ADAPTIVE INFERENCE-BASED LEARNING AND RULE GENERATION ALGORITHMS IN FUZZY NEURAL NETWORK FOR FAILURE PREDICTION

Vahid Behbood, Jie Lu, Guangquan Zhang
Decision Systems & E-Service Intelligence Research Laboratory
Center for Quantum Computing and Intelligent System, School of Software
Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney,
POBOX123, Broadway, NSW2007, Australia
Emails: vbehbood@it.uts.edu.au, jie.lu@uts.edu.au, guangquan.zhang@uts.edu.au

Abstract-- Creating an applicable and precise failure prediction system is highly desirable for decision makers and regulators in the finance industry. This study develops a new Failure Prediction (FP) approach which effectively integrates a fuzzy logic-based adaptive inference system with the learning ability of a neural network to generate knowledge in the form of a fuzzy rule base. This FP approach uses a preprocessing phase to deal with the imbalanced data-sets problem and develops a new Fuzzy Neural Network (FNN) including an adaptive inference system in the learning algorithm along with its network structure and rule generation algorithm as a means to reduce prediction error in the FP approach.

Keyword--Fuzzy neural network; Failure prediction; Adaptive fuzzy inference systems; Imbalanced data-sets

I. INTRODUCTION
Since the advent of various financial crises in the 1990s and 2000s - particularly the recent recession in mid-2008 - there have been extensive investments in the construction of accurate computational systems to predict the probability of financial crises and bank failures. From 1980 to 1996 three-quarters of the IMF countries experienced bank failures which were not restricted to particular geographic regions, levels of development or banking system structures [1]. These bank failures are given in previous research as clear evidence of serious distress. To tackle these problems, various data analysis models and prediction systems to forecast the bank failure have been developed. Although it has been proved that these models are useful to managers and regulators with authority to prevent the occurrence of failure [2, 3], some drawbacks make them inapplicable as inevitable and vital systems for business. Most of the existing approaches, which use statistical methods, have deficiencies such as: ignoring important sources of uncertainty in classification as an arbitrary definition of failure; data instability and arbitrary choice of the optimization criteria; and neglecting time dimension of failure [3]. In addition, almost all existing statistical financial prediction models have been criticized for their assumption which is more likely to be violated in the fields of finance and economics [4]. On the other hand, the growing development and application of computational intelligence techniques has led researchers to employ new methods in financial warning systems [5] such as Support Vector Machine (SVM), Case Based Reasoning (CBR) and Genetic Algorithm (GA) [3, 6, 7]. Ravi Kumar and Ravi provide a detailed review of these models and methods in the domain of bankruptcy prediction and demonstrate their better performance in differing aspects such as accuracy.

The most popular computational intelligence technique that has been significantly applied to the domain of forecasting is Artificial Neural Network (ANN) [2, 7, 8]. Although ANN is a well-known, efficient tool for prediction, it works as a ‘black box’ due to its computational framework. It learns only the relationship between inputs and outputs, without providing any knowledge about the relationship between inputs themselves and outputs which can be critical for decision making.

Fuzzy systems have been introduced in this area to tackle the imprecise nature of financial forecasting and effectively present expert knowledge about the influence of input variables on financial situation, as output through a fuzzy rule base [9-13]. However it is not as accurate as ANN.

Fuzzy Neural Network (FNN), which is a hybrid model, uses ANN and fuzzy systems to create a robust hybrid classifier and forecaster tool in different fields. In some recent research, different kinds of FNNs are used to classify and predict financial failures [4, 7, 14-17]. The main advantages of these models are their consistent fuzzy rule base gained from fuzzy systems along with their learning ability and accuracy obtained from ANN, to prevent probable future failure. Although their knowledge generation ability makes them suitable for prediction, their prediction accuracy suffers in comparison with ANN, which is the most accurate predicting method with the limitation of functioning as a ‘black box’ [7].
This paper outlines the development of a new approach for FP using FNN. The proposed FP approach contains three main phases: 1) A preprocessing technique called: Synthetic Minority Oversampling Technique (SMOTE), to deal with imbalanced data-sets problem in failure prediction, 2) A clustering technique and specifying the network structure and rule formulation algorithm to dynamically compute the input fuzzy clusters and fuzzy rules from numerical training data, and 3) An adaptive inference system with a parametric t-norm operator in the learning algorithm to reduce prediction error.

This approach can significantly improve the failure prediction accuracy. Along with supplying a valuable and apprehensive financial knowledge base, this approach will have a superior performance. The novelty of the proposed FP approach not only organizes the appropriate phases together as a framework to establish a FP approach, but also presents an efficient neural network structure, rule generation and learning algorithms to gain better results.

This paper is organized as follows: Section 2 introduces preliminaries including the class imbalanced data-sets problems in FP, proper measures to evaluate accuracy, and a fuzzy clustering method. In Section 3 we outline the FP approach and present the structure of the proposed FNN with its rule generation algorithm. Section 4 presents an adaptive inference-based learning. Finally, conclusion and further study are provided in Section 5.

II. PRELIMINARIES

In this section, the problem of imbalanced data-sets in failure prediction is introduced then we outline the preprocessing solution, which is applied to the problem. A measure to evaluate the accuracy is also presented and a method for clustering financial data is addressed.

A. The problem of imbalanced data-sets

This paper focuses on two class imbalanced data-sets problem, where there is only one positive class (failure bank) with a lower number of examples, and one negative class (survived bank) with a higher number of examples. During learning from imbalanced data-sets, the classifier will obtain a high predictive accuracy for the majority class, but will predict poorly for the minority class which is equally necessary in prediction [18]. Likewise, the classifier may consider the minority class as noise, which is then ignored.

B. The synthetic minority oversampling technique

There are a large number of techniques which have been proposed to deal with imbalance data-sets problem. There is some research demonstrating that oversampling methods, especially SMOTE, significantly improves the prediction accuracy of Fuzzy Rule Base Classifiers [19-21]. This approach effectively makes the decision region of minority class more general, and therefore the over-fitting problem is avoided and minority examples spread further into the majority samples [21].

C. Accuracy evaluation

Prediction accuracy is evaluated based on a confusion matrix shown in Table 1. In the table TP (TN) is the number of instances that is predicted correctly positive (negative). FP (FN) is the number of examples that is predicted wrongly positive (negative) and actually they belong to negative (positive) class.

<table>
<thead>
<tr>
<th>Confusion Matrix for two-class problem</th>
<th>Positive Prediction</th>
<th>Negative Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Class</td>
<td>True Positive (TP)</td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td>Negative Class</td>
<td>False Positive (FP)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

The important point in dealing with the imbalanced data-sets problem is that, in comparison with the majority class, the minority class has very little impact on accuracy (1), which is used mostly in experimental measures.

\[
\text{Acc} = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}
\]

More correct metric (2) is presented to evaluate accuracy:

\[
GM = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{FP + TN}} \tag{2}
\]

where \(\frac{TP}{TP + FN}\) is called sensitivity and \(\frac{TN}{FP + TN}\) is called specificity. The proposed metric (2) is a geometric mean of sensitivity and specificity, because both of them are expected to be high simultaneously [22, 23].

D. Discrete Incremental Clustering

Performing a cluster analysis is the first step towards modeling the problem. There is a novel self organizing clustering technique which outperforms other techniques; DIC method is a dynamic clustering technique avoiding drawbacks such as stability-plasticity and inflexibility found in other methods and computing trapezoidal-shaped fuzzy sets shown in Figure 1 [14]. These fuzzy sets are applied as \(\text{Term}_{ij}, i \in \{1,2,...,n\} \& j \in \{1,...,k_i\}\) and \(\text{OTerm}_{ij}, i \in \{1,2,...,m\} \& j \in \{1,...,k_j\}\) for defining input and output linguistic terms respectively in FNN structure.

![Fig1. Trapezoidal fuzzy set](image-url)
III. FUZZY NEURAL NETWORK STRUCTURE AND RULE GENERATION ALGORITHM

In this section, we first outline our FP approach. The proposed structure of FNN and its rule generation algorithm are described. According the rule weights, weak rules are pruned and only strong rules are used in inference and learning algorithm.

A. Bank failure prediction approach outline

The proposed FP approach contains three main phases: Phase 1: Applying SMOTE to deal with imbalance data-sets problem in failure prediction. Phase 2: Using DIC method and proposing a structure of FNN and developing a rule generation algorithm to dynamically compute the input fuzzy clusters and fuzzy rules from numerical training data. Phase 3: Developing and adaptive inference-based learning algorithm to reduce prediction error. This approach is shown in Figure 2.

A. Structure of Fuzzy Neural Network

Assume that \( X = [x_1, x_2, ..., x_n] \) and \( Y = [y_1, y_2, ..., y_m] \) represents the vector of inputs and outputs respectively to and from the network. In addition, assume the vector \( D = [d_1, d_2, ..., d_m] \) represents the desired (actual) outputs required during the learning phase.

![Fig2. FP approach](image)

The proposed network has five layers of nodes: (1) Input layer consisting of input nodes \( I_{ij}, i \in \{1,2, ..., n\} \), which have a single input \( x_i \). (2) Input cluster layer including cluster nodes \( ITerm_{ij}, i \in \{1,2, ..., n\} \& j \in \{1, ..., k_i\} \) which represent linguistic terms for each input. For instance, \( ITerm_{23} \) represents the third linguistic term of input \( x_2 \). (3) Rule layer consisting of rule nodes \( R_{hi}, h \in \{1,2, ..., l\}; l = \sum_{i=1}^{m} k_i \) which are representative of each rule and connects only one labels of each input \( (x_i; i \in \{1,2, ..., n\}) \) to one labels of each output \( (y_j; i \in \{1,2, ..., m\}) \). (4) Output cluster layer including clusters nodes \( OTerm_{ij}, i \in \{1,2, ..., m\} \& j \in \{1, ..., k_j\} \) which represent linguistic terms for each output. For instance, \( OTerm_{55} \) represents the fifth linguistic term of output \( y_5 \). (5) Output layer consisting of output nodes \( O_{ij}, i \in \{1,2, ..., m\}, \) which have single output \( y_i \). Figure 3 shows the structure of the proposed FNN.

![Fig3. FNN structure](image)

B. A rule generation algorithm

The rule generation algorithm is developing by formulating an algorithm which is based on the structure of our proposed FNN. In this algorithm, IS\( P_h \) is the terms for all input linguistic terms (layer 2 nodes) that contribute to antecedent of rule node \( R_h \) and OS\( P_h \) refers to all output linguistic terms (layer 4 nodes) that form the consequent of rule node \( R_h \):

\[
IS\!P_h = \{ITerm_{1P1}, ITerm_{2P2}, ..., ITerm_{nPn}\} \quad \text{and} \quad OS\!P_h = \{OTerm_{1Q1}, OTerm_{2Q2}, ..., OTerm_{mQm}\}
\]

\[ \text{if and only if} \]

\[
R_h : \text{If } x_1 \text{ is } ITerm_{1P1} \text{ and } x_2 \text{ is } ITerm_{2P2} \text{ and} \ldots \text{ and } x_n \text{ is } ITerm_{nPn}, \text{then } y_1 \text{ is } OTerm_{1Q1} \text{ and } y_2 \text{ is } OTerm_{2Q2} \text{ and } \ldots \text{ and } y_m \text{ is } OTerm_{mQm} \]

The rule generation algorithm has three main steps as follows: Algorithm 1:

**Step 1: Firing input and desired vectors**

The input vector \( X(T) = [x_1(T), ..., x_n(T)] \) and the desired vector \( D(T) = [d_1(T), ..., d_m(T)] \) are feed to the network through layer 1 and layer 5 respectively at T-th epoch. According input and output linguistic terms (fuzzy sets) membership function, the membership value of inputs and outputs in each term is evaluated and then the linguistic terms of each input and output with maximum membership value is selected:

\[
MI\!Term(T) = \{ITerm_{1b1}, ITerm_{2b2}, ..., ITerm_{nbn}\}
\]

\[
\mu_{ITerm_{ij}}(x_i(T)) = \max \left\{ \mu_{ITerm_{ij}}(x_i(T)), j \in \{1, ..., k_j\} \right\}
\]

Fig3. FNN structure
\[ \text{MOTerm}(T) = \{ \text{OTerm}_{1h}, \text{OTerm}_{2h}, \ldots, \text{OTerm}_{mh} \} \mu_{\text{OTerm}_{ih}}(d_i(T)) = \]
\[ \max \left\{ \mu_{\text{OTerm}_{ij}}(d_j(T)), j \in \{1, \ldots, k_i\} \right\} \tag{5} \]

**Step 2: Indicating nominated rule**

In this step, the rule which has to be nominated for updating in each epoch is selected. The rule which satisfies (6) in epoch T-th is qualified for an update:
\[ IS_{ph} = \text{MOTerm}(T) \text{ and } OS_{ph} = \text{MOTerm}(T) \]
\[ h \in \{1, 2, \ldots, l\}, \quad l = \sum_{i=1}^{n} k_i \times \sum_{i=1}^{m} k_i \]  
\[ \mu_{\text{OTerm}_{ih}}(d_i(T)) = \mu_{\text{OTerm}_{ij}}(d_j(T)) \tag{6} \]

**Step 3: Rule updating**

To update the rule weight in epoch T-th, equation (7) is applied:
\[ W_h(T) = W_h(T-1) + \left( M_{IS_{ph}}(T) \times M_{OS_{ph}}(T) \right) \]
\[ W_h(0) = 0, h \in \{1, 2, \ldots, l\} \tag{7} \]

where \( M_{IS_{ph}}(T) = \mu_{\text{ITerm}_{1h1}} \times \mu_{\text{OTerm}_{2h2}} \times \ldots \times \mu_{\text{OTerm}_{mhn}} \)
and \( M_{OS_{ph}}(T) = \mu_{\text{OTerm}_{1h1}} \times \mu_{\text{OTerm}_{2h2}} \times \ldots \times \mu_{\text{OTerm}_{mhn}} \).

To prune the weak rules at the end of the rule generation phase, a predefined threshold parameter \( \text{Thresh} \) is considered and then each rule satisfying (8) will be pruned at the end of the training step.
\[ l \times \left( \frac{w_h}{\sum_{k=1}^{l} w_k} \right) < \text{Thresh} \tag{8} \]

Through using this algorithm we can obtain an initial consistent and compact fuzzy rule base which needs to be modified and updated to gain more accurate results. In the next section, a learning algorithm is proposed to optimize the rule base.

**IV. ADAPTIVE INFERENCE-BASED LEARNING ALGORITHM**

The adaptive inference-based learning algorithm aims to update the fuzzy rule base, which is obtained by rule generation algorithm, to improve the prediction accuracy. This algorithm tries to adapt the parametric t-norm of inference system to reduce the significance of rules causing errors and consequently augment accuracy.

Assume that \( X(T) = [x_1(T), \ldots, x_2(T)] \) and \( D(T) = [d_1(T), \ldots, d_2(T)] \) are input and desired output vectors in T-th epoch of training respectively and there is a fuzzy rule base that includes \( l \) rules and each rule, \( R_i \) (\( i^{th} \) rule) has the form as shown in (9).
\[ R_i: \text{If } x_1 \text{ is } \text{ITerm}_{1i}, \ldots, x_n \text{ is } \text{ITerm}_{ni}, \text{Then } y_1 \text{ is } \text{OTerm}_{1h1}, \ldots, y_m \text{ is } \text{OTerm}_{mh} \tag{9} \]

The proposed **adaptive inference-based learning algorithm** has seven steps as follows:

**Algorithm 2:**

**Step 1: Fuzzyfying**

Input nodes act as singleton fuzzifier that fuzzyfy the crisp-valued inputs. So, the activation function of each node in layer 1 is defined as:
\[ Z_i^{(1)}(T) = f^{(1)}(x_i(T)) = \mu_{\text{OTerm}_{ij}}(\bar{x}_i(T)) = \]
\[ \begin{cases} 1, & \text{if } \bar{x}_i(T) = x_i(T) \\ 0, & \text{Otherwise} \end{cases} \quad i \in \{1, 2, \ldots, n\} \tag{10} \]

where \( \bar{x}_i(T) \) is the fuzzyfied equivalent of crisp input \( x_i(T) \) and \( Z_i^{(1)}(T) \) is the output of node \( It_i, i \in \{1, 2, \ldots, n\} \) at T-th epoch.

**Step 2: Antecedent matching**

To perform antecedent matching of fuzzyfied inputs against linguistic terms, input cluster nodes compute the similarity measure which is the input membership value in each term. The activation function of each node in layer 2 is defined as:
\[ Z_{ij}^{(2)}(T) = f^{(2)} \left( Z_i^{(1)}(T) \right) = \mu_{\text{ITerm}_{ij}}(x_i(T)) = \]
\[ \begin{cases} 0, & \text{if } x_i(T) \leq l_{ij} \\ \frac{x_i - l_{ij}}{u_{ij} - l_{ij}}, & \text{if } l_{ij} \leq x_i \leq u_{ij} \\ 1, & \text{if } u_{ij} \leq x_i \leq v_{ij} \\ \frac{v_{ij} - x_i}{v_{ij} - u_{ij}}, & \text{if } v_{ij} \leq x_i \leq r_{ij} \end{cases} \quad i \in \{1, 2, \ldots, n\}, \quad j \in \{1, 2, \ldots, k_i\} \tag{11} \]

where \( Z_{ij}^{(2)}(T) \) is the output of node \( It_{ij} \) at T-th epoch.

**Step 3: Rule fulfillment**

To calculate the strength of activation of an antecedent for all rules with input vector \( X(T) \), rule nodes compute the degree of rule fulfillment as their outputs. The higher the degree of fulfillment, the greater is the compatibility of the input to the antecedent of the rule. Hence, the activation function of each node, \( R_{ih} ; h \in \{1, 2, \ldots, l\} \), in layer 3 is defined as:
\[ Z_{R_{ih}}^{(3)}(T) = f^{(3)} \left( Z_{ij}^{(2)}(T), i \in \{1, 2, \ldots, n\} \right) = \]
\[ T \left( Z_{ij}^{(2)}(T), i \in \{1, 2, \ldots, n\} \right) \tag{12} \]

where \( Z_{ij}^{(2)}(T) \) is the output of the j-th term of the i-th input that is connected to the rule \( R_{ih} \) and \( T \) is the conjunction operator. In order to adapt the inference system, this operator...
is considered as parameterized t-norm. Some researchers [24, 25] have shown that a tuning parameter can significantly improve the accuracy of linguistic fuzzy systems. Therefore, Dubios t-norm shown in (13) is used as conjunction operator in this step because it provides better accuracy than other parametric t-norms [24].

\[ T_{\text{Dubios}}(x, y, \alpha) = \frac{xy}{\max(x, y, \alpha)} \quad (13) \]

Dubios t-norm operates as the minimum and an algebraic product with \( \alpha = 0 \) and \( \alpha = 1 \) respectively. The activation function is represented as:

\[ Z_{R_h}^{(3)}(T) = f^{(3)}(Z_{R_h}^{(2)}(T), i \in \{1, 2, ..., n\}) = \]

\[ T_{\text{Dubios}}(Z_{R_h}^{(2)}(T), i \in \{1, 2, ..., n\}, \alpha_h(T)) = \]

\[ (\prod_{i=1}^{n} Z_{R_h}^{(2)}(T))/\text{Max}\{Z_{R_h}^{(2)}(T), i \in \{1, 2, ..., n\}, \alpha_h(T)\} \quad (14) \]

where \( Z_{R_h}^{(3)}(T) \) is the output of node \( R_h \) at T-th epoch. In addition, \( \alpha(0) = (\alpha_1(0), \alpha_2(0), ..., \alpha_i(0)) \) is initialized as 0 at the beginning of the learning algorithm, and then, during the feedback learning phase, each \( \alpha_h \) may be increased separately.

**Step 4: Consequent derivation**

In this step, an aggregator operator is used to derive the consequent of fuzzy rules. Max operator which is considered as s-norm uses rule weights and rule outputs to compute the inferred output according to each rule in the rule base. Therefore, the activation function for each node in layer 4 becomes:

\[ Z_{i|h}^{(3)}(T) = f_{i|h}^{(3)}(Z_{R_h}^{(3)}(T), h \in \{1, ..., l\}, W_h) = \]

\[ \text{Max}(\prod_{h=1}^{l} W_h \times Z_{R_h}^{(3)}(T)) \quad (15) \]

where \( Z_{R_h}^{(3)}(T) \) is the output of the h-th rule node in layer 3 that is connected to \( O_{Term_{i|h}} \) as its consequent.

**Step 5: Defuzzifying**

To defuzzify the derived fuzzy outputs, the weighted center of averaging (COA) technique [26] is used. So, the activation function in layer 5 is as follows:

\[ y_i(t) = Z_{i|h}^{(5)}(t) = f_{i|h}^{(5)}(z_{i|h}^{(4)}(T), j = (1, 2, ..., h_j)) = \]

\[ \frac{\sum_{j=1}^{h_j} z_{i|h}^{(4)}(T) \times p_{ji}}{\sum_{j=1}^{h_j} z_{i|h}^{(4)}(T)} \quad (16) \]

where \( p_{ji} = \frac{u_{ji} + v_{ji}}{2} \) and \( y_i(T) \) is the output of i-th node \( O_{Term_i} \) in layer 5.

**Step 6: Error evaluation**

During the feed forward phase, the input vector \( X(T) \) is presented to the network and the output vector \( Y(T) \) results in T-th epoch. At the beginning of the feed backward pass, \( Y(T) \) is compared against the desired output vector \( D(T) \) and the error is used to modify the vector \( \alpha(T) \) in T-th epoch. Regarding the framework of imbalanced data-sets, the geometric mean presented in (2) should be used to calculate the error for all outputs. The following equation denotes the error in the t-th epoch.

\[ \text{Error}(T) = (1 - G(T)) = \]

\[ (1 - \sqrt{\frac{TP(T)}{TP(T) 	imes FN(T)} \times \frac{TN(T)}{FP(T) 	imes TN(T)}}) \quad (17) \]

where \( G(T) \) is the accuracy of model till T-th epoch of training.

**Step 7: Modifying t-norm**

All links in the proposed network do not have a particular weight which affects the error in the feedback phase. The first issue to be taken into consideration in order to achieve better accuracy is the significance of antecedents that make inaccuracy should be reduced. Dubios t-norm performs like minimum t-norm if the following equation (15) becomes satisfied and Dubios t-norm approaches to product t-norm, which reduces the significance of the antecedent causing inaccuracy, by increasing \( \alpha_h(T) \). Hence, better prediction can be achieved by increasing \( \alpha_h(T) \) of rule \( R_h \) which participates in producing \( \text{Error}(T) \).

\[ Z_{R_h}^{(2)}(T) \geq \alpha_h(T) \quad (18) \]

To modify the parameter \( \alpha_h(T) \) in each iteration (19) is used.

\[ \alpha_h(T + 1) = \alpha_h(T) - \sigma \times (\text{Error}(T) \times W_h) \quad (19) \]

where \( \sigma \) is a positive experimental consistent less than 1.

V. CONCLUSION AND FURTHER STUDY

Although many statistical and soft computing models and methods have been applied to the study of failure prediction, they can not explicitly explain the implicit and intrinsic relationship between financial covariates and failure phenomena, and they suffer from different deficiencies which make them inapplicable. This paper proposes an approach to classify and predict failure more accurately, as well as generate useful knowledge describing the influence of selected financial variables on failure. The proposed approach which utilizes SMOTE in the preprocessing phase, and an adaptive fuzzy inference scheme whose parameters are learnt in the learning phase, can remarkably improve prediction accuracy.

Applying the real data sets to run the proposed approach and analysis the experimental results including accuracy, standard deviation and fuzzy rules and, comparing the results with other popular FNN failure prediction models [7, 14] will be the next step in this studies.

Since knowledge is the fundamental base of a decision making framework, the rule formulation and knowledge...
creation ability of the proposed approach makes it a practical and useful component of a Decision Support System (DSS) in the finance industry. Establishing a DSS which applies the proposed approach is a challenging research area requiring further scientific research.

Likewise, as failure prediction incorporates an imbalanced data-sets problem, the information granulation concept and techniques can be applied to make a hierarchical fuzzy rule base which can bring about a significant improvement. This is another interesting research direction to be taken into account in the future.

ACKNOWLEDGEMENT

The work presented in this paper was supported by Australian Research Council (ARC) under the discovery project DP088739.

The authors would like to thank Dr Juan Yao from Faculty of Economics and Business in The University of Sydney for her consultation and advice in this study.

REFERENCES


