GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures

W.L. Tung\textsuperscript{a}, C. Quek\textsuperscript{a,}\textsuperscript{*}, P. Cheng\textsuperscript{b}

\textsuperscript{a}Intelligent Systems Laboratory, Nanyang Technological University, School of Computer Engineering, Block N4 #2A-32, Nanyang Avenue, Singapore 639798
\textsuperscript{b}National Graduate School of Management, Sir Roland Wilson Building (120), McCoy Circuit Australian National University, Canberra ACT 0200, Australia

Received 3 February 2003; accepted 4 November 2003

Abstract

Bank failure prediction is an important issue for the regulators of the banking industries. The collapse and failure of a bank could trigger an adverse financial repercussion and generate negative impacts such as a massive bail out cost for the failing bank and loss of confidence from the investors and depositors. Very often, bank failures are due to financial distress. Hence, it is desirable to have an early warning system (EWS) that identifies potential bank failure or high-risk banks through the traits of financial distress. Various traditional statistical models have been employed to study bank failures [J Finance 1 (1975) 21; J Banking Finance 1 (1977) 249; J Banking Finance 10 (1986) 511; J Banking Finance 19 (1995) 1073]. However, these models do not have the capability to identify the characteristics of financial distress and thus function as black boxes. This paper proposes the use of a new neural fuzzy system [Foundations of neuro-fuzzy systems, 1997], namely the Generic Self-organising Fuzzy Neural Network (GenSoFNN) [IEEE Trans Neural Networks 13 (2002c) 1075] based on the compositional rule of inference (CRI) [Commun ACM 37 (1975) 77], as an alternative to predict banking failure. The CRI based GenSoFNN neural fuzzy network, henceforth denoted as GenSoFNN-CRI(S), functions as an EWS and is able to identify the inherent traits of financial distress based on financial covariates (features) derived from publicly available financial statements. The interaction between the selected features is captured in the form of highly intuitive IF-THEN fuzzy rules. Such easily comprehensible rules provide insights into the possible characteristics of financial distress and form the knowledge base for a highly desired EWS that aids bank regulation. The performance of the GenSoFNN-CRI(S) network is subsequently benchmarked against that of the Cox’s proportional hazards model [J Banking Finance 10 (1986) 511; J Banking Finance 19 (1995) 1073], the multi-layered perceptron (MLP) and the modified cerebellar model articulation controller (MCMAC) [IEEE Trans Syst Man Cybern: Part B 30 (2000) 491] in predicting bank failures based on a population of 3635 US banks observed over a 21 years period. Three sets of experiments are performed-bank failure classification based on the last available financial record and prediction using financial records one and two years prior to the last available financial statements. The performance of the GenSoFNN-CRI(S) network as a bank failure classification and EWS is encouraging.

Keywords: Bank failure prediction; GenSoFNN; Compositional rule of inference; Discrete incremental clustering; And fuzzy rule analysis

1. Introduction

The collapse and failure of a bank could have devastating consequences to the entire banking system and an adverse repercussion effect on other banks and financial institutions. Some of the negative impacts are the massive bail out cost for a failing bank and the negative sentiments and loss of confidence developed by investors and depositors. Hence, bank failure prediction is an important issue for regulators of the banking industries. Very often, bank failures are due to financial distress. Various traditional models have been employed to study bank failures. The more popular statistical methodologies used are multivariate discriminant analysis (MDA) (Sinkey, 1975), logit analysis (Martin, 1977) and Cox’s proportional hazards model (the Cox’s model) (Lane, Looney, & Wansley, 1986) (Cole & Gunther, 1995). However, these models have various deficiencies (Cheng, 2002), and they are not able to identify the traits of financial distress that lead to bank failure, thus functioning as black boxes.
On the other hand soft computing (Zadeh, 1994), which emulates the human style of reasoning and decision-making when solving complex problems, can overcome the deficiencies of the traditional statistical models and can be employed to handle the classification of both failed and survived (non-failing) banks. The objective of various soft computing approaches is to synthesize the human ability to tolerate and process uncertain, imprecise and incomplete information during the decision-making process. A popular approach is the integration of neural network and fuzzy system to create a hybrid structure known as a neural fuzzy network. Neural fuzzy (or neuro-fuzzy) networks (Lin & Lee, 1996; Nauck, Klawonn, & Kruse, 1997) such as the Generic Self-organising Fuzzy Neural Network (GenSoFNN) (Tung & Quek, 2002c), the pseudo-outer-product fuzzy neural network (POPNFN) family of networks (Ang, Quek, & Pasquier, 2003; Quek & Zhou, 1996, 1999), the adaptive neuro-fuzzy inference system (ANFIS) (Jang, 1993), Falcon-ART (Lin & Lin, 1997), Falcon-MART (Quek & Tung, 2000) and other Falcon-based networks (Tung and Quek, 2003, 2002b) are the realizations of the functionality of fuzzy systems using neural techniques. The main advantage of a neural fuzzy network is its ability to model the characteristics of a given problem using a high-level linguistic model instead of low-level complex mathematical expressions. The linguistic model is essentially a fuzzy rule-base consisting of a set of IF–THEN fuzzy rules. The fuzzy rules are highly intuitive and easily comprehended by the human users. In addition, the hybrid structure of a neural fuzzy network is highly transparent as the fuzzy rules can be used to interpret the weights and linkages of the connectionist network. Moreover, a neural fuzzy network can self-adjust the parameters of the fuzzy rules using learning algorithms derived from the neural network paradigm.

Neural fuzzy networks are universal data-mining tools (Nauck et al., 1997; Lin & Lee, 1996) and possess strong capability to derive the intrinsic relationships between the observed inputs and outputs. In addition, the generalization attribute of neural fuzzy systems enables them to interpolate the decision-making process to new (unseen) cases. This serves very well the objectives of a bank failure prediction (and classification) system in the study of bank failures since neural fuzzy networks can be employed to identify the inherent characteristics of failed banks. It allows one to interpret the traits of the financial distress that leads to a bank failure. However, existing neural fuzzy systems have some major deficiencies (Tung & Quek, 2002c) that hinder their performances and thus the GenSoFNN network is developed to overcome such shortcomings. In this paper, the performance of the Cox’s model in predicting bank failures is compared against that of the CRI based GenSoFNN network (GenSoFNN-CRI(S)). The GenSoFNN-CRI(S) network serves as an early warning system (EWS) and is used to analyze the solvency of banks given the financial covariates extracted from the last annual financial statement as well as 1 and 2 years prior to the last annual financial statement of the observation period. That is, the banks are classified as failed or survived (non-failing) banks based on their financial performances. Details on the selection of the financial covariates are provided in Section 6. In the study of bank failures, there are different concepts of failure—economic, business and official—and there are further distinctions within each of these concepts (Cheng, 2002). In this paper, regulatory closure is the defining event of failure, the reasons being that the event of regulatory closure is unambiguous and is more important and consistent than the straight-forward identification of problem banks, as such banks might come good in the future, given time or financial assistance or both. Besides, only the regulatory authorities can revoke or remove a bank’s charter to operate under existing ownership.

This paper is organized as follows. Section 2 briefly discusses the CRI scheme which is responsible for the reasoning and decision-making capability of the GenSoFNN-CRI(S) network and Section 3 presents an overview of the structure, functionality and training cycle of the GenSoFNN framework from which the GenSoFNN-CRI(S) network is developed. Section 4 describes the computations performed by the various layers of nodes in the GenSoFNN-CRI(S) network and presents the step-by-step mapping of these computations to the logical inference steps of the CRI scheme. In Section 5, the back propagation (BP) learning equations of the GenSoFNN-CRI(S) network are highlighted. Section 6 introduces the Cox’s model used in traditional bank failure prediction. In addition, the process of selecting the financial indicators (covariates) used in the bank failure prediction experiment in the paper is also highlighted. Section 7 presents the experimental results of the GenSoFNN-CRI(S) network when applied to the prediction and classification of failed and survived (non-failing) banks. Section 8 concludes this paper.

2. The compositional rule of inference scheme

Consider a simple fuzzy rule as shown in Eq. (1).

\[
\text{If } \underline{x \text{ is } A} \quad \text{Then } \underline{y \text{ is } B} \quad \text{(1)}
\]

where \(x\) and \(y\) denote the input and output variable taking values in the universe of discourse \(U_1\) and \(U_2\) respectively; \(A\) is a fuzzy set defined on \(U_1\); and \(B\) is a fuzzy set defined on \(U_2\).

Under the CRI scheme (Zadeh, 1975), such a fuzzy rule is interpreted as a fuzzy relation \(R\) (Lin & Lee, 1996) restricting the possible values of the ordered pair \((x, y)\). This fuzzy relation \(R\) has a membership function \(\mu_R\) (since it is a fuzzy set) that can be resolved using a generalized.
implication operator \( I \) defined as follows.

\[
\mu_R : U_1 \times U_2 \rightarrow [0, 1]
\]

where \( \mu_R(x, y) = I(\mu_A(x), \mu_B(y)) \forall (x, y) \in U_1 \times U_2 \); and \( I : [0, 1]^2 \rightarrow [0, 1] \).

where \( \mu_A(x) \) and \( \mu_B(y) \) denote the membership functions of fuzzy sets \( A \) and \( B \), respectively. Since the introduction of the CRI scheme, many fuzzy implication operators have been proposed. The following lists six of them:

- \( \mu(x) = \max\{\min(\mu_A(x), \mu_B(x)), (1 - \mu_A(x))\} \) (Zadeh, 1975)
- \( \mu(x) = \min(1, \frac{\mu_B(y)}{\mu_A(x)}) \) (Yager, 1980)
- \( \mu(x) = \min(1 - \mu_A(x) + \mu_B(y)) \) (Mamdani, 1977)
- \( \mu(x) = \min(\mu_A(x), \mu_B(y)) \) (Bandler & Kohout, 1980)

The most commonly used implication operator in CRI schemes is the \( \min \) operator (Mamdani, 1977) due to its simplicity in computation. This is denoted as \( \mu_m \) in subsequent discussions. When a fuzzy rule is interpreted as a fuzzy relation \( R \) and its membership function \( \mu_R \) is resolved using an implication operator, it is then possible to derive a valid conclusion for the input to a fuzzy system from its fuzzy rule-base using the generalized modus Ponens (GMP) (Lin & Lee, 1996) inference rule.

2.1. Approximate reasoning using GMP

Assume a simple single-input–single-output (SISO) fuzzy system with the following description.

\[
\text{Fuzzy rule: } \begin{array}{l}
\text{If } \mu_A(x) \text{ Then } \mu_B(y) \\
\mu(x) = \mu_A(x) \quad \mu(x) = \mu_B(y)
\end{array}
\]

\[
\text{Observed input: } \mu(x) = \mu_A(x) \\
\mu(x) = \mu_A(x)
\]

\[
\text{Conclusion: } \mu_B(y) \quad \text{(To be inferred)}
\]

Where \( x \) and \( y \) denote the input and output variable taking values in the universe of discourse \( U_1 \) and \( U_2 \) respectively; \( A \) and \( \tilde{A} \) are fuzzy sets defined on \( U_1 \); and \( B \) and \( \tilde{B} \) are fuzzy sets defined on \( U_2 \). The following steps compute the inferred conclusion using the GMP rule.

Step 1: Translates the fuzzy rule into a fuzzy relation \( R \) and resolves its membership function \( \mu_R \) using an implication operator. This paper uses \( \mu_m \). From Eq. (2),

\[
\mu_R(x, y) = I(\mu_A(x), \mu_B(y)) = I_m(\mu_A(x), \mu_B(y))
\]

where \( \mu_A(x) \) and \( \mu_B(y) \) are fuzzy sets defined on \( U_1 \) and \( U_2 \), respectively. Since the introduction of the CRI scheme, many fuzzy implication operators have been proposed. The following lists six of them:

- \( \mu(x) = \max\{\min(\mu_A(x), \mu_B(x)), (1 - \mu_A(x))\} \) (Zadeh, 1975)
- \( \mu(x) = \min(1, \frac{\mu_B(y)}{\mu_A(x)}) \) (Yager, 1980)
- \( \mu(x) = \min(1 - \mu_A(x) + \mu_B(y)) \) (Mamdani, 1977)
- \( \mu(x) = \min(\mu_A(x), \mu_B(y)) \) (Bandler & Kohout, 1980)

The most commonly used implication operator in CRI schemes is the \( \min \) operator (Mamdani, 1977) due to its simplicity in computation. This is denoted as \( \mu_m \) in subsequent discussions. When a fuzzy rule is interpreted as a fuzzy relation \( R \) and its membership function \( \mu_R \) is resolved using an implication operator, it is then possible to derive a valid conclusion for the input to a fuzzy system from its fuzzy rule-base using the generalized modus Ponens (GMP) (Lin & Lee, 1996) inference rule.

Step 2: Fuzzify the observed input \( x \) is \( \tilde{A} \) according to Eq. (4). The label \( \tilde{A} \) denotes a fuzzy set defined on \( U_1 \) and may be interpreted as a singular fuzzy relation on \( U_1 \).

Input fuzzy proposition \( p' = \mu_A(x) \)

Step 3: Compute the fuzzy extension of \( \tilde{A} \), denoted as \( A \cup U_2 \), to relate the input space \( U_1 \) to the output space \( U_2 \).

\[
\mu_A(x) \cup U_2 (x, y) = \mu_A(x)
\]

Where \( A \cup U_2 \) defines a fuzzy relation spanning the Cartesian space \( U_1 \times U_2 \).

Step 4: Since both fuzzy relations \( R \) and \( A \cup U_2 \) are fuzzy sets spanning \( U_1 \times U_2 \), their intersection is computed using a \( T \)-norm operation as shown in Eq. (6).

\[
\mu_R \cap (A \cup U_2)(x, y) = T \left( \mu_A(x), \min(\mu_A(x), \mu_B(y)) \right)
\]

Step 5: The fuzzy relation \( R \cap (A \cup U_2) \) is subsequently projected (Lin and Lee, 1996) onto the output space \( U_2 \) to obtain the output \( y \) in relation to the input \( x \) and the rule-base of the fuzzy system. That is,

\[
\mu_B(y) = \mu \left( R \cap (A \cup U_2) \cup U_2 \right)(y), \quad \forall y \in U_2
\]

For simplicity, the \( \min \) operator is often used for the \( T \)-norm operation in Eq. (7). Hence, Eq. (8) results.

\[
\mu_B(y) = \sup_{x \in U_1} \{ \min(\mu_A(x), \mu_B(y)) \}, \quad \forall y \in U_2
\]
Consequently, Eq. (8) can be expressed as
\[
\mu_B(y) = \sup_{x \in U_1} \left\{ \min \{ \mu_A(x), \mu_B(y) \} \right\}, \quad \forall y \in U_2
\]

[Associative property of min operation]
\[
= \sup_{x \in U_1} \left\{ \min \{ \min(\mu_A(x), \mu_B(y)) \} \right\}
\]

[supoperation is independent of y]
\[
= \min_{x \in U_1} \left\{ \sup \{ \min(\mu_A(x), \mu_B(y)) \} \right\}
\]

Hence, the inferred conclusion y is \( \bar{B} \) is computed using Eq. (9). In Section 4, the inference steps of the CRI scheme are systematically mapped to the operations of the GenSoFNN-CRI(S) network.

3. The generic self-organising fuzzy neural network (GenSoFNN)

The main problems (Tung & Quek, 2002c) dogging most existing neural fuzzy systems are: (1) Susceptibility towards noisy/spurious training data due to the choice of clustering technique; (2) the Stability-Plasticity dilemma (Lin & Lee, 1996) in which the neural fuzzy system is not flexible enough to incorporate new clusters of data (knowledge) after training has completed; (3) Require prior knowledge of the number of clusters to be computed or has to predefine the number of fuzzy rules to be formulated; (4) Inconsistent rule-base or inconsistent representation of fuzzy labels (Nauck et al., 1997); and (5) Operations of the nodes are of heuristic nature or not clearly defined. These deficiencies are essentially due to the architectural design and training techniques employed to construct the neural fuzzy systems. Thus, the GenSoFNN (Tung & Quek, 2002c) is developed with a lean and structured training cycle that consists of three distinct phases: (a) Self-organising: clustering of the numerical training data into fuzzy sets; (b) Rule formulation: constructing a set of IF–THEN fuzzy rules using the computed fuzzy sets to adequately represent the underlying knowledge of the training data; and (c) Parameter learning: supervised tuning of the fuzzy sets to achieve the desired output response(s). The well-ordered training cycle of the GenSoFNN network serves as a basis for the crafting of a consistent rule-base and as a primary solution for the deficiencies listed above.

3.1. Structure of GenSoFNN

The GenSoFNN network (Tung & Quek, 2002c) (Fig. 1) consists of five layers of nodes. Each input node IV, \( i \in \{1, \ldots, n1\} \), has a single input denoted as \( x_i \). The vector \( X = [x_1, \ldots, x_i, \ldots, x_{n1}]^T \) represents all the inputs to the GenSoFNN network. Each output node OV, \( m \in \{1, \ldots, n5\} \), computes a single output denoted by \( y_m \). The vector \( Y = [y_1, \ldots, y_m, \ldots, y_{n5}]^T \) denotes the outputs of the GenSoFNN network with respect to the input stimulus \( X \). In addition, the vector \( D = [d_1, \ldots, d_m, \ldots, d_{n5}]^T \) represents the desired outputs required for the BP (Rumelhart, Hinton, & Williams, 1986) based parameter-learning phase of the training cycle.

The trainable weights of the GenSoFNN network are found in layers 2 and 5 (enclosed in rectangular boxes in Fig. 1). Layer 2 links contain the parameters of the input fuzzy sets while layer 5 links contain the parameters of the output fuzzy sets. The weights of the remaining connections are unity. The trainable weights (parameters) are interpreted as the corners of the normal trapezoidal-shaped fuzzy sets (Fig. 2) computed by the DIC (Tung & Quek, 2002a) technique employed by the GenSoFNN network, where the maximum membership is unity. They are denoted as \( l \) and \( r \) (left and right support points), \( u \) and \( v \) (left and right kernel points). The subscripts denote the pre-synaptic and post-synaptic nodes respectively. For clarity in subsequent discussions, the variables \( i,j,k,l,m \) are used to refer to arbitrary nodes in layers 1, 2, 3, 4 and 5, respectively. The output of a node is denoted as \( Z \) and the subscripts specify its origin.

Each input node IV may have different number of input fuzzy terms \( J_i \). The input fuzzy terms are denoted as \( \text{IL}_{ij} \), where \( i = \{1, \ldots, n1\} \) and \( j = \{1, \ldots, J_i\} \). Hence, the number of layer 2 nodes is \( n2 = \sum_{i=1}^{n1} J_i \). Layer 3 consists of the rule nodes \( R_k \), where \( k = \{1, \ldots, n3\} \). In layer 4, an output fuzzy term node \( \text{OL}_{lm} \) may have more than one fuzzy rule attached to it. Each output node \( \text{OV}_m \) in layer 5 can have different number of output fuzzy terms \( L_m \). Hence, the number of layer 4 nodes is \( n4 = \sum_{m=1}^{n5} L_m \). In Fig. 1, the solid arrows denote the links that are used during the feedforward normal operation of the GenSoFNN network. The dashed arrows denote the backward links used during the self-organising phase of the training cycle of the GenSoFNN. The GenSoFNN network adopts the Mamdani fuzzy model (Nauck et al., 1997). The \( k \)th fuzzy rule has the form as shown in Eq. (10).

\[
R_k : \text{IF} \ x_1 \ \text{IS} \ \text{IL}_{1(i_1)j_1}, \ldots, \ x_i \ \text{IS} \ \text{IL}_{i(i_j)j_i}, \ldots, \ x_{n1} \ \text{IS} \ \text{IL}_{n1(i_l)j_l}
\]

Then \( y_1 \ \text{IS} \ \text{OL}_{1(i_{l1})l_1}, \ldots, \ y_m \ \text{IS} \ \text{OL}_{m(i_{l_m})l_m}, \ldots, \ y_{n5} \ \text{IS} \ \text{OL}_{n5(i_l)l_h}
\]

(10)

where \( \text{IL}_{i(l)j} \) is the \( j \)th fuzzy label of the \( i \)th input that is connected to rule \( R_k \); and \( \text{OL}_{m(l_m)l_h} \) is the \( l_h \)th fuzzy label of the \( m \)th output to which rule \( R_k \) is connected to.

3.2. Self-organisation of GenSoFNN

The DIC (Tung & Quek, 2002a) technique is developed and integrated into the GenSoFNN network to automatically compute the input–output clusters from the numerical training data. The DIC technique maintains a consistent representation of the fuzzy sets (fuzzy labels) by performing clustering on a local basis. This is similar to the ART (Grossberg, 1976) concept. However, unlike ART,
the number of fuzzy sets for each input/output dimension may be different and if the fuzzy label (fuzzy set) for a particular input/output dimension already exists, then it is not ‘re-created’. Hence, DIC ensures that a fuzzy label is uniquely defined by a fuzzy set and this serves as a basis to formulate a consistent rule-base using the GenSoFNN network.

The DIC technique has five parameters: a plasticity parameter $\beta$, a tendency parameter $TD$, an expansion parameter $STEP$, a membership threshold $MT$ and a fuzzy set support parameter $SLOPE$. The plasticity parameter $\beta$ and the parameter $STEP$ control the appropriate expansion of a fuzzy set to include a new training data point. The tendency parameter $TD$ maintains the integrity of a fuzzy set so that only similar data points are clustered together while $SLOPE$ defines the gradients of the left and right-sided slopes of the trapezoidal fuzzy sets. The membership threshold $MT$ determines the minimum membership value a data point should have before it is considered for inclusion into an existing cluster. Else a new cluster is created to hold the dissimilar data point. During the self-organisation phase, the plasticity parameter $\beta$ and the tendency parameter $TD$ for an expanding cluster (fuzzy set) is gradually reduced to constrain its future expansion. For the interested reader, more details on the DIC technique and the use of its various parameters are reported in (Tung & Quek, 2002a,c).

The algorithmic form of the DIC technique is presented as follows. DIC is applied to cluster both the input and output data points.

**Algorithm DIC**

Assume a data set $\tilde{X} = \{X^{(1)}, \ldots, X^{(p)}, \ldots, X^{(P)}\}$, where $P$ is the number of training vectors and $X^{(p)} = \{x^{(p)}_1, \ldots, x^{(p)}_i, \ldots, x^{(p)}_n\}$ denotes the $p$th training vector in the space $\mathbb{R}^n$. Initialize $STEP$, $SLOPE$ and set $\beta = TD = 0.5$. The threshold $MT \in (0, 1]$ is user-defined.

For all training vector $X^{(p)}$, where $p \in \{1 \ldots P\}$ do {

For all dimensions $i \in \{1\ldots n\}$ do {

If there are no clusters (fuzzy sets) in the $i$th dimension (i.e. $J_i = 0$)

Create a new cluster using the point $x^{(p)}_i$

Otherwise

\[ Membership \ \mu_{R_i} (x_i) \]

For all $x_i \in \mathbb{R}^n$ do {

1.0

}\[ x_{i,j} \]

}\[ x_{i,j} \]

Fig. 1. Structure of the GenSoFNN network.

Fig. 2. Normal trapezoidal fuzzy set representing the $j$th fuzzy term of the $i$th input (denoted as $\mu_{\alpha_i}$).
Find the best-fit cluster Winner for \( x_i^{(p)} \) using Eq. (11).

\[
\text{Winner} = \arg \max_{j \in \{1 \ldots J\}} \mu_{ij}(x_i^{(p)})
\]  

(11)

where \( \mu_{ij} \) is the membership function of the \( j \)th fuzzy set in dimension \( i \). If \( \mu_{ij,\text{Winner}}(x_i^{(p)}) > MT \) /* Membership value greater than threshold */ Update the kernel of Winner /* grows the cluster Winner */ Update \( \beta \) and TD Otherwise Create a new cluster using the point \( x_i^{(p)} \)  

End For all dimensions \( i \in \{1 \ldots n\} \)  

End For all training vector \( X^{(p)}, p \in \{1 \ldots P\} \)

3.3. Rule formulation of GenSoFNN

The rule-base of the GenSoFNN network is formulated from the numerical training data pairs \( (X, D) \) using a rule mapping process named RuleMAP. Under the GenSoFNN framework, input space partition of rule \( k \) (ISP\(_k \)) is the collective term for all the input fuzzy labels (layer 2 nodes) that contribute to the antecedent of rule node \( R_k \). Similarly, output space partition of rule \( k \) (OSP\(_k \)) refers to all the output fuzzy labels (layer 4 nodes) that form the consequent of rule \( R_k \). During the rule mapping process, each rule \( R_k, k \in \{1, \ldots, n3\} \), activates its ISP (OSP) with a firing of layers 1 and 2 (layers 4 and 5) of the GenSoFNN network with the input stimulus \( X \) (desired outputs \( D \)) feeding into layer 1 (layer 5). The backward links depicted by the dashed, gray arrows in Fig. 1 are used for the activation of the OSPs. Fig. 3 presents the flowchart of the RuleMAP process with the embedded self-organising and parameter learning phases.

The function EstLink identifies the proper connections between the input fuzzy labels (layer 2 nodes), the fuzzy rules (layer 3 nodes) and the output fuzzy labels (layer 4 nodes). Overlapping input/output labels are annexed and their respective rules are combined if necessary to maintain a consistent rule-base. Details on the RuleMAP process are described in (Tung, 2001). The RuleMAP process is responsible for the structural learning of the GenSoFNN network. The crafted rule-base is consistent but not compact, as there may be numerous redundant and/or obsolete rules. Redundant and obsolete rules are the results of the dynamic training of the GenSoFNN where the fuzzy sets of the fuzzy rules are constantly tuned by the back-propagation algorithm. To maintain the integrity, accuracy as well as the compactness of the rule-base, these redundant rules are deleted at the end of each training epoch.

3.4. Cluster defragmentation

Although the DIC clustering technique employed by the GenSoFNN network does not require a prior knowledge of the number of clusters to be computed, it suffers from a condition known as ‘cluster fragmentation’. That is, numerous small clusters may be created...
3.5. Pruning of weak/insignificant rules

The rule formulation phase of the training cycle of the GenSoFNN network is responsible for the derivation of the fuzzy rules based on the computed clusters from the self-organizing phase. The appropriate input space partitions are mapped or linked to the appropriate output space partitions through the layer-3 rule nodes to derive the fuzzy rules. The GenSoFNN network adopts the incremental rule-learning approach. That is, no fuzzy rules initially existed and they are constructed only if there are training data points that justified the existence of the fuzzy rules. However, some rules may be more important than others in the modeling of the problem domain, especially if their input–output space partitions covered a significant portion of the input–output space. There are also insignificant/weak rules created due to the existence of noisy/spurious training data points. These insignificant or weak rules may interfere and contribute errors to the network outputs during the output inference process. Hence, such insignificant rules are identified and removed. In the GenSoFNN network, the strengths of the fuzzy rules are computed during the training cycle and rules with strengths that fall below a predefined threshold ThresPr are pruned away. The strength of a fuzzy rule \( R_k \), denoted as \( S_k \), is computed using Eq. (12).

\[
S_k(T+1) = S_k(T) + (F_{ISP}(T) \times F_{OSP}(T)), \quad S_k(0) = 0 \tag{12}
\]

where \( F_{ISP}(T) \) is the forward aggregated input to rule \( R_k \) due to the activation of its ISP; and \( F_{OSP}(T) \) is backward aggregated input to \( R_k \) due to the activation of its OSP at time \( T \); and \( S_k(0) \) is the initial strength of a newly created rule \( R_k \) in the GenSoFNN network. Fig. 5 illustrates the concept of \( F_{ISP}(T) \) and \( F_{OSP}(T) \) of a fuzzy rule \( R_k \).

Hence, at the end of the training cycle, the aggregated sum of all the rule strengths in the GenSoFNN network is computed and a rule \( R_k \) is pruned if Eq. (13) equates as true.

\[
\frac{1}{N} \sum_{k=1}^{N} S_k \leq \text{ThresPr} \quad \tag{13}
\]

where ThresPr is a user pre-defined parameter for pruning of weak/insignificant rules.

Two motivations drive the development of the GenSoFNN network. The first is to define a systematic way of crafting the linguistic model required in neural fuzzy systems to describe the dynamics and characteristics of a problem domain. The second motivation is to develop a generalised network architecture whereby different fuzzy inference schemes such as the Compositional Rule of Inference (CRI) (Zadeh, 1975), truth-value restriction (TVR) (Mantaras, 1990) and approximate analogical reasoning schema (AARS) (Turksen & Zhong, 1990) can be readily mapped onto with ease. This correlates to the definition of a neural fuzzy network (system). That is, a neural fuzzy network is the integration of fuzzy system and neural network whereby the operations of the hybrid system should be functionally equivalent to a similar standalone fuzzy system based on the adopted fuzzy inference scheme. Hence, the operations and outputs of the various nodes in the GenSoFNN network are defined by the fuzzy inference scheme adopted by the network. In this paper, the CRI inference scheme is mapped onto the GenSoFNN network to define the node functions. The resultant network is henceforth referred to as GenSoFNN-CRI(S), where \( S \) denotes the singleton fuzzifiers (Mendel, 2001) implemented at the input layer of the network. Section 4 presents the detailed operations of the various layers of nodes in the GenSoFNN-CRI(S) network.

4. The GenSoFNN-CRI(S) network

The GenSoFNN-CRI(S) network is synthesized by mapping the operations of the CRI inference scheme onto the generic structure of the GenSoFNN network. The aggregation and activation functions of an arbitrary node in layer \( I \in \{1, \ldots, 5\} \) are denoted as \( f^{(I)} \) and \( a^{(I)} \), respectively.

4.1. Layer 1: singleton fuzzifier

Layer 1 nodes are the input nodes of the GenSoFNN-CRI(S) network. They act as singleton fuzzifiers that
perform fuzzification of crisp-valued inputs presented to the GenSoFNN-CRI(S) network. Fuzzification of the inputs is necessary in order to map the operations of the GenSoFNN-CRI(S) network to the CRI inference scheme. With respect to input node IV, Net synaptic input of node IV, Net synaptic output of node IV, where \( x_i \) is the \( i \)th input to the GenSoFNN-CRI(S) network and \( \tilde{x}_i \) is the fuzzified equivalent of crisp input \( x_i \). The fuzzified input \( \tilde{x}_i \) is the only element in the fuzzy set \( \tilde{X}_i \) whose membership function \( \mu_{\tilde{X}_i} \) is defined as:

\[
\mu_{\tilde{X}_i}(\tilde{x}_i) = \\
\begin{cases} 
1, & \text{if } \tilde{x}_i = x_i \\
0, & \text{otherwise}
\end{cases}
\]  

(16)

The graphical interpretation of the singleton fuzzifier is depicted as Fig. 6.

4.2. Layer 2: antecedent matching

Layer 2 nodes are known as the input fuzzy label (or term) nodes and they encapsulate the input fuzzy sets of the GenSoFNN-CRI(S) network. The function of the layer 2 nodes is to perform antecedent matching of the fuzzified inputs against the encapsulated fuzzy sets and compute a similarity measure that is the membership value (MV). This membership value is presented as output of the layer 2 nodes. With respect to the input fuzzy term node IL, Net synaptic input of node IL, Net synaptic output of node IL, where \( x_i \) is the \( i \)th fuzzy label of the GenSoFNN-CRI(S) network; and \( \tilde{x}_i \) is the fuzzified equivalent of crisp input \( x_i \). The fuzzified input \( \tilde{x}_i \) is the only element in the fuzzy set \( \tilde{X}_i \) whose membership function \( \mu_{\tilde{X}_i} \) is defined as:

\[
\mu_{\tilde{X}_i}(\tilde{x}_i) = \\
\begin{cases} 
1, & \text{if } \tilde{x}_i = x_i \\
0, & \text{otherwise}
\end{cases}
\]  

(16)

The graphical interpretation of the singleton fuzzifier is depicted as Fig. 6.

4.3. Layer 3: rule fulfillment

Layer 3 of the GenSoFNN-CRI(S) network contains the fuzzy rule-base of the network. Each layer 3 node \( R_k \) is a fuzzy rule. There are a total of \( n^3 \) rule nodes in layer 3. Each rule node \( R_k \) computes the degree of fulfillment of the current inputs with respect to the antecedents of the fuzzy rule it denotes. The higher the degree of fulfillment, the greater is the compatibility of the inputs to the input space partition (ISP) of rule \( R_k \). Each rule node \( R_k \) presents the computed degree of rule fulfillment as its output. The net synaptic input of node \( R_k \), denoted as \( Net_{R_k} \), and the net synaptic output \( Z_{R_k} \) are defined as:

Net synaptic input of node \( R_k \):

\[
Net_{R_k} = f^{(3)}(Z_{(1)j}, \ldots, Z_{(n)j}, \ldots, Z_{(n)1})
\]  

(20)

Net synaptic output of node \( R_k \):

\[
Z_{R_k} = d^{(3)}(Net_{R_k})
\]  

(21)

where \( Z_{(j)l} \) is the output of the \( j \)th fuzzy label of the \( l \)th input (\( IL_{ij} \)) that is connected to rule \( R_k \); and \( n \) is the number of inputs to the GenSoFNN-CRI(S) network.

4.4. Layer 4: consequent derivation

The layer 4 nodes in the GenSoFNN-CRI(S) network are the output fuzzy term nodes that encapsulate the consequent of the fuzzy rules in layer 3. Each layer 4 node denotes an output fuzzy set. The parameters of the output fuzzy sets are implemented as weights on layer 5 links (Fig. 1). Each \( OL_{lm} \) term node may have more than one fuzzy rule feeding into it because different rules can share the same consequent. However, each \( OL_{lm} \) node is connected to only one output node \( OV_m \) in layer 5. The net synaptic input \( Net_{lm} \) and net

![Image](88x75 to 251x171)
synaptic output $Z_{lm}$ of node $OL_{lm}$ are defined as:

Net synaptic input of node $OL_{lm}$,

$$\text{Net}_{lm} = f^{(4)}(Z_{R_1}^{(lm)}, \ldots, Z_{R_{LZ}}^{(lm)})$$

where $Z_{R_k}^{(lm)}$ is the output of the $k$th rule in the GenSoFNN-CRI(S) network as its consequent; and $L_m$ is the total number of rules in the GenSoFNN-CRI(S) network.

Layer 5 nodes are the output nodes of the GenSoFNN-CRI(S) network. There are a total of $n_5$ output nodes. During the feed-forward operation of the GenSoFNN-CRI(S) network, the output nodes are responsible for the defuzzification of the derived fuzzy outputs from the CRI inference scheme before presenting them as crisp network outputs. A weighted center of averaging (COA) (Lin & Lee, 1996) defuzzification technique is used to compute the crisp network outputs. The net synaptic input $Net_{OV_m}$ and net synaptic output $y_m$ for output node $OV_m$ are defined as:

Net synaptic input for node $OV_m$,

$$\text{Net}_{OV_m} = f^{(5)}(Z_{l_{m1}}, \ldots, Z_{l_{mL_m}}, \ldots, Z_{l_{mL_m}})$$

where $Z_{l_{mL_m}}$ is the output of node $OL_{lm}$ in layer 4; $L_m$ is the number of output term nodes $OV_m$ has; and $\tilde{m}_{lm}$ is the mid point of the kernel of the fuzzy set represented by output term $OL_{lm}$. The parameter $\tilde{m}_{lm}$ is defined by Eq. (26).

$$\tilde{m}_{lm} = \frac{(u_{lm} + v_{lm})}{2}$$

As a neural implementation of a CRI-based fuzzy system with singleton fuzzifiers (henceforth denoted as CRI-FS), the operations and outputs of the nodes in the GenSoFNN-CRI(S) network are derived from the inference steps of such a fuzzy system. The functional equivalence between the GenSoFNN-CRI(S) network and the CRI-FS is demonstrated as follows. Assume that the CRI-FS has $n_1$ inputs and $n_5$ outputs.

**Step 1: Fuzzification**

Since the inputs to the CRI-FS are crisp-valued, fuzzification has to be performed before the inference engine can make use of the fuzzified inputs to compute the appropriate fuzzified outputs. The vector $X = [x_1, x_2, \ldots, x_{n_1}]^T$ denotes the inputs to the CRI-FS. For crisp input $x_j$, it is fuzzified into its corresponding fuzzy set $\tilde{X}_j$ using the singleton fuzzifier defined in Eq. (16). The operation of the singleton fuzzifier is subsequently mapped onto layer 1 of the GenSoFNN-CRI(S) network using Eqs. (14) and (15). Hence, the input vector $X = [x_1, x_2, \ldots, x_{n_1}]^T$ becomes $X = [\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_{n_1}]^T$.

**Step 2: Antecedent matching**

The fuzzified inputs from Step 1 are then compared against their corresponding input labels that form the antecedent section of the fuzzy rules in CRI-FS. For fuzzified input $\tilde{X}_j$, its corresponding $j$th fuzzy label is denoted as $IL_{ij}$. Both $\tilde{X}_j$ and $IL_{ij}$ are fuzzy sets defined on the universe of discourse $U_j$, the range of possible values for input $x_j$. The antecedent matching between $\tilde{X}_j$ and $IL_{ij}$ computes a similarity measure (denoted as $MV$ in Section 4.2) by taking the intersection of their fuzzy sets as shown.

$$MV = \tilde{X}_j \cap IL_{ij}$$

The intersection of two fuzzy sets is computed using the $T$-norm operator. Hence, $MV = \tilde{X}_j \cap IL_{ij} = T(\mu_{\tilde{X}_j}(x_j), \mu_{IL_{ij}}(x_j)), \forall x_j \in U_j$.

But $\tilde{X}_j$ is a singleton with only one element $\bar{x}_j$ that is defined when $\bar{x}_j = x_j \in U_j$. Therefore, $MV = T(\mu_{\tilde{X}_j}(\bar{x}_j), \mu_{IL_{ij}}(\bar{x}_j)), \forall x_j \in U_j$.

$$MV = T(\mu_{\tilde{X}_j}(\bar{x}_j), \mu_{IL_{ij}}(\bar{x}_j))$$

Thus, Eq. (29) demonstrated that performing the antecedent matching between the inputs and the antecedent section of the fuzzy rules is essentially to compute the membership values of the inputs with respect to the input.
fuzzy sets. One can easily observe that Eq. (29) is readily mapped onto layer 2 of the GenSoFNN-CRI(S) network using Eqs. (17) and (18).

Step 3: Rule fulfillment

This step is to compute the overall similarity of the inputs to the antecedent of the fuzzy rules. Since the propositions in the antecedent section of a fuzzy rule \( R_k \) are connected by the ‘AND’ conjunctive, the operator \( \min \) is used to compute the aggregated rule fulfillment denoted as \( RF_k \).

\[
RF_k = \min_{i \in \{1, \ldots, n\}} \{ \mu_{\Pi_{\xi_i}}(x_i), \ldots, \mu_{\Pi_{\xi_i}}(x_i), \ldots, \mu_{\Pi_{\xi_i}}(x_i) \} \tag{30}
\]

where \( \mu_{\Pi_{\xi_i}} \) is the membership function of the \( j \)th fuzzy label of the \( i \)th input \( \Pi_{\xi_i} \) connecting to rule \( R_k \). Eq. (30) and Eqs. (20) and (21) perform the same function. Therefore, Step 3 of the inference process of CRI-FS is mapped onto layer 3 of the GenSoFNN-CRI(S) network. The aggregated rule fulfillment \( RF_k \) is presented as the output \( Z_{RF_k} \) of \( R_k \).

Step 4: Consequent derivation

In Step 4, the consequences of firing the fuzzy rules in the CRI-FS are determined. The label \( OL_{l,m} \) denotes the \( l \)th fuzzy label of the \( m \)th output. Hence, the inferred output for \( OL_{l,m} \) due to the firing of rule \( R_k \), denoted as \( OL_{l,m,k} \), is computed as follows.

\[
\mu_{\text{OL}_{l,m,k}}(y_m) = T(Z_{RF_k}^{(l,m)}), \mu_{\text{OL}_{l,m}}(y_m) \tag{31}
\]

where \( \mu_{\text{OL}_{l,m,k}} \) is the membership function of the inferred fuzzy output \( OL_{l,m,k} \) due to rule \( R_k \); \( \mu_{\text{OL}_{l,m}} \) is the membership function of the fuzzy label \( OL_{l,m} \); and \( Z_{RF_k}^{(l,m)} \) is the output of rule \( R_k \) that is connected to \( OL_{l,m} \). When the \( T \)-norm computation is resolved using the \( \min \) operator, Eq. (31) becomes

\[
\mu_{\text{OL}_{l,m,k}}(y_m) = \min(Z_{RF_k}^{(l,m)}, \mu_{\text{OL}_{l,m}}(y_m)), \quad \forall y_m \in U_m \tag{32}
\]

where \( U_m \) is the universe of discourse of output \( m \). Eq. (32) specifies the ‘clipping’ method widely used in fuzzy systems to infer the output consequents. Fig. 7 depicts this.

However, different fuzzy rules may share the same consequent \( OL_{l,m} \). Therefore, the inferred conclusion for \( OL_{l,m} \) varied for different fuzzy rules. Thus, some form of aggregation has to be performed prior to defuzzification. In the literature, two approaches are proposed: the Infer-First-Then-Aggregate (IFTA) and Aggregate-First-Then-Infer (AFTI) methods (Fuller, 1999). The former approach is adopted in the GenSoFNN-CRI(S) network as it best suit the computing nature of the hybrid structure. That is, compute the inferred output due to the individual rules and subsequently aggregates the effects of the various rules to derive the overall fuzzified output \( \tilde{OL}_{l,m} \):

\[
\tilde{OL}_{l,m} = \bigcup_{k \in \{1, \ldots, N_{l,m}\}} \tilde{OL}_{l,m,k} = \max_{k \in \{1, \ldots, N_{l,m}\}} (\mu_{\text{OL}_{l,m,k}}(y_m)) \tag{33}
\]

[Because \( \tilde{OL}_{l,m,k} \forall k \in \{1, \ldots, N_{l,m}\} \), are fuzzy sets]

\[
= \max_{k \in \{1, \ldots, N_{l,m}\}} (\min(Z_{RF_k}^{(l,m)}, \mu_{\text{OL}_{l,m}}(y_m)))
\]

\[
= \min_{k \in \{1, \ldots, N_{l,m}\}} \left( \max_{k \in \{1, \ldots, N_{l,m}\}} (Z_{RF_k}^{(l,m)}), \mu_{\text{OL}_{l,m}}(y_m) \right)
\]

[Where \( Z_{l,m} = \max_{k \in \{1, \ldots, N_{l,m}\}} (Z_{RF_k}^{(l,m)}) \) (33)]

where \( N_{l,m} \) is the total number of rules in GenSoFNN-CRI(S) having \( OL_{l,m} \) as consequent. Thus, the operations of Eq. (33) are essentially realized in the functions of the layer 4 nodes of the GenSoFNN-CRI(S) network using Eqs. (22) and (23).

Step 5: Defuzzification

The last step in the inference process of the CRI-FS system is to defuzzify the derived fuzzy conclusions and present them as crisp outputs. For each output \( y_m \), the derived fuzzy conclusions for all its output labels are aggregated using a COA technique to compute the final output. This technique is implemented in the GenSoFNN-CRI(S) network using Eqs. (24)–(26).

This concludes the step-by-step mapping of the operations of the GenSoFNN-CRI(S) network to the corresponding

![Fig. 7. The clipping method for fuzzy inference.](image-url)
inference steps of a CRI-based fuzzy system. Hence, the GenSoFNN-CRI(S) network has been demonstrated to be the neural implementation of the CRI-based fuzzy system. In addition, one can easily verify that Eq. (33) is computationally equivalent to Eq. (9) for the case of multiple inputs. This reinforces the notion that the operations of the GenSoFNN-CRI(S) network implement the inference steps of the CRI scheme and provides a strong fuzzy logic foundation to the functionality of the GenSoFNN-CRI(S), thus enhancing interpretability and tractability of its computations. Section 5 presents the learning equations of the BP algorithm for the parameter learning phase of the GenSoFNN-CRI(S) network.

5. Parameter learning of GenSoFNN-CRI(S)

The tuning of the fuzzy sets in layers 2 and 5 aimed to minimize the cost error function $CError$ defined in Eq. (34).

$$CError = \frac{1}{2} \sum_{m=1}^{n_5} (d_m - y_m)^2$$

(34)

where $d_m$ and $y_m$ are the $m$th desired and actual outputs, respectively.

The error $CError$ is computed with respect to each training pair $\{X(p), D(p)\}$, where $p \in \{1...P\}$. The variable $P$ denotes the number of training pairs in one training epoch and $X$ and $D$ are the input and desired output vectors, respectively. The incremental (online) approach is adopted in the BP based parameter-learning phase. That is, the parameters are updated after each data presentation. This learning paradigm is vital to the online parameter tuning of the GenSoFNN-CRI(S) network that is used to model complex and dynamic problem domains with changing characteristics. The error signals and the updating rules for the parameters (weights) starting from layer 5 to layer 2 of the GenSoFNN-CRI(S) network, derived with respect to the training data pair $\{X(p), D(p)\}$ and the training step $T$ (where $T = p$), are shown as follows. For the interested reader, please refer to (Tung, 2001) for a full derivation of the learning equations.

5.1. Layer 5 (output layer)

During the parameter-learning phase, the input $X(T)$ is presented to the GenSoFNN-CRI(S) network during the forward pass. A set of computed outputs $Y(T) = [y_1(T), ..., y_{n_5}(T), ..., y_{n_5}(T)]^T$ is obtained. During the backward pass, $Y(T)$ is compared against $D(T)$ and the error signals and the update equations for the parameters are derived. For the output node $OL_{lm}$, the error signal $\delta_{lm}(T)$ is defined as

$$\delta_{lm}(T) = -\frac{\partial CError(T)}{\partial y_{lm}(T)} = d_{lm}(T) - y_{lm}(T)$$

(35)

The learning (update) equations for the four parameters ($l_{lm}$, $u_{lm}$, $v_{lm}$, and $r_{lm}$) of its $l$th fuzzy label $OL_{lm}$ are similar and has the general form:

$$\text{Param}_{lm}(T + 1) = \text{Param}_{lm}(T) + \Delta \text{Param}_{lm}(T)$$

$$= \text{Param}_{lm}(T) + \eta \{d_{lm}(T) - y_{lm}(T)\} \left\{ \frac{Z_{lm}(T)}{\sum_{m=1}^{n_5} Z_{lm}(T)} \right\}$$

(36)

where $\text{Param}_{lm}$ denotes either $l_{lm}$, $u_{lm}$, $v_{lm}$ or $r_{lm}$; $\eta$ is the learning rate; and $Z_{lm}(T)$ is the output from node $OL_{lm}$ at time step $T$.

5.2. Layer 4 (consequent layer)

All the links in layer 4 have unity weights. Hence, only the error signal $\delta_{lm}(T)$ needs to be computed. For the output term node $OL_{lm}$, the error signal $\delta_{lm}(T)$ is

$$\delta_{lm}(T) = -\frac{\partial CError(T)}{\partial Z_{lm}(T)} = \delta_{lm}(T) \times \frac{\hat{m}_{lm}(T) - y_{lm}(T)}{\sum_{m=1}^{n_5} Z_{lm}(T)}$$

(37)

where $\hat{m}_{lm}(T)$ is the mid-point of the kernel of $OL_{lm}$; and $Z_{lm}(T)$ is the output from node $OL_{lm}$ at training step $T$.

5.3. Layer 3 (rule layer)

All the links in layer 3 have unity weights. For rule node $R_k$, the error signal $\delta_k$ is defined as

$$\delta_k(T) = -\frac{\partial CError(T)}{\partial Z_{R_k}(T)}$$

$$= \sum_{m=1}^{n_5} \left\{ \delta_{lm}(T) \times \frac{\partial Z_{lm}(T)}{\partial Z_{R_k}(T)} \right\}$$

(38)

where $Z_{R_k}(T)$ is the output from rule node $R_k$ at training step $T$; $Z_{lm}(T)$ is the output of the node $OL_{lm}$ that rule $R_k$ connects to at training step $T$; and $\delta_{lm}(T)$ is the error signal from node $OL_{lm}$ (that is connected to node $R_k$) and back-propagated to $R_k$ at training step $T$. For the CRI-mapped GenSoFNN-CRI(S) network,

$$\frac{\partial Z_{lm}(T)}{\partial \delta_k(T)} = \begin{cases} 1, & \text{if } Z_{lm}(T) = Z_{R_k}(T) \\ 0, & \text{Otherwise} \end{cases}$$

(39)

Hence, the error signal $\delta_k(T)$ of rule node $R_k$ may be interpreted as the aggregation (summation) of the error signals from layer 4 nodes to which $R_k$ provides the maximum input at time step $T$.

5.4. Layer 2 (antecedent layer)

Layer 2 contains the antecedent section of the fuzzy rules. For input fuzzy label $IL_{ij}$, the error signal $\delta_{ij}(T)$ is
defined as
\[
\delta_{i,j}(T) = \sum_{n=1}^{m} \left( \frac{1}{\tau_m} \sum_{i=1}^{\tau_m} \delta_{i,n}(T) \right)
\]
where \(\delta_{i,n}(T)\) is the error signal from the \(n\)th output term of the output node \(OV_n\) (denoted as \(OL_{i,n}\)) that \(IL_{i,j}\) contributes to through some rule node \(R_i\); and \(\tau_m\) is the total number of such term nodes that \(IL_{i,j}\) contributes to at layer 4 (consequent layer). The updating rules for the parameters of the trapezoidal-shaped fuzzy set of \(IL_{i,j}\) are defined as:
\[
l_{i,j}(T+1) = \begin{cases} 
  l_{i,j}(T) - \eta \delta_{i,j}(T) \frac{u_{i,j}(T) - x_i(T)}{[u_{i,j}(T) - l_{i,j}(T)]^2}, & \text{If } \delta_{i,j}(T) \neq 0 \text{ and } l_{i,j}(T) < x_i(T) < u_{i,j}(T) \\
  l_{i,j}(T), & \text{Otherwise}
\end{cases}
\]
\[
u_{i,j}(T+1) = \begin{cases} 
  v_{i,j}(T) + \eta \delta_{i,j}(T) \frac{r_{i,j}(T) - x_i(T)}{[r_{i,j}(T) - v_{i,j}(T)]^2}, & \text{If } \delta_{i,j}(T) \neq 0 \text{ and } v_{i,j}(T) < x_i(T) < r_{i,j}(T) \\
  v_{i,j}(T), & \text{Otherwise}
\end{cases}
\]
\[
u_{i,j}(T+1) = \begin{cases} 
  r_{i,j}(T) + \eta \delta_{i,j}(T) \frac{x_i(T) - v_{i,j}(T)}{[r_{i,j}(T) - v_{i,j}(T)]^2}, & \text{If } \delta_{i,j}(T) \neq 0 \text{ and } x_i(T) < r_{i,j}(T) \\
  r_{i,j}(T), & \text{Otherwise}
\end{cases}
\]
where \(l_{i,j}(T+1), u_{i,j}(T+1), \) and \(r_{i,j}(T+1)\) are the parameters of the fuzzy set of \(IL_{i,j}\) after the update; \(l_{i,j}(T), u_{i,j}(T), \) and \(r_{i,j}(T)\) are the present values at time step \(T\); \(x_i(T)\) is the \(i\)th input to the GenSoFNN-CRI(S) network at time step \(T\); and \(\eta\) is the learning rate.

6. Cox’s proportional hazards model

In simplified form, the hazard function of Cox’s proportional hazards model (Cole & Gunther, 1995; Lane et al., 1986) is given as
\[
h(t|z) = \exp(\beta z) h_0(t)
\]
where \(z\) is a vector of variables associated with the problem domain; and \(\beta\) is the corresponding vector of regression coefficients.

The baseline hazard rate \(h_0(t)\) in Eq. (45) can be interpreted as the hazard rate for an average bank, and the values of the variables are equal to the population means. As all the variables are fixed at the beginning, the ratio of the hazard rates of two banks with distinct values of \(z\) is a constant independent of time; that is, the hazard rates of the two banks are proportional as shown in Eq. (46).
\[
\frac{h(t|z_1)}{h(t|z_2)} = \frac{\exp(\beta z_1)}{\exp(\beta z_2)} = \exp[\beta(z_1 - z_2)]
\]

6.1. Financial variables (covariates)

The financial variables (covariates) used in the bank failure prediction application are extracted from the Call Reports, which are downloaded from the website of Federal Reserve Bank, Chicago (Chicago Bank). The expected impacts of the variables on bank failures are explained in Cheng (2002). Apart from loan loss provisions for the period (PLAQLY), all the variables have been found significant in past studies (Cole & Gunther, 1995; Lane et al., 1986; Martin, 1977; Sinkey, 1975). Normality plots of these variables indicate that the variables are not normally distributed. The statistical significance of the variables is investigated by best score selection, stepwise selection and purposeful selection (Cheng, 2002). Based on the findings of these selection procedures and an analysis of the correlations between the variables, only the nine variables listed in Appendix A are incorporated into the Cox’s model and the subsequent GenSoFNN-CRI(S) based EWS for bank failure classification and prediction.

7. Experimental results and analysis

This section presents the simulation results using the GenSoFNN-CRI(S) network as a bank failure classification and EWS based on the nine selected financial covariates.
introduced in Section 6. Three different sets of experiments are performed: (1) Bank failure classification using the financial covariates extracted from the last available financial statements and bank failure predictions using the same set of financial covariates but extracted from (2) 1 year, and (3) 2 years prior to the last available financial statements. The observation period of the survived (non-failing) banks consists of 21 years from January 1980 to December 2000 inclusively. For consistency, the data for the failed and survived banks have the same balance sheet dates. The parameters used in the GenSoFNN-CRI(S) network for all the three sets of experiments are as follows: back-propagation learning rate \( \eta \) is 0.5; plasticity and tendency parameters \( \beta \) and \( TD \) for DIC are set at 0.5; input \( MT \) and output \( MT \) are 0.6; \( STEP \) is 0.01; and \( SLOPE \) is one variance for each of the input/output dimensions

7.1. Bank failure classification using last available financial statements

The original data set has been preprocessed to filter out the last available financial statement for each of the banks during the observation period. For the failed banks, it would be the records prior to failure while the records for the surviving banks are those of year 2000 (last year of the observation period). From the filtered financial statements, nine variables (known as financial covariates) are extracted. These covariates (highlighted in Section 6 and Appendix A) are selected based on classical analytical study to determine their significance and expected impact on the financial health of the banking institutions. The interim data set consists of 702 failed banks (with failure dates spreading across the entire observation period) and 2933 banks that survived the observation period, leading to a total of 3635 observed banks. However, banks whose record has missing fields are removed leading to the final data set of 548 failed and 2555 survived (non-failing) banks. Hence, there are a total of 3103 observed banks. The failed banks constituted approximately 17.7% of the data set while the surviving banks made up the remaining 82.3%.

The data set is split into one training and one test set. The training set consists of 20% of the data set while the test set contains the remaining 80%. There are five cross-validation groups. The five cross-validation groups are denoted as CV1, CV2, CV3, CV4 and CV5, respectively. Each cross-validation group consists of training and test sets that are randomly generated. The data set is initially split into two pools: failed and survived (non-failing) banks. For each cross-validation group, 20% of both the two pools are randomly selected to form the training set. Hence, the number of survived banks is much greater than that of failed banks. This is termed an ‘unbalanced’ training scenario. The data in the training set is shuffled to randomize the presentation order. The training sets of the five cross-validation groups are mutually exclusive. One output is used to differentiate between failed and survived banks. Failed banks are denoted with output ‘0’ while survived (non-failing) banks are identified by output ‘1’. The GenSoFNN-CRI(S) network is subsequently used to model the inherent relationships between the financial covariates and their impact on the financial solvency of the respective banks.

The GenSoFNN-CRI(S) network is trained using the data pairs in the training set and the modeling capability of the trained network is subsequently evaluated using the test set. The simulation is repeated for all the five cross-validation groups. The classification threshold (to discern between failed and survived (non-failing) banks based on the nine input financial covariates) is varied to obtain the receiver-operating-characteristic (ROC) curves depicted in Fig. 8. Type I error is defined as the error of classifying a failed bank as a survived (non-failing) one whereas Type II error is the classification of a non-failing bank as a failed bank. Fig. 8 illustrates the error rates for both Type I and Type II errors expressed in percentage. The EER line denotes the case of equal error rates (EER), where the Type I equals Type II errors.

As one can easily observe from Fig. 8, Type I error rate is greater than Type II error rate for the ‘unbalanced’ training scenario given the optimal settings of the classification threshold. For bank failure classification, the relative cost of wrongly classifying a failed bank as a survived bank is much higher than that of classifying a non-failing bank as a failed one. Thus, it is highly desirable to minimize Type I error instead of Type II error. A plausible reason for the higher Type I error rate than Type II error rate can be attributed to the overwhelming effect of survived (non-failing) banks over failed banks in the training sets of the five cross-validation groups. Hence, the GenSoFNN-CRI(S) network is trained to be more sensitive to the traits of survived (non-failing) banks than failed banks. Thus, the higher classification rates for survived banks over failed banks. The simulation is subsequently repeated with a ‘balanced’
training scenario. The training sets of the five cross-validation groups are modified by randomly pruning away redundant survived banks until the number of survived and failed banks is equal. The test sets of the cross-validation groups remain the same as before. The ROC curves of the newly trained GenSoFNN-CRI(S) network with the balanced training scenario are depicted as Fig. 9.

The bank failure classification results of the GenSoFNN-CRI(S) network trained with the balanced training sets displayed a smaller Type I error rate than Type II error rate as evidenced by Fig. 9. Hence, Figs. 8 and 9 demonstrated that the GenSoFNN-CRI(S) network based bank failure classification and EWS (prediction) should be trained with balanced training sets consisting of equal number of failed and survived (non-failing) banks. This reflects a consideration in the training of the GenSoFNN-CRI(S) network and most neural fuzzy models in general. That is, the ability to discern closely similar classes partly depends on the ratio of these classes in the training set. This corresponds closely to the human cognitive process, as one tend to relate better to often-encountered situations.

7.2. Bank failure prediction using financial statements one year prior to the last record

To implement the GenSoFNN-CRI(S) based bank failure classification model as an EWS, financial statements obtained one year prior to the last available records are used to train the network. Thus, the trained GenSoFNN-CRI(S) network can be used as a prediction (early warning) system by forecasting the financial health of the banking institutions one year in advance using financial covariates extracted from the financial statements. The experimental setup and generation of the training and test sets of the various cross-validation groups are similar to that of Section 7.1.

Fig. 10 presents the prediction performances of the GenSoFNN-CRI(S) network for the forecasting of the financial health of the banks using financial covariates extracted from statements one year prior to the last record. Again, Fig. 10 has clearly shown that Type I error can be minimized in the presence of Type II error by using balanced training sets. Comparing Figs. 8–10, one can conclude that the classification rates of the GenSoFNN-CRI(S) network deteriorates from using financial covariates extracted from statements one year prior to the last record. This is expected since noise and uncertainty sets in when the prediction period becomes longer.

7.3. Bank failure prediction using financial statements two years prior to the last record

To evaluate the prediction accuracy and robustness of the GenSoFNN-CRI(S) network as an EWS for prediction of bank failures, the experiment is repeated using financial
statements two years prior to the last record. For failed banks, the records are obtained two years prior to the failure year and for the survived (non-failing) banks, the records are obtained from the year 1998. Fig. 11 illustrates the ROC curves obtained from the GenSoFNN-CRI(S) based EWS trained using both the unbalanced and balanced training sets.

Similar to what is observed in the previous results, the balanced training sets minimized Type I error in place of Type II error. In addition, comparing the classification results computed using the last financial statements (Figs. 8 and 9) and one year prior to that (Fig. 10), it is noticed that the classification accuracy further deteriorates as the prediction period is two years in advance. Analyzing only the classification and prediction results based on balanced training sets, one can observe that the average classification rate for the failed banks using the last financial statements is about 93% (Fig. 9) and it deteriorates to around 85% with statements obtained one year prior to the last record (Fig. 10(b)) and subsequently to about 75% with financial statements two years prior to the last available record (Fig. 11(b)). Fig. 12 summarizes the bank failure and prediction results derived using the GenSoFNN-CRI(S) network.

A more detailed analysis of the bank failure classification and prediction results of the GenSoFNN-CRI(S) network can be performed by evaluating the EER of the ROC curves. The EER values of the ROC curves for the various cross-validation groups (CV1–CV5) based on the six different sets of simulations performed (Figs. 8–11) are extracted and tabulated as Table 1.

The EER values give an indication of the accuracy of the simulation results. Thus, Fig. 12 and Table 1 reinforced the observation between the failed bank classification rate and the prediction period. That is, the longer the prediction period, the less accurate is the classification and prediction result. In addition, the balanced training scenarios are demonstrated to be more effective in the modeling and prediction of banking failures, as shown by the mean EER values.

7.4. Performance benchmark

The bank failure classification and prediction performances of the GenSoFNN-CRI(S) network (based on balanced training sets extracted from the last available financial statements) are subsequently benchmarked against that of the traditional Cox’s proportional hazards model (Cole & Gunther, 1995; Lane et al., 1986) as shown by Table 2.

The detailed results of the Cox’s model are reported in Cheng (2002). For the Cox’s model, the range of relative misclassification cost considered is 1:1 (equal misclassification costs) and with an increment of 5 to 30:1 where the cost of misclassifying a failed bank (Type I error) is 30
times higher than that of misclassifying a survived bank (Type II error). The optimal cut-off point used in the Cox’s model, which is used to distinguish failed banks from survived banks, is chosen such that the total probability of misclassification is minimized (Afifi & Clark, 1996). Such an optimal cut-off point may be associated to the classification threshold used in the GenSoFNN-CRI(S) network based models. It must be noted that the cut-off points used in the comparison (Table 2) have been optimized for the Cox’s model. In addition, the relative misclassification costs of the Cox’s model do not apply to the GenSoFNN-CRI(S) models.

From Table 2, one can conclude that the GenSoFNN-CRI(S) network based classification and prediction system has a superior performance over that of the Cox’s model in minimizing Type I error. However, the GenSoFNN-CRI(S) network has a much higher Type II error rate than the Cox’s model albeit being more consistent and robust in the classification of survived banks. That is, the change in Type II error rate for the GenSoFNN-CRI(S) network is much smaller than that of the Cox’s model.

The benchmarking process is subsequently repeated using a back-propagation trained MLP network (Lin & Lee, 1996) with a 9-10-1 structure that has been empirically determined to give the optimal results and the Modified Cerebellar Model Articulation Controller (MCMAC) (Ang & Quek, 2000; Ng & Quek, 1996) that acts as an adaptive look-up table. The bank failure classification results for these two benchmarking architectures and the GenSoFNN-CRI(S) network using the financial covariates extracted from the last financial statements are depicted as Fig. 13.

From Fig. 13, one can observed that the MLP neural network has superior performance to both the MCMAC structure and the GenSoFNN-CRI(S) network. However, it is common knowledge that the MLP functions as a black box and knowledge solicitation from its trained structure is nearly impossible. Although it proves to be a good classifier of failed and survived (non-failing) banks, the MLP network

Table 1
EER readings extracted from the ROC curves derived using the GenSoFNN-CRI(S) network

<table>
<thead>
<tr>
<th>Scenario</th>
<th>CV1 (%)</th>
<th>CV2 (%)</th>
<th>CV3 (%)</th>
<th>CV4 (%)</th>
<th>CV5 (%)</th>
<th>Mean EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last record (balanced)</td>
<td>7.08</td>
<td>9.38</td>
<td>5.63</td>
<td>13.96</td>
<td>4.79</td>
<td>8.17</td>
</tr>
<tr>
<td>Last record (unbalanced)</td>
<td>4.40</td>
<td>10.99</td>
<td>5.49</td>
<td>15.60</td>
<td>16.48</td>
<td>10.59</td>
</tr>
<tr>
<td>1 Year prior (balanced)</td>
<td>17.74</td>
<td>18.87</td>
<td>12.45</td>
<td>23.40</td>
<td>13.21</td>
<td>17.13</td>
</tr>
<tr>
<td>1 Year prior (unbalanced)</td>
<td>27.78</td>
<td>27.41</td>
<td>28.89</td>
<td>35.19</td>
<td>25.19</td>
<td>28.89</td>
</tr>
<tr>
<td>2 Year prior (balanced)</td>
<td>31.71</td>
<td>31.71</td>
<td>18.05</td>
<td>25.85</td>
<td>31.71</td>
<td>27.81</td>
</tr>
<tr>
<td>2 Year prior (unbalanced)</td>
<td>36.50</td>
<td>27.50</td>
<td>43.50</td>
<td>40.00</td>
<td>33.00</td>
<td>36.10</td>
</tr>
</tbody>
</table>

Table 2
Comparison of classification performance between the Cox’s model and the GenSoFNN-CRI(S) network to differentiate between failed and survived banks

<table>
<thead>
<tr>
<th>Relative misclassification cost: failed vs. survived banks</th>
<th>Optimal survived threshold to classify failed and survived banks</th>
<th>Type I error (%)</th>
<th>Type II error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cox</td>
<td>GenSoFNN-CRI(S)</td>
<td>Cox</td>
</tr>
<tr>
<td>1:1</td>
<td>0.73316</td>
<td>54.00</td>
<td>0.69</td>
</tr>
<tr>
<td>5:1</td>
<td>0.76949</td>
<td>49.50</td>
<td>0.23</td>
</tr>
<tr>
<td>10:1</td>
<td>0.81490</td>
<td>44.00</td>
<td>0.00</td>
</tr>
<tr>
<td>15:1</td>
<td>0.86031</td>
<td>36.50</td>
<td>0.00</td>
</tr>
<tr>
<td>20:1</td>
<td>0.90571</td>
<td>29.00</td>
<td>0.00</td>
</tr>
<tr>
<td>25:1</td>
<td>0.95112</td>
<td>18.50</td>
<td>0.00</td>
</tr>
<tr>
<td>30:1</td>
<td>0.99863</td>
<td>6.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

a Note: The Cox’s model has the ability to assign different costs of misclassification for failed and survived banks. This does not apply to the GenSoFNN-CRI(S) network based models.

![Fig. 13](image-url) Bank failure classification results based on the financial covariates extracted from the last available financial statement using a 9-10-1 structured MLP, the modified CMAC structure and the GenSoFNN-CRI(S) network.
fails to provide any insight into the possible traits of financial distress that lead to banking failure. Meanwhile, the MCMAC structure seems to have a slightly better classification performance than the GenSoFNN-CRI(S) network. This is because it is a local generalization model while the GenSoFNN-CRI(S) network is a global generalization model which is more sensitive to contradicting/highly similar training data pairs. However, because it is essentially a look-up table and due to the huge number of computing cells in its trained structure, it is difficult to extract any form of linguistic IF–THEN fuzzy rules from the MCMAC structure. For this simulation, the MCMAC requires a structure of at least 500 computing cells. In comparison, the GenSoFNN-CRI(S) network generates an average of

Fig. 14. Fuzzy sets for the nine selected financial covariates computed by the GenSoFNN-CRI(S) network using the balanced training set of cross-validation group CV1.
around 50 fuzzy rules in order to model the underlying characteristics of the training data pairs.

7.5. Fuzzy rules analysis

In addition to better classification and prediction rates over the Cox’s model (which are critical for an EWS for the detection of bank failures), the GenSoFNN-CRI(S) system formulates an intuitive set of IF–THEN fuzzy rules from the numerical training data. Such rules describe the inherent interactions between the nine selected financial covariates and their impact on the financial health of the observed banks. Thus, important insights into the possible traits of financial distress that leads to eventual bank failure can be obtained by examining the formulated fuzzy rules of the GenSoFNN-CRI(S) based EWS. In comparison, the Cox’s model, the MLP and MCMAC functioned as black boxes as it is difficult to interpret their classification decisions. To show the intuitiveness of the formulated fuzzy rules, the fuzzy sets of the nine selected financial covariates computed by the GenSoFNN-CRI(S) network using the balanced training set of cross-validation group CV1 are depicted as Fig. 14.

Semantic labels such as Low, Medium and High are attached to the respective fuzzy sets to extract the formulated fuzzy rules from the trained structure of the GenSoFNN-CRI(S) network. Ten of the most fired rules used to discern failed and non-failing (survived) banks are listed as Table 3. The fuzzy rules are extracted from the trained structure of the GenSoFNN-CRI(S) network simply by tracing the linkages and connections between the layer 2 nodes (input fuzzy terms), the layer 3 nodes (rule nodes) and the layer 4 nodes (output fuzzy terms). Since the GenSoFNN-CRI(S) network is employed as a bank failure classification and prediction EWS, positive rules referred to the rules used to classify an input bank as a failed/failing bank while negative rules denote the fuzzy rules that classify the input bank as one that is not failing (survived) bank. Table 3 lists five of the positive (negative) rules with the highest firing strength/frequency amongst the set of positive (negative) fuzzy rules. As one can easily observed, the five best-fired rules from

<table>
<thead>
<tr>
<th>Positive rules</th>
<th>Firing strength (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1: IF ROE is Low THEN input bank is Failed Bank</td>
<td>8.73</td>
</tr>
<tr>
<td>Rule 2: IF PLAQLY is Medium THEN input bank is Failed Bank</td>
<td>7.17</td>
</tr>
<tr>
<td>Rule 3: IF CAPADE is Low THEN input bank is Failed Bank</td>
<td>6.35</td>
</tr>
<tr>
<td>Rule 4: IF CAPADE is Low and ROE is Low THEN input bank is Failed Bank</td>
<td>5.87</td>
</tr>
<tr>
<td>Rule 5: IF CAPADE is Low and PLAQLY is Medium and ROE is Low THEN input bank is Failed Bank</td>
<td>5.78</td>
</tr>
<tr>
<td>Total</td>
<td><strong>33.9</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Negative rules</th>
<th>Firing strength (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1: IF CAPADE is High THEN input bank is not Failed Bank</td>
<td>8.88</td>
</tr>
<tr>
<td>Rule 2: IF CAPADE is Medium and OLAQLY is Low and PROBLO is Low and PLAQLY is Low and NIEOIN is Marginal Low and NINMAR is Marginal Low and ROE is Medium and LIQUID is Low and GROWLA is Marginal Low THEN input bank is not Failed Bank</td>
<td>8.21</td>
</tr>
<tr>
<td>Rule 3: IF CAPADE is Medium and OLAQLY is Marginal Low and PROBLO is Low and PLAQLY is Low and NIEOIN is Marginal Low and NINMAR is Marginal Low and ROE is Medium and LIQUID is Low and GROWLA is Marginal Low THEN input bank is not Failed Bank</td>
<td>6.94</td>
</tr>
<tr>
<td>Rule 4: IF CAPADE is Medium and OLAQLY is Low and PROBLO is Low and PLAQLY is Low and NIEOIN is Low and NINMAR is Marginal Low and ROE is Medium and LIQUID is Low and GROWLA is Marginal Low THEN input bank is not Failed Bank</td>
<td>6.91</td>
</tr>
<tr>
<td>Rule 5: IF CAPADE is Marginal Low and OLAQLY is Low and PROBLO is Low and PLAQLY is Low and NIEOIN is Marginal Low and NINMAR is Marginal Low and ROE is Medium and LIQUID is Low and GROWLA is Marginal Low THEN input bank is not Failed Bank</td>
<td>6.67</td>
</tr>
<tr>
<td>Total</td>
<td><strong>37.61</strong></td>
</tr>
</tbody>
</table>
failure analysis. The GenSoFNN network is a generic approach named GenSoFNN (Tung & Quek, 2002c) to bank failure. This paper attempts to apply a novel neural fuzzy system to the study of bank failure. It is difficult to explicitly specify what constitutes a financial distress and the intrinsic relationship between financial distress and a failed bank. This paper attempts to apply a novel neural fuzzy system named GenSoFNN (Tung & Quek, 2002c) to bank failure analysis. The GenSoFNN network is a generic network whose nodal operations are defined by the fuzzy inference scheme adopted by the network. In this paper, the CRI (Zadeh, 1975) scheme is mapped onto the connectionist structure of the GenSoFNN network to synthesize the GenSoFNN-CRI(S) network. The trained GenSoFNN-CRI(S) network serves as a bank failure classification and prediction system and the formulated fuzzy rule-base sheds light on the inherent contributions of the selected financial covariates to bank failure. Experiments have demonstrated that the GenSoFNN-CRI(S) network consistently outperforms the Cox’s model in classifying failed and survived banks using a set of US banking data (Chicago Bank). Although it has been subsequently shown that the MLP network has a superior performance and the MCMAC has a slightly better performance over the GenSoFNN-CRI(S) network, the strengths of the GenSoFNN-CRI(S) network lies in its ability to formulate a set of intuitive IF–THEN fuzzy rules while the other two architectures function as black boxes.

Currently, extensive efforts have been invested at the Intelligent Systems Laboratory (ISL) (ISL Online), School of Computer Engineering, Nanyang Technological University (Singapore), to further improve the classification rates and reduce the error rates of the GenSoFNN-CRI(S) based bank failure classification and prediction system. The focus of the research is on enhancing the bank failure prediction capability of the GenSoFNN-CRI(S) network for it to serve as an efficient EWS. Furthermore, research to develop a combined strategy for the reconstruction of missing financial data (which is often encountered in the study of bank failure) and prediction models focuses on understanding the reasons why banks failed and therefore identify the symptoms of financial distress a bank experiences prior to its failure. In addition, banking analysts may examine hypothetical scenarios by modifying the fuzzy quantifiers to the prediction system. This is also under current investigation at the ISL, which undertakes the investigation and development of advanced hybrid neural fuzzy architectures such as (Ang et al., 2003; Quek & Zhou, 1996, 1999; Tung & Quek, 2002b,c) for the modeling of complex, dynamic and non-linear systems. These techniques have been successfully applied to numerous novel applications such as automated driving (Pasquier, Quek, & Toh, 2001), signature forgery detection (Quek & Zhou, 2002), gear control for continuous variable transmission (CVT) system in automobile (Ang, Quek, & Wahab, 2001), and fingerprint verification (Quek, Tan, & Sagar, 2001).

8. Conclusions

Many statistical models such as the Cox’s model (Cole & Gunther, 1995; Lane et al., 1986) have been applied to the study of bank failure. However, these classical models have not attempted to identify the possible traits of financial distress that eventually leads to bank failure. It is difficult to explicitly specify what constitutes a financial distress and the intrinsic relationship between financial distress and a failed bank. This paper attempts to apply a novel neural fuzzy system named GenSoFNN (Tung & Quek, 2002c) to bank failure analysis. The GenSoFNN network is a generic approach whose nodal operations are defined by the fuzzy inference scheme adopted by the network. In this paper, the CRI (Zadeh, 1975) scheme is mapped onto the connectionist structure of the GenSoFNN network to synthesize the GenSoFNN-CRI(S) network. The trained GenSoFNN-CRI(S) network serves as a bank failure classification and prediction system and the formulated fuzzy rule-base sheds light on the inherent contributions of the selected financial covariates to bank failure. Experiments have demonstrated that the GenSoFNN-CRI(S) network consistently outperforms the Cox’s model in classifying failed and survived banks using a set of US banking data (Chicago Bank). Although it has been subsequently shown that the MLP network has a superior performance and the MCMAC has a slightly better performance over the GenSoFNN-CRI(S) network, the strengths of the GenSoFNN-CRI(S) network lies in its ability to formulate a set of intuitive IF–THEN fuzzy rules while the other two architectures function as black boxes.

Currently, extensive efforts have been invested at the Intelligent Systems Laboratory (ISL) (ISL Online), School of Computer Engineering, Nanyang Technological University (Singapore), to further improve the classification rates and reduce the error rates of the GenSoFNN-CRI(S) based bank failure classification and prediction system. The focus of the research is on enhancing the bank failure prediction capability of the GenSoFNN-CRI(S) network for it to serve as an efficient EWS. Furthermore, research to develop a combined strategy for the reconstruction of missing financial data (which is often encountered in the study of bank failure) and prediction models focuses on understanding the reasons why banks failed and therefore identify the symptoms of financial distress a bank experiences prior to its failure. In addition, banking analysts may examine hypothetical scenarios by modifying the fuzzy quantifiers to the prediction system. This is also under current investigation at the ISL, which undertakes the investigation and development of advanced hybrid neural fuzzy architectures such as (Ang et al., 2003; Quek & Zhou, 1996, 1999; Tung & Quek, 2002b,c) for the modeling of complex, dynamic and non-linear systems. These techniques have been successfully applied to numerous novel applications such as automated driving (Pasquier, Quek, & Toh, 2001), signature forgery detection (Quek & Zhou, 2002), gear control for continuous variable transmission (CVT) system in automobile (Ang, Quek, & Wahab, 2001), and fingerprint verification (Quek, Tan, & Sagar, 2001).

Appendix

See Table A1.
Table A1
Definition of covariates and their expected impact on failure (Numbers in brackets are the identification of the data elements from the Call Reports)

<table>
<thead>
<tr>
<th>CAMEL category</th>
<th>Covariates</th>
<th>Expected impact on failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital adequacy</td>
<td><strong>CAPADE</strong>&lt;br&gt;Average total equity capital (3210)/average total assets (2170) &lt;br&gt;(higher is the ratio, greater is the capacity to absorb losses, smaller is the probability of failure)</td>
<td>− ve</td>
</tr>
<tr>
<td>Asset (loan) quality</td>
<td><strong>OLAQLY</strong>&lt;br&gt;Average (accumulated) loan loss allowance (3123)/average total loans and leases, gross (1400) &lt;br&gt;(smaller is the ratio, better is the loan quality, smaller is the probability of failure)</td>
<td>− ve</td>
</tr>
<tr>
<td></td>
<td><strong>PROBLO</strong>&lt;br&gt;Average (accumulated) loans 90 + days late (1407)/average total loans and leases, gross (1400) &lt;br&gt;(higher is the ratio, poorer is the loan quality, higher is the probability of failure)</td>
<td>+ ve</td>
</tr>
<tr>
<td></td>
<td><strong>PLAQLY</strong>&lt;br&gt;(Annual) loan loss provisions (4230)/average total loans and leases, gross (1400) &lt;br&gt;(higher is the ratio, poorer is the loan quality expected to be, higher is the probability of failure)</td>
<td>+ ve</td>
</tr>
<tr>
<td>Management</td>
<td><strong>NIEOIN</strong>&lt;br&gt;Non-interest expense (4093)/operating income (4000) &lt;br&gt;(higher is the ratio, less operationally efficient and profitable is the bank, higher is the probability of failure)</td>
<td>+ ve</td>
</tr>
<tr>
<td>Earnings</td>
<td><strong>NINMAR</strong>&lt;br&gt;Total interest income (4107) − interest expense (4073)/average total assets (2170) &lt;br&gt;(higher is the net interest margin, more profitable is the bank, smaller is the probability of failure)</td>
<td>− ve</td>
</tr>
<tr>
<td></td>
<td><strong>ROE</strong>&lt;br&gt;Net income (after tax) (4340) + applicable income taxes (4302)/average total equity capital (3210) &lt;br&gt;(higher is return on equity before tax, smaller is the probability of failure)</td>
<td>− ve</td>
</tr>
<tr>
<td>Liquidity</td>
<td><strong>LIQUID</strong>&lt;br&gt;Average cash (0010) + average federal funds sold (1350)/average total deposits (2200) + average fed funds purchased (2800) + average banks’ liability on acceptances (2920) + average other liabilities (2930) &lt;br&gt;(higher liquidity indicates inefficient utilization of resources; it can also reflect an expectation of unfavourable events (runs on deposits for example). Overall, higher liquidity suggests a higher probability of failure)</td>
<td>+ ve</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td><strong>GROWLA</strong>&lt;br&gt;Total loans and leases, gross (1400), − total loans and leases, gross (1400), −&lt;br&gt;− 1/total loans and leases, gross (1400), −&lt;br&gt;− 1 (with appropriate credit control and adequate loan loss provisions, a bank with higher loan growth rate would have better profitability and smaller probability of failure)</td>
<td>− ve</td>
</tr>
</tbody>
</table>

References


