5	6
PLAQLY	10	10	10	8	10	10
NIEOIN	4	7	7	2	3	4
NINMAR	0	1	3	0	4	3
ROE	1	6	9	4	7	10
LIQUID	2	9	10	2	3	8
GROWLA	6	2	2	1	2	3

rules employed by GARSINFIS perform better than crisp rules employed by C4.5 decision tree. Based on Fig. 5 and Table 8, we are proud to state that GARSINFIS automatically obtains highly interpretable fuzzy rules that match the expert knowledge. Furthermore, when comparing against other benchmarking models, GARSINFIS systematically constructs simple yet accurate fuzzy inference rules without human intervention and expert guidance and are highly reliable (high accuracy and sensitivity) and comprehensible (compact knowledge base) to human users.

Performances of all models decrease with an increase of the prediction time of bank failures. GARSINFIS utilizes more number of features and employs more number of rules to deal with the increasing complexity. It is encouraging to learn that GARSINFIS becomes more reliable with the increase of the prediction time because more positive rules are employed to achieve the highest sensitivity among all models. In bank failure prediction, sensitivity is more critical than specificity. Hence, GARSINFIS is the most reliable model (it gener-

ates the most number of correct early warnings to failing banks) to predict bank failures in one or two years' time.

Table 14 summarizes the input features selected by GARSINFIS and C4.5 decision tree. If a feature is utilized more than half of the time, then its value is highlighted to denote the importance. Although in all the three scenarios, GARSINFIS utilizes more number of input features than C4.5 decision tree, features selected by GARSINFIS are more concentrated. Among all nine financial covariates, CAPADE and PLAQLY are utilized in all GARSINFIS experiments and most C4.5 decision tree experiments. It is also noteworthy that PLAQLY in the Asset (Loan) Quality category (see Table 5) is significantly selected more times than the other two features in the same category and ROE in the Earnings category is significantly selected more times than NINMAR in the same category. Because financial covariates in the same category are highly correlated, they often contain highly overlapped knowledge. Therefore, feature selection and attribute reduction is necessary and

desired for a simple yet accurate inference rule base and GARSINFIS is shown to be reliable by employing such a compact knowledge base.

It is stated earlier in this paper that there are other NFIS models applied to the same bank failure prediction data set for different purposes with different configurations. Their performances are not compared and benchmarked again in this paper, but we highlight their pros and cons as follows. Tung et al. [41] first study this data set and compare their results to the traditional Cox's model [5]. Although the results are promising, the systematically generated trapezoidal shaped fuzzy membership functions overlap too much between each other and the number of fuzzy membership functions are unnecessarily large in some features (there are seven membership functions defined in GROWLA while only two defined by GARSC as shown in Fig. 5). Tan et al. [39] study this data set and compare their results to five other models. Although they managed to achieve 100% specificity and utilize simple fuzzy rules, the average sensitivity is less than 53%. Furthermore, the fuzzy membership functions overlap too much between each other and the number of fuzzy membership functions are unnecessarily large in some features (there are seven membership functions defined in CAPADE while only three defined by GARSC as shown in Fig. 5). Teddy et al. [40] study this data set and compare their results to four other models. Although the results are promising and improvements are clearly shown, the architecture employs a large number of synthesized memory cells (over 6000), which are impossible for human users to comprehend. Nguyen et al. [25] study this data set using a bio-inspired architecture, but the architecture still employs a large number of memory cells, which are incomprehensible. Quek et al. [30] study this data set with different configurations on the formation of training data sets. However, their sensitivity is significantly worse than specificity (especially for the two years prior scenario). Generally speaking, when comparing to the above mentioned models, GARSINFIS maintains a well balance between sensitivity and specificity, achieves highly competitive overall prediction accuracy and has an outstanding level of interpretability.

7. Conclusion and Future Works

Bank failure prediction is an important study to a bank's policy-makers, regulators and clients,

especially when there is an increasing number of deteriorating or failed banks in the past several years. Generally speaking, bank failures are normally due to financial distress and it is believed that financial distress does not develop out of the blue. The deterioration of the financial condition of distressed banks can be observed over time. Thus, the performance of a bank may be tracked and studied from its annual financial statements over a period of time. Many traditional statistical methods are applied to predict bank failures in the literature. However, they cannot explicitly specify what constitutes a financial distress and the intrinsic relationship between financial distress and failed banks. In this paper, we propose a novel neural fuzzy inference system to function as an early warning system, which is able to identify the inherent traits of financial distress based on nine financial covariates derived from publicly available annual financial statements. In contrast to other benchmarking models, our proposed early warning system provides a great level of interpretability and a highly competitive level of accuracy.

Our proposed self-organizing model is denoted as Genetic Algorithm and Rough Set Incorporated Neural Fuzzy Inference System (GARSINFIS), which utilizes the inference rule base automatically obtained by our proposed Genetic Algorithm based Rough Set Clustering (GARSC) technique. To systematically construct an early warning system, users only need to define a limited number of control parameters and constraints and there are no human intervention and expert guidance required. Empirical studies on the prediction of bank failures using GARSINFIS show encouraging results. GARSINFIS is shown to be a reliable system because it generates the most number of correct early warnings to failing banks in one or two years' time. It is also shown that the inference rules employed by GARSINFIS are accurate and straightforwardly comprehensible to all users because the rule base is compact. Furthermore, it is encouraging to learn that the employed rules are consistent with expert knowledge and the selected features are representative in their respective financial categories. Although genetic algorithm (an iterative optimization process) is incorporated, we managed to obtain a competitive level of prediction accuracy within 34 minutes of training time (testing time is negligible). In summary, GARSINFIS maintains a well balance between sensitivity

and specificity, achieves a highly competitive overall prediction accuracy, and has an outstanding level of interpretability. GARSINFIS is definitely in the front of the queue, which consists of all competent early warning systems to predict bank failures.

Although empirical results show the capability of GARSINFIS, further improvements are still required for more accurate prediction of bank failures. One possible improvement is to include more features other than the financial covariates used in this paper. For instance, Sarkar and Sriram [33] use several audit evidences such as the bank size, ownership characteristics, and management deficiencies to predict bank failures. Another possible improvement is on the formation of membership functions. As all statistical models do, we also assume the financial covariates are normally distributed. However, they are actually positively skewed [6]. Using membership functions with asymmetric geometry is expected to better describe the actual data distribution. A possible approach is to employ asymmetric Gaussian type of membership function [21], which allows two different widths defined on the opposite sides of the center. In the future, we will continue to improve GARSINFIS for a better prediction of bank failures and we will apply GARSINFIS to other financial forecasting applications as well.

References

- [1] E. I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, 23(4):589–609, 1968.
- [2] W. H. Beaver. Financial ratios as predictors of failure. *Journal of Accounting Research*, 4(1966):71–111, 1966.
- [3] T. Bernd, M. Kleutges, and A. Kroll. Nonlinear black box modelling - fuzzy networks versus neural networks. *Neural Computing and Applications*, 8:151–162, 1999.
- [4] J. C. Bezdek. *Pattern Recognition With Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, 1981.
- [5] R. A. Cole and J. W. Gunther. Separating the likelihood and timing of bank failure. *Journal of Banking and Finance*, 19(6):1073–1089, 1995.
- [6] E. B. Deakin. Distributions of financial accounting ratios: some empirical evidence. *Accounting Review*, 51:90–96, 1976.
- [7] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [8] I. Eccles and M. Su. Illustrating the curse of dimensionality numerically through different data distribution models. In *Proceedings of International Symposium on Information and Communication Technologies*, pages 232–237, 2004.
- [9] FDIC. Federal Deposit Insurance Corporation. Available online: <http://www.fdic.gov/>, accessible till July 2013.
- [10] FRBC. Federal Reserve Bank of Chicago. Available online: <http://http://www.chicagofed.org/>, 2012.
- [11] H. Frydman, E. I. Altman, and D.-L. Kao. Introducing recursive partitioning for financial classification: The case of financial distress. *Journal of Finance*, 40(1):269–291, 1985.
- [12] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing, 1989.
- [13] S. Haykin. *Neural networks: A Comprehensive Foundation*. Prentice Hall, 1998.
- [14] J. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- [15] J. S. R. Jang. ANFIS: Adaptive network-based fuzzy inference systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 23:650–684, 1993.
- [16] J. S. R. Jang, C. T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Pearson Education, 1996.
- [17] N. K. Kasabov and Q. Song. Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10:144–154, 2002.
- [18] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, 1992.
- [19] W. R. Lane, S. W. Looney, and J. W. Wansley. An application of the cox proportional hazards model to bank failure. *Journal of Banking and Finance*, 10(4):511–531, 1986.
- [20] K. C. Lee, I. Han, and Y. Kwon. Hybrid neural network models for bankruptcy predictions. *Decision Support Systems*, 18(1):63–72, 1996.
- [21] C. J. Lin and W. H. Ho. An asymmetry-similarity-measure-based neural fuzzy inference system. *Fuzzy Sets and Systems*, 152(3):535–551, 2005.
- [22] C. T. Lin and C. S. G. Lee. *Neural Fuzzy Systems*. Prentice-Hall, 1996.
- [23] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 26(12):1182–1191, 1977.
- [24] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1996.
- [25] M. N. Nguyen, D. Shi, and C. Quek. A nature inspired Ying-Yang approach for intelligent decision support in bank solvency analysis. *Expert Systems with Applications*, 34(4):2576–2587, 2008.
- [26] J. A. Ohlson. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, 18(1):109–131, 1980.

- [27] R. P. Paiva and A. Dourado. Interpretability and learning in neuro-fuzzy systems. *Fuzzy Sets and Systems*, 147(1):17–38, 2004.
- [28] Z. Pawlak. Rough sets. *International Journal of Information and Computer Science*, 11(5):341–356, 1982.
- [29] Z. Pawlak. *Rough Sets*. Kluwer Academic Publishers, 1991.
- [30] C. Quek, R. W. Zhou, and C. H. Lee. A novel fuzzy neural approach to data reconstruction and failure prediction. *Intelligent Systems in Accounting, Finance and Management*, 16(1-2):165–187, 2009.
- [31] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [32] S. M. Ross. *Introduction to Probability Models*. Academic Press, 1985.
- [33] S. Sarkar and R. S. Sriram. Bayesian models for early warning of bank failures. *Management Science*, 47(11):1457–1475, 2001.
- [34] Q. Shen and A. Chouchoulas. Rough set-based dimensionality reduction for supervised and unsupervised learning. *Applied Mathematics and Computer Science, Special Issue on Rough Sets and Their Applications*, 11(3):583–601, 2001.
- [35] Q. Song and N. Kasabov. ECM: A novel on-line, evolving clustering method and its applications. In *Proceedings of Conference on Artificial Neural Networks and Expert Systems*, pages 87–92, 2001.
- [36] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:13–33, 1988.
- [37] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 15(1):116–132, 1985.
- [38] K. Y. Tam. Neural network models and the prediction of bank bankruptcy. *Omega*, 19(5):429–445, 1991.
- [39] T. Z. Tan, C. Quek, and G. S. Ng. Biological brain-inspired genetic complementary learning for stock market and bank failure prediction. *Computational Intelligence*, 23(2):236–261, 2007.
- [40] S. D. Teddy, C. Quek, and E. M.-K. Lai. PSECMAC: A novel self-organizing multiresolution associative memory architecture. *IEEE Transaction on Neural Networks*, 19(4):689–712, 2008.
- [41] W. L. Tung, C. Quek, and P. Cheng. GenSo-EWS: A novel neural-fuzzy based early warning system for predicting bank failures. *Neural Networks*, 17(4):567–587, 2004.
- [42] D. Wang, C. Quek, and G. S. Ng. Novel self-organizing Takagi-Sugeno-Kang fuzzy neural networks based on ART-like clustering. *Neural Processing Letters*, 20(1):39–51, 2004.
- [43] L. X. Wang. *Adaptive fuzzy systems and control: Design and stability analysis*. Prentice-Hall, 1994.
- [44] Y. Y. Yao. A comparative study of rough sets and fuzzy sets. *Journal of Information Sciences*, 109:227–242, 1998.
- [45] L. A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.