ELSEVIER

European Journal of Operational Research 109 (1998) 390–402

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

A hybrid intelligent system for predicting bank holding structures

R.R. Hashemi [b], L.A. Le Blanc [a,*], C.T. Rucks [b], A. Rajaratnam [b]

[a] *Department of Management, College of Business Administration, University of Arkansas, Little Rock, 2801 South University Avenue, Little Rock, AR 72204-1099, USA*

[b] *University of Arkansas, Little Rock, AR 72204-1099, USA*

Received 1 March 1997

Abstract

A composite model of neural network and rough sets components was constructed to predict a sample of bank holding patterns. The final model was able to correctly classify 96% of a testing set of four types of bank holding structures. Holding structure is defined as the number of banks under common ownership. For this study, forms of bank holding structure include: banks that are not owned by another company, single banks that are held by another firm, pairs of banks that are held by another enterprise, and three or more banks that are held by another company. Initially, input to the neural network model was 28 financial ratios for more than 200 banks in Arkansas for 1992. The 28 ratios are organized by categories such as liquidity, credit risk, leverage, efficiency, and profitability. The ratios were constructed with 70 bank variables such as net worth, deposits, total assets, net loans, total operating income, etc. The first neural network model correctly classified 84% of the testing set at a tolerance level of 0.20. Another artificial intelligence (AI) procedure known as two-dimensional rough sets was then applied to the dataset. Rough sets reduced the number of input variables from 28 to 18, a drop of 36% in the number of input variables. This version of rough sets also eliminated a number of records, thereby reducing the information system (i.e., matrix) on both vertical and horizontal dimensions. A second neural network was trained with the reduced number of input variables and records. This network correctly classified 96% of the testing set at a tolerance level of 0.20, an increase of 11% in the accuracy of the prediction. By applying two-dimensional reducts to the dataset of financial ratios, the predictive accuracy of the neural network model was improved substantially. Banking institutions that are prime candidates for mergers or acquisitions can then be more accurately identified through the use of this hybrid decision support system (DSS) which combines different types of AI techniques for the purposes of data management and modeling. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Finance; Classification; Decision support systems; Neural networks; Rough sets

1. Introduction

Financial applications of neural networks typically focus on pattern matching, classification and forecasting. These functions includes mortgage

* Corresponding author. Fax: +1 501 569 8898; e-mail: laleblanc@ualr.edu.

0377-2217/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved.
PII S0377-2217(98)00065-4

underwriting judgments (Collins et al., 1988), credit card fraud detection (Rochester, 1990), prediction of corporate bond ratings (Dutta and Shekhar, 1988), and forecasting credit card risk from customer applications (Trippi and Turban, 1989). Salchenberger et al. (1992) employed a neural network model that processed input data of financial ratios to discriminate between healthy and failing financial institutions. A similar studies with neural network for forecasting bankruptcy was conducted by Messier and Hansen (1988) and Fletcher and Goss (1993). Tam and Kiang (1992) employed neural networks to predict cases of bank failure in Texas.

A potential application for neural networks is to identify firms that are good candidates for merger and acquisition, especially financial institutions such as banks. In the current competitive environment in the US banking industry, low cost provision of services is critical and size (i.e., economics of scale) is one approach or strategy to achieve low costs. However, appropriate candidates for merger or acquisition may be difficult to identify. Bank performance is measured to a large degree by ratios of financial condition (Staats, 1988). Bank or thrift institutions with interest in merger or acquisition could select such targets with the use of a predictive tool such as a neural network, assuming accurate performance in identifying appropriate marks.

Although neural network models are being developed in a wide variety of disciplines including finance, there has been no published report investigating their potential use in predicting merger or acquisition targets, including banks and thrift institution. In this study, a neural network is employed to identify possible bank merger targets. A second artificial intelligence (AI) tool, i.e., rough sets, is employed to restructure the input variables to improve the output and performance of the neural network.

1.1. Integration of artificial intelligence and decision support systems

In this paper, the neural network functions as the model component of a decision support systems (DSS) (see Fig. 1). A machine learning technique, rough sets, provides AI assistance to the data management component (Turban and Watkins, 1986) and may improve the operation and maintenance of the data base. Jarke and Vassiliou (1984) incorporated an AI technique within the data management component of a DSS. An AI technique in the data management component may increase the performance of the model component as well as the entire DSS.

For DSS, using sorting rules developed by rough sets may lead to a "burdensome" situation where a record does not match any of the sorting or classification rules (Slowinski and Zopounidis, 1995). On the other hand, the neural network approach classifies every record by its weighting mechanism, although in a "black box" fashion. The hybrid model described herein incorporates the best features of both AI techniques: rough sets to reduce the information system (i.e., decision table) and the neural network for pattern recognition. This is the rationale for *not* constructing the classification system based on rough sets only.

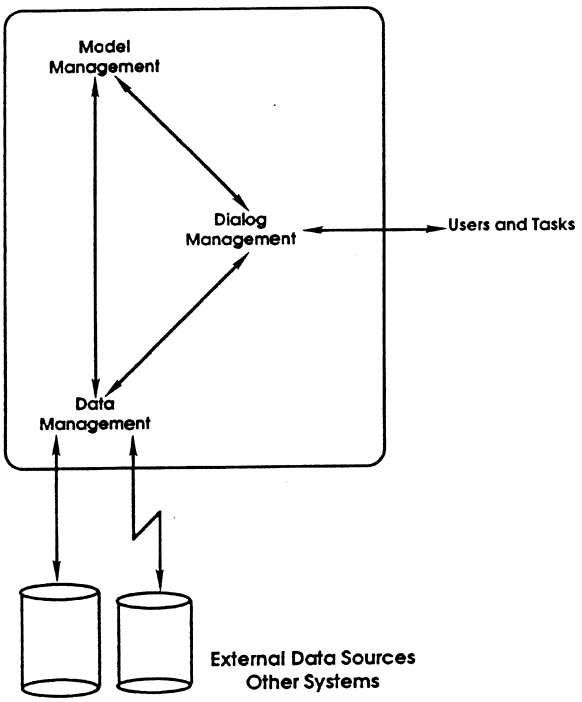

Fig. 1. Functional components of a decision support system.

The rough set pre-processing of data for neural networks has been proposed and analyzed in Jelonek et al. (1995). However, the results as reported were not promising. The rough set analysis employed by these authors used only attribute reduction (i.e., horizontal reduction). In analyzing five different data sets, misclassification rates increased in four of the sets as a result of attribute reduction for these neural networks. The increase in misclassification ranged from +2.03% to +11.06%. After pre-processing of data with rough sets in the hybrid DSS presented herein, the misclassification rate dropped 8.3% and 12.5%, respectively, for the neural network at two levels of tolerance and decreased 20% for a statistical model that also employed the reduced information system. The rough sets procedure and analysis reported herein reduces the information system (i.e., decision table) on both horizontal and vertical dimensions. In other words, the rough sets removes not only attributes that are unrelated to the decision outcome but also eliminates redundant records.

Initially, a reduct was only a vertical reduction of a decision table and there could be several reducts. No label exists for a decision table that is reduced both horizontally (like traditional rough sets) as well as reduced vertically (i.e., the elimination of redundant records). This two-dimensional reduct (2D-Reduct) introduced here is different from the usual definition of reduct as popularized by Pawlak (1982).

2. Background on neural networks

A brief description of the neural network model and the back-propagation algorithm is presented here. More detail may be found in Hertz et al. (1991) and Wasserman (1989).

2.1. Neural network architecture

A neural network is composed of a set of artificial neurons (nodes) grouped in a number of layers. The first layer and the last layer within a neural network are called input and output layers, respectively. The inner layers are known as hidden layers. The output of each neuron in a given layer (except the output layer) is fed as an input to every artificial neuron of the next layer. The input to the input layer is not the output of any layer. See Fig. 2.

Node N_i in layer L_n has four properties. These properties are: (1) an *input vector*, $I_i = [i_1, \ldots, i_k]$; (2) an *output*, a_i; (3) an *activation function*, f; and, (4) a learning *rate*, μ_i. To each element of the input vector, a weight is associated which collectively makes a *weight vector*, $V_i = [v_1, \ldots, v_k]$. The inner product of $I_i \cdot V_i = s_i = \sum i_j v_j$ (for $j = 1, \ldots, k$) represents the total weighted inputs (signals) received by node N_i. The activation function f determines the level of excitation for the node N_i. The same activation function is used for

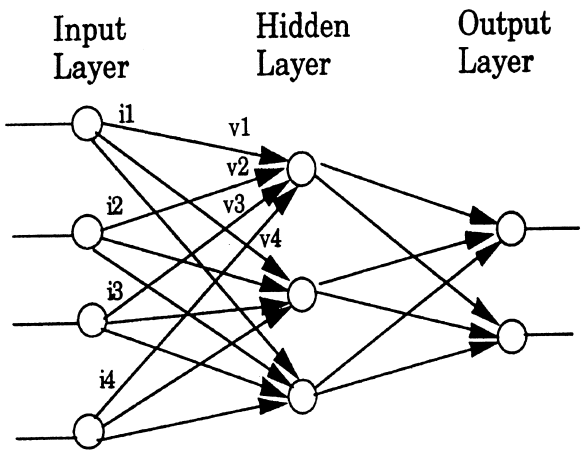

Fig. 2. Configuration of a neural network.

all the nodes of a neural network. The output a_i equals $f(I_i \cdot V_i)$. When the activation functions is a sigmoid function, then

$$a_i = f(s_i) = \frac{1}{1 + e^{-s_i}}. \quad (1)$$

The learning rate is a coefficient ($0 < \mu_i < 1$) that will be used in training of the node N_i. It may be the same for all the nodes within a neural network.

For the nodes $N_1, \ldots, N_j$ in layer L_n, the weight vectors of $V_1, \ldots, V_j$ related to input vectors of $I_1, \ldots, I_j$ make a matrix of J columns and K rows, W_n. The output for the layer L_n is a vector $A_n = [a_1, \ldots, a_j]$; and it is calculated as follows:

$$A_n = f(I_n \cdot W_n). \quad (2)$$

Since all the nodes in the layer L_n have the same input vectors, then $I_n = I_i$ (although the input vectors for all the nodes of layer L_n are the same but their associated weight vectors are not). If the neural network is composed of $L_1, \ldots, L_m$ layers, then the output for each layer is calculated according to Wasserman (1989). The output of layer L_i is fed as input to the next layer, L_{i+1}. This process continues until the final output vector is produced. The process of taking an input and sending it through all the layers of a neural network to generate the final output vector is referred to as a *forward pass*.

2.2. Neural network training

Using a training paradigm, a neural network may be trained to generate the desired output vector for each individual input vector of a training data set. The training paradigm of interest is the *backpropagation* algorithm. For a given input vector, it generates the output vector by a forward pass. Then, the difference between the output vector and the target (i.e., desired) output vector is backpropagated through the neural network (from the output layer to the input layer) to modify the weight matrices for the entire neural network. This process is referred to as a *reverse pass*.

In the reverse pass, the training of the neural network takes place. Training the network means that all the weights in the weight matrices will be modified based on the Δ rule (Grossberg, 1974). For example, the weight v_j associated with input i_j for the node N_j in the output layer is changing to a new weight v_j as follows:

$$v_j = v_j + \Delta_j. \quad (3)$$

The value of Δ_j is calculated using

$$\Delta_j = \mu_j i_j a_j (1 - a_j)(t_j - a_j), \quad (4)$$

where μ_j is the learning rate for the node N_j, i_j an input to the node N_j (the weight associated to this input will be modified), a_j the output, and t_j the target output for node N_j.

When the difference between the output and the target output for node N_j (noted by σ_j) is less than or equal to a *tolerance level*, then node N_j has learned the input pattern. The tolerance level is a permissible range of deviation of the calculated output a_j from the target output, t_j. The weight matrices related to the input vectors of the nodes in the hidden layers are also modified according to Eqs. (3) and (4). Since a target output is not known for a given node in an inner layer, then σ_j for the node is calculated differently as follows.

Let M_k be a node in an inner layer (the layer immediately before the output layer). Let the output of M_k be a_{mk} which is fed as input to all the h nodes of the output layer. Let the weights associated with these h inputs, make the weight vector of $G = [g_1, \ldots, g_h]$. Let the σs calculated for the h nodes of the output layer make the vector $D = [\sigma_1, \ldots, \sigma_h]$. The inner product of G and D is:

$$G \cdot D = \alpha = \sum_{i=1}^{h} g_i \sigma_i, \quad (5)$$

$$\sigma_{mk} = a_{mk}(1 - a_{mk})\alpha. \quad (6)$$

As shown above, the σs for the nodes in inner layer L_j are calculated based on the σs for the nodes in layer L_{j+1}. After a neural network is trained, it is tested against the records of a testing data set which have not been previously encountered by the network.

3. Rough sets

Rough set theory, introduced by Pawlak (1982, 1984) and Pawlak et al. (1995), is a new mathematical tool to deal with vagueness and uncertainty. Two major advantages of rough set theory are: (1) it does not need any preliminary or additional information about data, such as probability distributions; and, (2) unlike statistical approaches, it is able to handle conflicting records.

The major difference between the rough set and the traditional set is that the latter has a clear cut boundary. But the boundary of the rough set is located somewhere within an interval. The lower and upper range of this interval is determined by lower and upper approximate spaces. Therefore, the set of objects belonging to each predictive category, d_i, are divided into lower and upper approximate spaces of d_i.

Objects in lower approximate space of d_i, *certainly* contribute to the prediction process of the category and the objects of its upper approximate space *roughly* contribute to the prediction process of the category. Hashemi et al. (1993b, 1997) provides details about this process.

3.1. A brief review of rough sets theory

An *approximation space* P is an ordered pair $P=(U, R)$ where U is a set called a universe and R is a binary equivalence relation on U. The relation R is called an indiscernibility relation. Each member of U is termed an *object* and each equivalence class of R is called an elementary set. If two objects of universe belong to the same elementary set of R, then the two objects are indistinguishable. A finite union of elementary sets is called a *definable set* in P.

Let $A \subseteq U$ and let E^* be a family of equivalence classes of R. The set A is definable in P if for some $G \in E^*$, set A is equal to the union of all the sets in G; otherwise, A is a *non-definable* or a *rough set*. The rough set A is represented by its upper and lower approximations.

The upper approximation of A in P is defined by $\overline{R}A = \{a \in U | [a]_R \cap A \neq \varnothing\}$, where $[a]_R$ is the equivalence class of R containing a. The lower approximation of A in P is defined by $\underline{R}A = \{a \in U | [a]R \subseteq A\}$. The set $M_p(A) = \overline{R}A - \underline{R}A$ is called a *boundary of A in P*.

The representation of data as an information system (i.e., decision table) is a main attraction of the rough sets and it is defined as follows.

An information system is a quadruple (U, Q, V, δ) in which U is a non-empty finite set of Objects, o. Q is a finite set of attributes. $V = \bigcup_{q \epsilon Q} V_q$ and V_q is the domain of attribute q. δ is a mapping function such that $\delta(o, q) \in V_q$ for every $q \in Q$ and $o \in U$.

Let $S=(U, Q, V, \delta)$ be an information system, $F \subseteq Q$, and $o_i, o_j \in U$. The objects o_i, o_j are indiscernible by set of attributes F in S, denoted by $o_i \tilde{F} o_j$, iff for each f in F, $\delta(o_i, f) = \delta(o_j, f)$. $\tilde{F}$ is an equivalence relation on U for every F in Q. The set of all equivalence classes of $\tilde{F}$ is called a *classification generated by F in S*. An ordered pair $(U, \tilde{F})$ is an approximate space P. For any $A \subseteq U$, the lower and upper approximations of A in P will be called F-lower and F-upper approximations of A in S and shown by $\underline{F}A$ and $\bar{F}A$.

The set of Q attributes may be viewed as a set of condition attributes (C) and a set of decision attributes (D). Let $\Psi = (Y_1, \ldots, Y_k)$ and $\Psi' = (Y'_1, \ldots, Y'_n)$ be the classifications generated by D and C in S, respectively. Y_i is an equivalence class of $\tilde{D}$ and Y'_i is an equivalence class of $\tilde{C}$. Let $L(C,D)$ be defined as: $L(C,D) = \bigcup_{i=1}^{k} \underline{C}Y_i$. The degree of dependency between the condition attributes C and the decision attribute D is defined as:

$$\gamma(C,D) = \frac{|L(C,D)|}{|U|}.$$

Let $F \subseteq C$ and $F' \subset F$. The set of attributes F is a reduct of attribute C if: $\gamma(F,D) = \gamma(C,D)$ and $L(F,D) \neq L(F',D)$ for every F' in F.

Table 1 represents an information system. The universe $U = \{o_1, \ldots, o_9\}$ and each object in U is represented by a record. For this universe, Q is composed of two parts which are: a set of condition attributes, $C = \{c_1, c_2, c_3, c_4\}$; and a decision attribute, D. (Since there is only one attribute in the decision set D, that attribute is also termed D.)

Table 1
An information system and its two-dimensional reduct

The information system

U	C_1	C_2	C_3	C_4	D
o_1	1	1	2	2	1
o_2	2	3	2	1	3
o_3	3	1	1	3	2
o_4	1	1	1	3	2
o_5	2	2	3	2	1
o_6	3	3	2	3	2
o_7	1	1	2	2	1
o_8	2	1	1	3	2
o_9	3	3	2	1	3

The two-dimensional reduct

U	C_1	C_4	D
o_1	1	2	1
o_2	2	1	3
o_3	3	3	2
o_4	1	3	2
o_5	2	2	1
o_6	2	3	2

3.2. Reduction of an information system

An information system may be horizontally reduced by removing its redundant condition attributes. In a horizontally reduced information system, the condition attributes are independent and no attribute can be eliminated further without losing some information from the information system.

This paper employs a two-dimensional reduction (2D-Reduction) approach introduced by Hashemi et al. (1993a, 1997). In this approach, the information system is reduced both horizontally and vertically. The outcome of this reduction is called a 2D-Reduct. To define a 2D-reduced information system, vertical and horizontal reductions are defined.

The *Vertical Reduction* of an information system S is $\mathbf{S} = \text{VERT(S)}$ where $\mathbf{S}$ has all the properties of S, but it is free of duplicate records (i.e., objects).

The Horizontal Reduction of an information system S is $\mathscr{S} = \text{HORIZON(S)}$ where: (1) $\mathscr{S}$ has all the objects of S; and, (2) the set of condition attributes in $\mathscr{S}$ is the reduct of attributes in S (Slowinski et al., 1997).

The 2D-Reduction of information system S is $\mathbb{S} = \text{2D-RED(S)}$ where $\mathbb{S} = \text{HORIZON(VERT(S))}$.

A 2D-Reduct is generated using the 2D-Reduction algorithm. An information system may have several 2D-Reducts.

3.3. Algorithm 2D-reduction

Given: An information system, S. The set of conditions in S is C and the decision attribute is D. The conditions' values for object o_i in S is the set k_i and its decision is d_i.

Objective: A minimal 2D-Reduct of S.

Step 1. Vertical reduction (i.e., removal of the redundant objects from S). For two objects, o_i and o_j in S: $(k_i = k_j)\ (d_i = d_j) \Rightarrow (o_i = o_j)$. Thus, o_j is a redundant object and it will be removed from S.

Step 2. Horizontal reduction (i.e., removal of redundant condition attributes). Let $\mathbf{S}$ be the vertically reduced S, $\mathbf{S} = \text{VERT(S)}$. Furthermore, let the number of objects in $\mathbf{S}$ be $n = |\mathbf{S}|$. S and $\mathbf{S}$ have the same set of condition attributes. Let $P(C) = \{C^{(1)}, \ldots, C^{(m)}\}$ be the power set of C. Let S_i be an information system in which the conditions' set is $C^{(i)}$, $(C^{(i)} \neq C$ and $C^{(i)} \neq \varnothing)$ and the objects are the same as objects in $\mathbf{S}$. Let $\mathbf{S}_i = \text{VERT}(S_i)$. If $|\mathbf{S}_i| = |\mathbf{S}|$ then $C^{(i)}$ is a reduct of C. C may have several reducts ($C^{(a)}, \ldots, C^{(n)}$.) If $|C^{(i)}| = \text{Min}(|C^{(a)}|, \ldots, |C^{(n)}|)$ then, $C^{(i)}$ is the minimal reduct of C and $\mathbf{S}_i$ is called a horizontal reduction of S. (i.e., $= \text{HORIZON}\ (\mathbf{S})$). Also, $\mathbf{S}_i$ is called a minimal 2D reduct of S or simply a *2D-Reduct* of S. Since C may have several minimal reducts, S may have several 2D-reducts.

In our hybrid classification system, the 2D-Reduct increased the predictive power of the neural network model by removing redundant conditions that do not contribute to the correct classification performance of the network. The results revealed that the decision table generated by the application of the 2D-Reduct approach provided more effective input to the neural network, resulting in a more effective classification of the testing set than the initial decision table.

4. Banking data set

The banking data set contained 28 ratios for 256 Arkansas banks for 1992. These ratios were constructed from 70 variables published by the Federal Reserve System, the US central bank. In Arkansas, there were 63 banks independently owned, 114 individual banks owned by another firm, 34 pairs of banks owned by a third party, and 45 sets of three or more banks owned by another enterprise.

The holding structure of this sample reflects the general economic conditions in the mid-south (US). It is interesting to develop a prediction model specifically tailored to the economic environment of this region. Involving banks from the same region, instead of those from other states, increases the sample's homogeneity. Each of the 256 Arkansas banks are described by 28 financial ratios, similar to those used by Tam and Kiang (1992). The list of ratios is shown in Table 2. The selection of variables followed closely the CAMEL criteria used by the Federal Deposit Insurance Corporation (FDIC). CAMEL is an acronym for Capital, Asset, Management, Equity, and Liquidity which is generally adopted by all US bank regulatory agencies.

The 28 ratios can be grouped into categories representing the aforementioned criteria. From Table 2, ratios 6–8 represent the capital adequacy of the bank; ratios 20–25 measure asset quality; the bank's earnings are captured by ratios 14–19; liquidity is represented by ratios 1 and 2; and, the quality of management is quantified by ratios 9–13. In addition, these 28 ratios on Arkansas banks contain measures of credit risk (ratios 3–5) and measures of profitability (ratios 14–19).

4.1. Testing and training sets

Ten percent of each category of bank holding structure was selected for the testing set. Since there were unequal numbers of banks in each category, there were different numbers of banks in each category in the testing set. In order of catego-

Table 2
List of financial variables for arkansas banks

Liquidity
1. Net loans/deposits
2. Cash/demand deposits

Credit risk
3. Loan loss provisions/net loans [a]
4. Allowances/net loans [a]
5. Charge-offs/net loans [a]

Leverage (Capital)
6. Equity/capital
7. Net worth/net loans [a]
8. Net worth/deposits [a]

Efficiency (Quality of management)
9. Non-interest expense/total operating expense
10. Non-interest expense/total operating income
11. Non-interest income/total operating income
12. Non-interest income/total assets [a]

Productivity (Quality of management)
13. Total operating income/total assets [a]

Profitability (Earnings)
14. Net income/total operating income
15. Net income/total assets
16. Net interest income/total assets [a]
17. Net income/net interest income
18. Total assets/equity
19. Return on equity [a]

Loan portfolio (Asset quality)
20. Real estate loans/total loans
21. Agricultural loans/total loans
22. Commercial loans/total loans
23. Loans to individuals/total loans
24. Acceptances of other banks/total loans
25. Reserves for loans and leases/net loans

Others
26. Total deposits/total liabilities
27. Total non-accruals/total liabilities
28. Total demand deposits/total liabilities [a]

[a] Indicates which ratios were dropped by the rough sets.

Table 3
Construction of initial testing and training sets

	Independent	Single holding	Pair holding	Three holding	Total
Original sample	63	114	34	45	256
Testing set	6	11	3	4	24
Remainder	57	103	31	41	
Training set	31	31	31	31	124

ry, the numbers in the testing set were six, eleven, three and four, respectively.

For the training set, an equal number from the remaining records in each category was selected. The objective was not to bias the training toward one category of bank holding structure. Since the smallest category, among the remaining records, belongs to the third classification (i.e., a pair of banks owned by a third party) with 31 records; Thus, 31 bank records were selected from each of the four categories of bank holding structures to train the neural network (see Table 3). For details of this process, consult Hashemi and Stafford (1993).

4.2. Data reduction

The algorithm 2D-Reduct was applied to the original data set of 28 variables (columns) and 256 banks (rows). The algorithm identified 13 duplicate records (i.e., banks with identical values for conditions and decision). Twelve of these records were discarded and one retained. Of the original 28 attributes (variables), the algorithm identified ten variables to be discarded. The outcome of the reduction process was a minimal 2D-Reduct with 244 objects (records) and 18 attributes.

The 28 original financial ratios (i.e., variables) in eight categories are given in Table 2. All three credit risk ratios were eliminated as well as the one productivity measure. Two of three leverage ratios and two of six profitability ratios were also eliminated. There was at least one variable or ratio remaining in six of the eight classifications. Overall, 36% of the attributes were eliminated by the 2D-Reduct procedure.

The minimal 2D-Reduct was also divided into training and testing sets with the same procedure that was used for the original data. Ten percent of the reduced data set was selected for the second testing set. Therefore, 29 bank records were randomly selected from each of the four categories of bank holding structures to constitute the training set (see Table 4).

As a result, there are two pairs of training and testing sets, namely initial training and testing sets and reduced training and testing sets. Two different neural networks were developed to model the prediction of bank holding structures. The two pairs of training and testing sets were employed for training and testing of these networks.

Even though the reduced training set is smaller than the initial training set, it is of higher quality. Reducing the number of columns removes "noisy" attributes. Eighteen of 28 variables have a relationship with the outcomes in the decision table. In other words, the decision attribute of the objects in the original dataset is a function of only 18 attributes, not 28.

Table 4
Construction of reduced testing and training sets

	Independent	Single holding	Pair holding	Three holding	Total
Original sample	61	111	32	40	244
Testing set	6	11	3	4	24
Remainder	55	100	29	36	
Training set	29	29	29	29	116

Table 5
Analysis of NET-1: Training set

Hidden nodes	Output nodes	Epochs	Records in training set	Records trained
3	4	300	124	64 (51%)
4	4	300	124	74 (89%)
5	4	300	124	95 (92%)
6	4	300	124	119 (96%)
7	4	300	124	124 (100%)
8	4	300	124	124 (100%)
9	4	300	124	120 (97%)
10	4	300	124	120

5. Classification of bank holding structures

5.1. Initial neural network (NET-1)

In the development of NET-1, determining the *best* configuration for the neural net is the main concern. That is, the decision on the number of layers, number of nodes for each layer, and the activation function for the network must be made. One layer was deemed sufficient, as a network with one hidden layer is powerful enough to "represent" such a problem. Representation means that the network with one hidden layer is able to simulate the prediction. As a result, the network has one input layer, one hidden layer, and one output layer. Also, in the training of NET-1, the process of "averaging" takes place and employs a sigmoid function.

To determine the number of nodes for each layer, the process begins with the output layer. The output of NET-1 is a prediction value. The possible values for a prediction are 0, 1, 2, and 3 which represent the four types of bank holding structures defined earlier. From past experience, the best number of nodes in the output layer for this type of problem is four. The prediction value (i.e., output) of 0, 1, 2, and 3 are represented by four-bit patterns of (1,0,0,0), (0,1,0,0), (0,0,1,0), and (0,0,0,1), respectively. This means that for the prediction value 0, 1, 2, and 3, the nodes 1–4 of the output layer will be fired (having value 1), respectively.

To determine the number of nodes for input and hidden layers, there are 28 conditions in each record of the initial training set which logically demand 28 input nodes for the input layer. But the number of nodes for the hidden layer is determined by trial and error process. This approach determines what is the best number of nodes for the hidden layer such that the system will be trained for the most number of records in the initial training set and be able to predict correctly more records of the initial testing set. NET-1 was trained using different numbers of nodes in the hidden layer (3–10 nodes); refer to Table 5. The results in Table 5 reveal that having seven or eight nodes for the hidden layer is a proper choice. Seven nodes were selected because it reduced the training time for the network.

To conclude, 28 nodes for the input layer, seven nodes for the hidden layer and four nodes for the output layer constitute the best configuration for NET-1. The detailed analysis of the predictive performance by NET-1 on the initial testing set using two different tolerance levels (i.e., 0.15 and 0.20) are illustrated in Tables 6 and 7. The tolerance level refers to the difference between the actual predicted value provided by the neural network and the prediction values of 0, 1, 2 and 3, representing the four output categories. For example, if the tolerance level is 0.20, a predicted value from the network anywhere between 0.80 and 1.20 would we consider as correctly classified in the first category

Table 6
Classification performance of NET-1 at 0.15 tolerance level

Category	Correct	Incorrect	Total	Percent
Independent	5	1	6	83.3
Single holding	10	1	11	90.9
Pair holding	3	0	3	100.0
Three holding	0	4	4	0.0
Total	18	6	24	75.0

Table 7
Classification performance of NET-1 at 0.20 tolerance level

Category	Correct	Incorrect	Total	Percent
Independent	6	0	6	100.0
Single holding	11	0	11	100.0
Pair holding	3	0	3	100.0
Three holding	0	4	4	0.0
Total	20	4	24	83.3

(a prediction value of 1). But at the same tolerance level 0.20, a predicted value of 0.75 or 1.25 would not be a correct classification in the first category.

Table 6 provides the classification results from the NET-1 model of using the complete set of 28 attributes (i.e., financial ratios) as input. The overall correct classification rate was 75% at a 0.15 tolerance level. Individual category or group classification rates (i.e., percent correctly classified) by the neural network were 83.3% for independently owned bank, 90.9% for single banks owned by another firm, 100% for pairs of banks owned by another enterprise, and zero percent for three or more banks held by another business organization. (This information can be read down the last column in Table 6.)

Table 7 provides the classification results from the NET-1 model but at a 0.20 tolerance level. Membership of Arkansas bank records for three of the four categories of holding structure was correctly classified at 100% accuracy. Even at the relaxed tolerance level, the network was not able to correctly classify any of the bank cases in the fourth category – three or more banks owned by a separate entity. The overall correct classification rate increased to over 83% at this tolerance level. At a tolerance level of 0.30, the initial neural network correctly classified 100% of the 24 cases in the testing set.

5.2. Second neural network (NET-2)

The configuration of NET-2 was determined by the same procedure used for NET-1. The result was a network structure with 18 nodes for the input layer, seven nodes for the hidden layer, and four nodes for the input layer. The detailed performance analysis for the reduced testing set of NET-2, using two different tolerance levels of 0.15 and 0.2 are illustrated in Tables 8 and 9, respectively.

Table 8 provides the classification results for NET-2 which employed the reduced set of 18 attributes (i.e., financial ratios) as input. The overall correct classification rate was 83.3% at a 0.15 tolerance level. Individual category or group classification rates (i.e., percent correctly classified) by the neural network were 83.3% for independently owned bank (no change), 100% for both single banks owned by another firm (9% increase) and pairs of banks owned by another enterprise (no change), and zero percent for three or more banks held by another business organization.

Table 9 provides the classification results from the NET-2 model at a 0.20 tolerance level. The number of nodes for input, hidden and output layers was 18, five, and four, respectively. Membership for the Arkansas bank records for three of the four categories of holding structure was correctly classified at 100% accuracy. But at the relaxed tolerance level, the network was now able to correctly classify 75% of the bank cases in the fourth category. The overall correct classification rate increased to 96% at this tolerance level.

At a tolerance level of 0.30, the second neural network correctly classified 100% of the 24 cases in the testing set.

Table 8
Classification performance of Net-2 at 0.15 tolerance level

Category	Correct	Incorrect	Total	Percent
Independent	5	1	6	83.3
Single holding	11	0	11	100.0
Pair holding	3	0	3	100.0
Three holding	0	4	4	0.0
Total	20	4	24	83.3

Table 9
Classification performance of NET-2 at 0.20 tolerance level

Category	Correct	Incorrect	Total	Percent
Independent	6	0	6	100.0
Single holding	11	0	11	100.0
Pair holding	3	0	3	100.0
Three holding	3	1	4	75.0
Total	23	1	24	95.8

Table 10
Classification performance of discriminant analysis on initial testing set

Category	Cases	Group 1	Group 2	Group 3	Group 4
Independent	6	2	2	2	0
		33.3%	33.3%	33.3%	0.0%
Single holding	11	4	1	4	2
		36.4%	9.1%	36.4%	18.2%
Pair holding	3	0	2	1	0
		0.0%	66.7%	33.3%	0.0%
Three holding	4	0	0	2	2
		0.0%	0.0%	50.0%	50.0%

Percent of "grouped" cases correctly classified: 25%.

5.3. Multiple discriminant analysis

To provide a comparative basis for judging the predictive capability of NET-1 and NET-2, a multiple discriminant analysis (Hair et al., 1992) was performed. The comparison applies the discriminant analysis on the same two pairs of initial and reduced training and testing datasets. For implementation of the discriminant analysis, SPSS/PC+ was employed. Tables 10 and 11 show that this statistical procedure could only correctly predict bank holding structures in 25% of the initial testing set and 45.8% of the reduced testing set. (The "Group" heading for the columns in Tables 10 and 11 also refer to the "Category" row titles.)

To read these particular tables, determine how many cases were in each category of bank holding structure in the testing set. For example, in the above table there are six cases (i.e., records) representing independent banks. The first multiple discriminant model predicted that there were but two banks in the first ("independent" group), two in the second group, two in the third and none in the fourth group. This model correctly classified one-third or 33.3% of the first category of independently owned banks. Overall, the model only classified one-quarter or 25% of the 24 banks in the testing set.

The second multiple discriminant model which employed the reduced set of explanatory variables showed an improved classification performance. The rough sets determined group of input variables helped construct a better model, at least in a relative sense. Predictive performance of the discriminant model increased from 25 to 45.83%, an substantial improvement over the initial statistical model but still more than 25 percentage points below the least accurate neural network (i.e., NET-1 at 0.15 tolerance level).

6. Summary and conclusions

A combination of AI techniques, namely 2D-Reduct and neural networks, have demonstrated

Table 11
Classification performance of discriminant analysis on reduced testing set

Category	Cases	Group 1	Group 2	Group 3	Group 4
Independent	6	2	3	1	0
		33.3%	50.0%	16.7%	0.0%
Single holding	11	4	5	1	1
		36.4%	45.5%	9.1%	0.1%
Pair holding	3	0	0	2	1
		0.0%	0.0%	66.7%	33.3%
Three holding	4	1	0	1	2
		25.0%	0.0%	25.0%	50.0%

Percent of "grouped" cases correctly classified: 45.83%.

the effectiveness of this association. By applying rough sets to the decision table of Arkansas bank attributes, this research has demonstrated its usefulness in reducing the noise inherent in a rather large matrix of ratio data. With statistical models, reducing the number of explanatory variables, at best, will not change the predictive accuracy of the model certainly for the training set and perhaps on the test set. Usually, their predictive accuracy and explanatory power decreases, such as a drop in the explained variation in a regression or other linear model. By reducing the decision table, the 2D-Reduct reduces noise (unexplained variation) by potentially removing both attributes as well as redundant records.

By using the revised decision table, the second neural network (i.e., NET-2) provided a substantial increase in overall classification accuracy. This network exhibited an 8% increase (from 75 to 83) and 13% increase (from 83 to 96) at 0.15 and 0.20 tolerance levels, respectively. If the fourth category of holding structure (three or more banks owned by a single entity) was excluded, the predictive accuracy of the neural networks would be 90% and 95%, respectively at 0.15 and 0.20 tolerance levels. Of the 256 Arkansas banks in the dataset, only 45 of them (or 17.3%) were of this last type. If there had been more banks of the latter two kinds in the dataset, the training set for all types could have been larger, and thereby possibly providing an opportunity for the neural net to train to classify this type of bank holding structure with more accuracy.

As suggested in the paper, the practical application of this hybrid classification tool is to identify potential target banks for merger or acquisition. Larger banks are the primary beneficiary, but not in every instance. There exists a "food chain" style of bank growth. Larger banks purchase or merge with smaller ones. A large group of banks merge or acquire another but smaller network of banks. The most likely targets for merger and acquisition are the independent banks, as well as the small bank (i.e., few controlled banks) holding companies. Therefore, predictive accuracy for the smaller networks, i.e., the first three categories of bank ownership, is more critical. The second neural network at 0.15 tolerance level correctly predicted 95% of the testing set of banks. The smaller tolerance level is much preferred, not only because of its higher accuracy, but also due to its increased reliability.

Another approach for this hybrid tool is to use desired values for the 18 input attributes of the neural network. These variables represent the CAMEL criteria of the FDIC. The network would identify which category that a desirable bank would be a member. Depending on the actual values of the input ratios employed, a desirable bank merger or acquisition target may be a member of any of the four categories. Since the second category is the largest, containing 114 banks representing over 44.4% of Arkansas' banks, there is a greater likelihood for potential targets among that group.

Neural networks combined with rough sets make a potent hybrid tool for a number of applications in banking and finance. The identification of merger and/or acquisition targets represents an application for such a combination of applied AI approaches.

In contrast to the traditional reduct generated by the rough sets approach, a 2D-Reduct as employed in this application reduces redundant attributes as well as redundant objects from the decision table. Therefore, the 2D-Reduct contains less "noisy" data and provides a decision table that can yield a substantially lower misclassification rate when the decision table is subsequently processed by other AI techniques such as a neural network. This two-step process can take place in a hybrid or integrated decision support system.

References

Collins, E., Ghosh, S., Scofield, C., 1988. An application of a multiple neural network learning system to emulation of mortgage underwriting judgments. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 459–466.

Dutta, S., Shekhar, S., 1988. Bond rating: A non-conservative application of neural networks. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 443–450.

Fletcher, D., Goss, E., 1993. Forecasting with neural networks: An application using bankruptcy data. Information and Management 24, 159–167.

Groosberg, S., 1974. Classical and instrumental learning by neural networks. In: Progress theoretical biology, Vol. 3, pp. 51-141 Academic Press, New York, NY.

Hair, J.F., Jr., Anderson, R.E., Tatham, R.L., Black, W.C. 1992. Multivariate data analysis with readings, Macmillan, Newyork, NY.

Hashemi, R., Stafford, N., 1993b. A backpropagation neural network for risk assessment. In: Proceedings of the 1993 IEEE International Phoenix Conference on Computers and Communications, Scottsdale, AZ, pp. 565–570.

Hashemi, R.R., Jelovsek, F.R., Razzaghi, M., 1993a. Developmental toxicity risk assessment: A rough sets approach. International Journal of Methods of Information in Medicine 32, 47–54.

Hashemi, R., Choudhury, A., Stafford, N., Talburt, J., 1993b. Prediction capability of neural networks trained in Monte-Carlo paradigm. In: Proceedings of the 1993 ACM/IEEE International Symposium on Applied Computing, Indianapolis, pp. 9–13.

Hashemi, R.R., Pearce, B.A., Arani, R.B., Hinson, W.G., Paule, M.G., 1997. A fusion of rough sets, modified rough sets and genetic algorithms for hybrid diagnostic systems. In: Lin, Ty., Cercone, N. (Eds.), Rough Sets and Data Mining: Analysis of Imprecise Data, Kluwer Academic Publishers, Dordrecht, 1997, pp. 149–176.

Hertz, J., Krogh, A., Palmer, R.G. 1991. Introduction to the theory of neural computation. Addison-Wesley, Redwood City, CA.

Jarke, M., Vassiliou, Y., 1984. Coupling expert systems with database management systems. In: Reitman, W. (Ed.), Artificial Intelligence Applications for Business, Aplex, Norwood, NJ.

Jelonek, J., Krawiec, K., Slowinski, R., 1995. Rough set reduction of attributes and their domains for neural networks. Computational Intelligence 11 (2), 339–347.

Messier, W.F., Jr., Hansen, J.V., 1988. Inducing rules for expert system development: An example using default and bankruptcy data. Management Science 34 (12), 1403–1415.

Pawlak, Z., 1982. Rough-sets. International Journal of Information and Computer Science 11, 341–356.

Pawlak, Z., 1984. Rough classification. International Journal of Man-Machine Studies 20, 469–483.

Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W., 1995. Rough sets. Communications of the ACM 38 (11), 89–95.

Rochester, J.D., 1990. New business uses for computing. I/S Analyzer 28, 1–16.

Salchenberger, L.M., Cinar, E.M., Lash, N.A., 1992. Neural networks: A new tool for predicting thrift failures. Decision Sciences 23 (4), 899–916.

Slowinski, R., Zopoundis, C., 1995. Application of the rough set approach to evaluation of bankruptcy risk. International Journal of Intelligent Systems in Accounting, Finance and Management 4 (1), 27–41.

Slowinski, R., Zopoundis, C., Dimitras, A.I., 1997. Prediction of company acquisition in Greece by means of the rough set approach. European Journal of Operational Research 100 (1), 1–15.

Staats, W.F., 1988. Key concepts of financial management for bankers, Graduate School of Banking of the South, Baton Rouge.

Tam, K.Y., Kiang, M.Y., 1992. Managerial applications of neural networks: The case of bank failure predictions. Management Science 38 (7), 926–947.

Trippi, R., Turban, E., 1989. The impact of parallel and neural computing on managerial decision making. Journal of Management Information Systems 6, 85–97.

Turban, E., Watkins, P.R., 1986. Integrating expert systems and decision support systems. MIS Quarterly 10 (2), 121–136.

Wasserman, P., 1989. Neural Computing: Theory and Practice. Van Nostrand Reinhold, New York.