A Comparative Study of Hierarchical ANFIS and ANN in Predicting Student Academic Performance

QUANG HUNG DO [1], JENG-FUNG CHEN [2,*]
Department of Industrial Engineering and Systems Management
Feng Chia University
No. 100 Wenhwa Rd., Taichung, Taiwan, R.O.C. 40724
TAIWAN
[1] quanghung2110@gmail.com, [2] chenjengfung@gmail.com
* Corresponding author

Abstract: -The accurate prediction of student academic performance is of importance to institutions as it provides valuable information for decision making in the admission process and enhances educational services by allocating customized assistance according to the predictions. The purpose of this study is to investigate the predictive ability of two models: the hierarchical ANFIS and ANN. We used previous exam results and other factors, such as the location of the student's high school and the student's gender, as input variables, and predicted the student's expected performance. The simulation results of the two models were then discussed and analyzed. It was found that the hierarchical ANFIS model outperformed the ANN model in the prediction of student academic performance. These results show the potential of the hierarchical ANFIS model as a predictor. It is expected that this work may be used to assist in student admission procedures and strengthen the service system in educational institutions.

Key Words: **-** Adaptive Neuro-Fuzzy Inference System, Artificial neural network, Prediction, Student academic performance, Higher education

1 Introduction

Predicting student academic performance is necessary in educational institutions. When admission officers review applications, accurate predictions help them to distinguish between suitable and unsuitable candidates for an academic program and identify candidates who would likely do well in the university. The failure to perform an accurate admission decision may result in an unsuitable student being admitted to the university. The quality of an educational institution is mainly reflected in its research and training. Hence, the quality of admitted candidates affects the quality level of institutions. Moreover, the results obtained from the prediction of academic performance may be also used for classifying students, which enables educational managers to offer them additional support such as customized assistance and tutoring resources. The results of prediction can also be used by lecturers to specify the most suitable teaching actions for each group of students and provide them with further assistance tailored to their needs. Accurate prediction of student achievement is one way to enhance the quality level and provide better educational services. Thus, developing an accurate prediction tool is very important for educational institutions.

Soft computing techniques have been recognized as attractive alternatives to the standard, well-established hard computing paradigms. Soft computing techniques, which emphasize gains in understanding system behavior in exchange for unnecessary precision, have been proven to be able to efficiently solve complicated problems. Soft computing techniques have also enabled the development of more efficient models which predicts student academic performance more accurately than previously possible. There are several soft computing techniques used in this field: neural networks, decision tree, and k-mean clustering [1]. Among these, the artificial neural network (ANN) has been widely used. Kanakana and Olanrewaju [2] utilized a multilayer perception neural network to predict student performance. They used the average point scores of grade 12 students as inputs and the first year college results as output. Their research showed that an ANN-based model is able to predict student performance in semester one with high accuracy. Lykourentzou et al. [3] used a multiple feed-forward neural network to predict students' final achievements and to group them into two groups. In their work, a student achievement prediction method was applied to a 10-week course. The results showed that accurate prediction is possible at an early stage, more specifically, at the third week of the 10-week course. Oladokun et al. [4] used an ANN model to predict the performance of a candidate being considered for admission into university. Their results indicated that the ANN model is able to correctly predict the performance of more than 70% of the prospective students. The aforementioned studies reveal the great success of neural networks in the prediction of students' academic performance.

The ANFIS, a hybrid intelligent system, is a combination of artificial neural networks and fuzzy systems; therefore, it has the advantages of both methods [5, 6, 7]. Fuzzy systems are appropriate if sufficient expert knowledge about the process is available, while neural networks are useful if sufficient process data is available or measurable. ANFIS can effectively solve non-linear problems [5] and is particularly useful in applications where classical approaches fail or are too complicated to be used. ANFIS has been widely used in different fields such as economics, medicine, and engineering. Some researchers have applied ANFIS in the area of education. Taylan and Karagözoglu [8] introduced an approach based on ANFIS for the design of a fuzzy inference system to assess student academic performance. In their study, quiz, major, midterm, final and performance appraisals were used as inputs, and the output was a student's academic performance. In the ANFIS model, they used a real dataset obtained from the students' achievements in the engineering economy course. The results showed that the ANFIS model was able to produce the same outputs as the statistical method. Norazah et al. [9] employed ANFIS to provide a way to determine a student's learning achievement. They used four inputs, namely, score, time, attempts, and help, to classify student performance into three categories. In their work, a complete fuzzy rule base was reduced to a concise fuzzy rule base. This approach was able to determine important attributes that could be used to represent the decision system. The great success of ANFIS motivated us to utilize it in our research.

The objective of this study is to evaluate the performance of hierarchical ANFIS and ANN models in the prediction of student academic performance. As far as we are aware, the application of new approaches in the prediction of student academic performance is not well documented in the existing literature. In addition, the application of this artificial intelligence method (hierarchical ANFIS) is applied for the first time in this research. The paper is organized into five sections. After the introduction in Section 1, the ANN and hierarchical ANFIS models are presented in Section 2 and Section 3, respectively. Section 4 is dedicated to describing the research design. The results and discussion are given in Section 5. Finally, the conclusion is discussed in Section 6.

2 Artificial Neural Networks

ANNs are the form of artificial intelligence which is based on the function of human brain and nervous system. An artificial neural network has two types of basic components, namely, neuron and link. A neuron is a processing element and a link is used to connect one neuron with another. Each link has its own weight. Each neuron receives stimulation from other neurons, processes the information, and produces an output. Neurons are organized into a sequence of layers. The first and the last layers are called input and output layers, respectively, and the middle layers are called hidden layers. The input layer is a buffer that presents data to the network. It is not a neural computing layer because it has no input weights and no activation functions. The hidden layer has no connections to the outside world. The output layer presents the output response to a given input. The activation coming into a neuron from other neurons is multiplied by the weights on the links over which it spreads, and then is added together with other incoming activations. A neural network in which activations spread only in a forward direction from the input layer through one or more hidden layers to the output layer is known as a multilayer feed-forward network. For a given set of data, a multi-layer feed-forward network can give a good non-linear relationship. Studies have shown that a feed-forward network even with only one hidden layer can approximate any continuous function [10, 11]. Therefore, a feed-forward network is an attractive approach [12]. Fig.1 shows an example of a feed-forward network with three layers. In Fig.1, *R*, *N*, and *S* are the numbers of input, hidden neurons, and output, respectively; *iw* and *hw* are the input and hidden weights matrices, respectively; *hb* and *ob* are the bias vectors of the hidden and output layers, respectively; *x* is the input vector of the network; *ho* is the output vector of the hidden layer; and *y* is the output vector of the network. The neural network in Fig.1 can be expressed through the following equations:

$$ho_i = f\left(\sum_{j=1}^{R} iw_{i,j}.x_j + hb_i\right), \quad for\ i=1,..,N \tag{1}$$

$$y_i = f\left(\sum_{k=1}^{N} hw_{i,k}.ho_k + ob_i\right), \quad for\ i=1,..,S \tag{2}$$

where f is an activation function.

When implementing a neural network, it is necessary to determine the structure in terms of number of layers and number of neurons in the layers. The larger the number of hidden layers and nodes, the more complex the network will be. A network with a structure that is more complicated than necessary over fits the training data [13]. This means that it performs well on data included in the training set, but may perform poorly on that in a testing set.

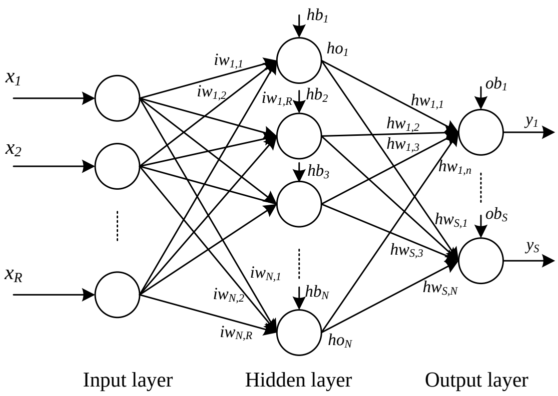

Fig.1: A feed-forward network with three layers.

Once a network has been structured for a particular application, it is ready for training. Training a network means finding a set of weights and biases that will give desired values at the network's output when presented with different patterns at its input. When network training is initiated, the iterative process of presenting the training data set to the network's input continues until a given termination condition is satisfied. This usually happens based on a criterion indicating that the current achieved solution is good enough to stop training. Some of the common termination criteria are sum squared error (SSE) and mean squared error (MSE). Through continuous iterations, the optimal or near-optimal solution is finally achieved, which is regarded as the weights and biases of a neural network. Suppose that there are m input-target sets, $x_k - t_k$ for $k=1, 2, \ldots, m$, for neural network training. Thus, network variables arranged as iw, hw, hb, and ob are to be changed to minimize a cost function. E, such as the MSE between network outputs, y_k, and desired targets, t_k, is as follows:

$$MSE = \frac{1}{m}\sum_{k=1}^{m} e_k^2 = \frac{1}{m}\sum_{k=1}^{m}(t_k - y_k)^2 \qquad (3)$$

3 Hierarchical Adaptive Neuro-Fuzzy Inference System

Neural networks and fuzzy set theory, which are soft computing techniques, are tools for establishing intelligent systems. A fuzzy inference system (FIS) employs fuzzy if-then rules when acquiring knowledge from human experts to deal with imprecise and vague problems [9]. FISs have been widely used in many applications including optimization, control, and system identification. A simple FIS is presented in Fig.2. However, fuzzy systems cannot learn from or adjust themselves [14]. A neural network has the capacity to learn from its environment, self-organize, and adapt in an interactive way. For these reasons, a neuron-fuzzy system, which is the combination of a fuzzy inference system and neuron network, has been introduced to produce a complete fuzzy rule base system [15, 16]. The advantage of neural networks and fuzzy systems can be integrated in a neuron-fuzzy approach. Fundamentally, a neuron-fuzzy system is a fuzzy network that has its function as a fuzzy inference system. The system can overcome some limitations of neural networks, as well as the limits of fuzzy systems [17, 18], when it has the capacity to represent knowledge in an interpretable manner and the ability to learn. The details of the neuron-fuzzy system were proposed by Takagi and Hayashi [19]. Among the neuron-fuzzy systems, ANFIS, introduced by Jang [15], has been one of the most common tools. In the FIS, the fuzzy if-then rules are determined by experts, whereas in the ANFIS, it automatically produces adequate rules with respect to input and output data, and facilitates the learning capabilities of neural networks.

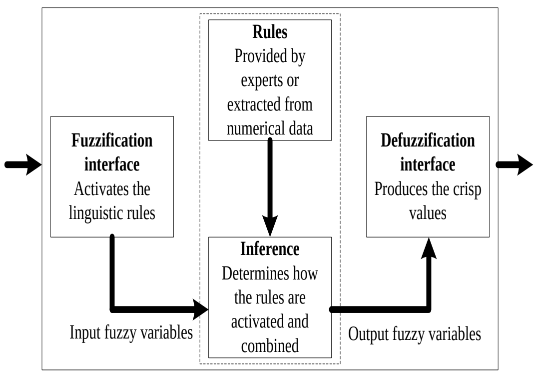

Fig.2: A simple fuzzy inference system.

ANFIS is a multilayer feed-forward neural network, which employs neural network learning algorithms and fuzzy reasoning to map from input space to output space. The architecture of ANFIS includes five layers, namely, the fuzzification layer, the rule layer, the normalization layer, the defuzzification layer, and a single summation node. To present the ANFIS architecture and simplify the explanations, assume that the FIS has two inputs, x_1 and x_2, two rules, and one output, y, as shown in Fig.3. Each node within the same layer performs the same function. The circles are used to indicate fixed nodes while the squares are used to denote adaptive nodes. A FIS has two inputs and two fuzzy if-then rules [20] that can be expressed as:

Rule 1: If x_1 is A_1 and x_2 is B_1 then $y_1=p_1x_1+q_1x_2+r_1$,
Rule 2: If x_1 is A_2 and x_2 is B_2 then $y_2=p_2x_1+q_2x_2+r_2$, (4)

where x_1 and x_2 are the inputs; A_1, A_2, B_1, B_2 are the linguistic labels; p_i, q_i, and r_i, (i=1 or 2) are the consequent parameters [15] that are identified in the training process; and y_1 and y_2 are the outputs within the fuzzy region. Eq. (4) represents the first type of fuzzy if-then rules, in which the output part is linear. The output part can also be constants [21], represented as

Rule 1: If x_1 is A_1 and x_2 is B_1 then $y_1=C_1$,
Rule 2: If x_1 is A_2 and x_2 is B_2 then $y_2=C_2$, (5)

where C_i (i=1 or 2) are constant values. Eq. (5) represents the second type of fuzzy if-then rules.

For complicated problems, the first type of if-then rules is widely utilized to model the relationships of inputs and outputs [22]. In this research, we also used a linear function for the output.

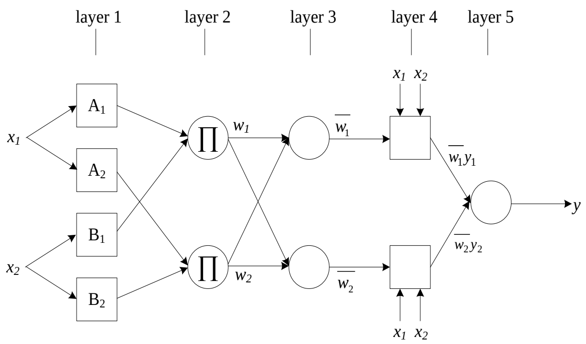

Fig.3: An ANFIS architecture of two inputs and two rules.

The brief description of the functions of each layer is as follows:

Layer 1 - fuzzification layer: Every node in this layer is a square node. The nodes produce the membership values. Outputs obtained from these nodes are calculated as follows:

$O_{1,i}=\mu_{Ai}(x_1)$ for i=1, 2 or
$O_{1,i}=\mu_{Bi-2}(x_2)$ for i=3, 4, (6)

where $O_{1,i}$ denotes the output of node i in layer 1, and $\mu_{Ai}(x_1)$ and $\mu_{Bi-2}(x_2)$ are the fuzzy membership functions of A_i and B_{i-2}. The fuzzy membership functions can be in any form, such as triangular, trapezoidal, or Gaussian functions.

Layer 2 - rule layer: Every node in this layer is a circular node. The output is the product of all incoming inputs.

$O_{2,i}=w_i=\mu_{Ai}(x_1)\times \mu_{Bi}(x_2)$ for i=1, 2, (7)

where $O_{2,i}$ denotes the output of node i in layer 2, and w_i represents the firing strength of a rule.

Layer 3 - normalization: Every node in this layer is a circular node. Outputs of this layer are called normalized firing strengths. The ith node is calculated by the ith node firing strength to the sum of all rules' firing strengths.

$$O_{3,i}=\overline{w_i}=\frac{w_i}{w_1+w_2} \text{ for } i=1, 2, \quad (8)$$

where $O_{3,i}$ denotes the output of node i in layer 3, and $\overline{w_i}$ is the normalized firing strength.

Layer 4 - defuzzification layer: Every node in this layer is an adaptive node with a node function.

$$O_{4,i}=\overline{w_i}\,y_i \text{ for } i=1, 2, \quad (9)$$

where $O_{4,i}$ denotes the output of node i in layer 4, $\overline{w_i}$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are consequent parameters of the Sugeno fuzzy model.

Layer 5 - a single summation node: The node is a fixed node. This node computes the overall output by summing all the incoming signals from the previous layer:

$$O_{5,i}=\sum_i \overline{w_i}\,y_i=\frac{\sum_i w_i y_i}{\sum_i w_i} \text{ for } i=1, 2, \quad (10)$$

where $O_{5,i}$ denotes the output of node i in layer 5. The results are then defuzzified using a weighted average procedure.

It can be seen that the ANFIS architecture has two adaptive layers: layer 1 and layer 4. Layer 1 has parameters related to the fuzzy membership functions and layer 4 has parameters $\{p_i, q_i, r_i\}$ related to the polynomial. The aim of the hybrid learning algorithm in the ANFIS architecture is to adjust all these parameters in order to make the output match the training data. Adjusting the parameters includes two steps. In the forward pass of the learning algorithm, the premise parameters are fixed, functional signals go forward until layer 4, and the consequent parameters are identified by the least squares method to minimize the measured error. In the backward pass, the consequent parameters are fixed, the error signals go backward, and the premise parameters are updated by the gradient descent method [23]. This hybrid learning algorithm is able to decrease the complexity of the algorithm and increase the learning efficiency [24]. Due to this advantage of the hybrid learning algorithm, it was utilized in this study.

According to Güneri [25], too many inputs in the ANFIS structure makes the system complicated and limits its applicability. In addition, many studies pointed out that ANFIS gives better solutions with a simple structure. To deal with this issue, several low-dimensional rule bases should be arranged in a hierarchical structure [26]. To model a hierarchical ANFIS, it is necessary to identify: a suitable hierarchical structure, the number of inputs for each sub-ANFIS model, and a rule base for each sub-ANFIS model.

When identifying the rule base for ANFIS, the problems under consideration are: (1) there are no standard methods for transforming human knowledge or experience into the rule base; and (2) it is necessary to tune the membership functions to maximize the performance and minimize the errors [15]. There are several methods to identify FIS. In this paper, the grid partition method was utilized. This method divides the data space into rectangular sub-spaces using an axis-paralleled partition based on the number of membership functions and their types in each dimension. An example of a grid partition with two input variables and two membership functions for each input variable is illustrated in Fig.4. The combination of a grid partition and ANFIS has been mentioned by Kennedy et al. [27]. The grid partition is suitable for problems with a small number of inputs [22].

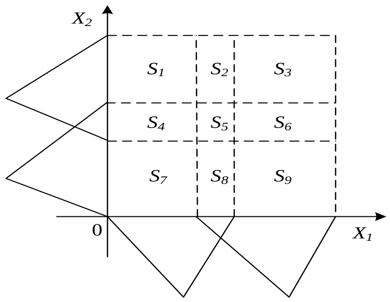

Fig.4: An example of grid partition method with two input variables.

4 Research Design

This section applies the models to the prediction of student academic performance. In this paper, an application related to the context of Vietnam was used as an illustration.

4.1 Identifying Input and Output Variables

Through a literature review and discussion with admission officers and experts, a number of academic, social-economic, and other related factors that are considered to have influence on the student academic performance were determined and chosen as input variables. The input variables were obtained from the admission registration profile and are as follows: university entrance exam results (normally, in Vietnam, candidates take three exams for the fixed group of subjects they choose), the overall average score from a high school graduation examination, elapsed time between graduating from high school and obtaining university admission, location of high school (there are four regions, as defined by the government of Vietnam: Region 1, Region 2, Region 3 and Region 4. Region 1 includes localities with difficult economic and social conditions; Region 2 is rural areas; Region 3 consists of provincial cities; and Region 4 includes central cities), type of high school attended (private or public), and gender (male or female). The output variable is the average score of the first academic year in college. It is based on the current grading system used by the university. In our study, the domain of output variables has its range from 0 to 10 (nominal scores). The output variable can be changed to predict student performance in their first through fourth year at an academic institution. In brief, there are eight input variables and one output variable in the proposed model.

The input and output variables were pre-processed to be suitable for the prediction model. The input variables and ranges are presented in Table 1. The output variable has a range from 0 to 10.

Table 1: Input variables

Input variable		Range
University entrance examination score	Subject 1	0.5-10
	Subject 2	0.5-10
	Subject 3	0.5-10
The average overall score of high school graduation examination		5-10
Elapsed time between graduating from high school and obtaining university admission (0 year: 0; 1 year: 1; 2 years: 2; and 3 years – above: 3)		0, 1, 2, 3
Location of student’s high school (Region 1: 0; Region 2: 1; Region 3: 2; Region 4: 3)		0, 1, 2, 3
Type of high school attended (private:0; public:1)		0, 1
Student’s gender (male:0; female:1)		0, 1

4.2 Data Set

We obtained our data from the University of Transport Technology, which is a public university in Vietnam. We used a real data set from students in the Department of Bridge Construction as input variables and their achievements from the 2011-2012 academic year as output variables. The data set consisted of 653 cases and was divided into two groups. The first group (about 60%) was used for training the model. The second group (about 40%) was employed for testing the model. The training data set served in model building while the other group was used for the validation of the developed model.

4.3 Development of the ANN Model

The structure of an ANN is dictated by the choice of the number in the input, hidden, and output layers. Each data set has its own particular structure, and therefore determines the specific ANN structure. The number of neurons comprised in the input layer is equal to the number of features (input variables) in the data. The number of neurons in the output layer is equal to the number of output variables. In this study, the data set includes eight input variables and one output variable; hence, the numbers of neurons in the input and output layers are eight and one, respectively. The three layer feed-forward neural network is utilized in this work as it can be used to approximate any continuous function [28, 29]. Regarding the number of hidden neurons, the choice of an optimal size of hidden layer has often been studied, but a rigorous generalized method has not been found [30]. In this paper, the choice was made through extensive simulation with different choices of the number of hidden nodes. For each choice, we obtained the performance of the concerned neural networks, and the number of hidden nodes providing the best performance was used for presenting results. The optimum number of neurons in the hidden layer was determined by varying their number, starting with a minimum of 1 and then increasing in steps by adding one neuron each time. Hence, various network architectures were tested to achieve the optimum number of hidden neurons. The best performing architecture was found to be 8-11-1, i.e., with one hidden layer and 11 neurons. In this paper, the activation function from input to hidden is sigmoid. With no loss of generality, a commonly used form, $f(n) = 2/(1+e^{-2n}) - 1$, was utilized, while a linear function was used from the hidden to the output layer.

The purpose of the training step is to reduce the errors, which are the difference between predicted and actual values, until the ANN learns the training data. Several algorithms, including gradient descent (GD), gradient descent with momentum (GDM), scaled conjugate gradient (SCG), and Lavenberg-Marquardt (LM), are available for training neural networks. It is difficult to know which algorithm is the most appropriate for a given problem since the type of problem, the data set, and the network architecture may affect the performance of a training algorithm [31]. In this study, the performance of several commonly used training algorithms was evaluated. Finally, the Lavenberg-Marquardt algorithm was chosen since it gave better performance than the others.

4.4 Development of the Hierarchical ANFIS Model

The number of input membership functions has a great influence on the ANFIS training process. If a model uses too many membership functions, the inference rules will be more complicated and the training time will increase. In our research, two membership functions were chosen for each input in the model. There are different types of membership functions that can be used in an ANFIS model, such as a triangular membership function, a trapezoidal membership function, a bell-shaped membership function, and a Gaussian membership function. Each membership function was tested separately to obtain the most effective model with minimum errors, and then the bell-shaped membership function was chosen to train input/output data. The bell-shaped membership function is specified by three parameters, $\{a, b, c\}$, and the membership value is derived by the formula:

$$\mu(x) = f(x; a, b, c) = 1 \Big/ \left(1 + \left|\frac{x-c}{a}\right|^{2b}\right), \qquad (11)$$

where b is positive, and c represents the center of the curve. These parameters are referred to as premise parameters or antecedent parameters.

As discussed earlier, a four-layer ANFIS structure was introduced to decrease the dimension of the rule base. Layers 1-3 have three input variables and layer 4 has two input variables; each layer has one output. Fig.5 represents the hierarchical ANFIS model, where x_1-x_8 are input variables and y represents one output.

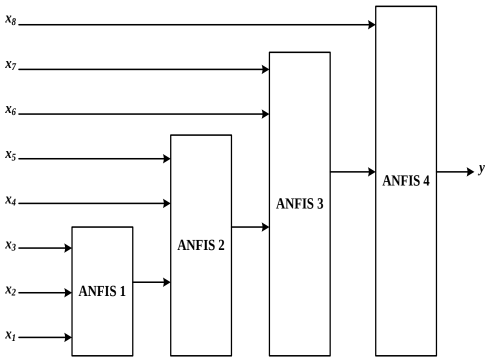

Fig.5: The hierarchical ANFIS model.

4.5 Model Performance Evaluation

To examine the performance of a prediction model, several criteria are used. These criteria are applied to the model to know how well it works. The criteria are used to compare predicted values and actual values. They are as follows:

Root mean squared error (RMSE): This index estimates the residual between the actual value and desired value. A model has better performance if it has a smaller RMSE. An RMSE equal to zero represents a perfect fit.

$$RMSE = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(t_k - y_k)^2}\ , \tag{12}$$

where t_k is the actual (desired) value, y_k is the predicted value produced by the model, and m is the total number of observations.

Mean absolute percentage error (MAPE): This index indicates an average of the absolute percentage errors; the lower the MAPE the better.

$$MAPE = \frac{1}{m}\sum_{k=1}^{m}\left|\frac{t_k - y_k}{t_k}\right| \tag{13}$$

Mean absolute error (MAE): This index indicates how close predicted values are to the actual values.

$$MAE = \frac{1}{m}\sum_{k=1}^{m}\left|t_k - y_k\right| \tag{14}$$

Correlation coefficient (R): This criterion reveals the strength of relationships between actual values and predicted values. The correlation coefficient has a range from 0 to 1, and a model with a higher R means it has better performance.

$$R = \frac{\sum_{k=1}^{m}\left(t_k - \bar{t}\right)\left(y_k - \bar{y}\right)}{\sqrt{\sum_{k=1}^{m}\left(t_k - \bar{t}\right)^2 . \sum_{k=1}^{m}\left(y_k - \bar{y}\right)^2}}\ , \tag{15}$$

where $\bar{t} = \frac{1}{m}\sum_{k=1}^{m} t_k$ and $\bar{y} = \frac{1}{m}\sum_{k=1}^{m} y_k$ are the average values of t_k and y_k, respectively.

5 Results and Discussion

The models were coded and implemented in the MATLAB environment and simulation results were then obtained. A 10-fold cross validation method was used to avoid over-fitting problem. The comparison between actual values and corresponding output values obtained by the hierarchical ANFIS and ANN model are shown in Fig.6 and Fig.7. These figures present scatter diagrams that illustrate the degree of correlation between predicted values and actual values. In each figure, an identity line was drawn as a reference. In a scatter diagram, the identity line represents that the two sets of data are identical. The more the two data sets agree, the more the points tend to concentrate in the vicinity of the identity line. Here the training data and testing data were used to test the two models. It may be observed that most predicted values are close to the actual values. This indicates a good agreement between the predicted values obtained by the two models and the actual values.

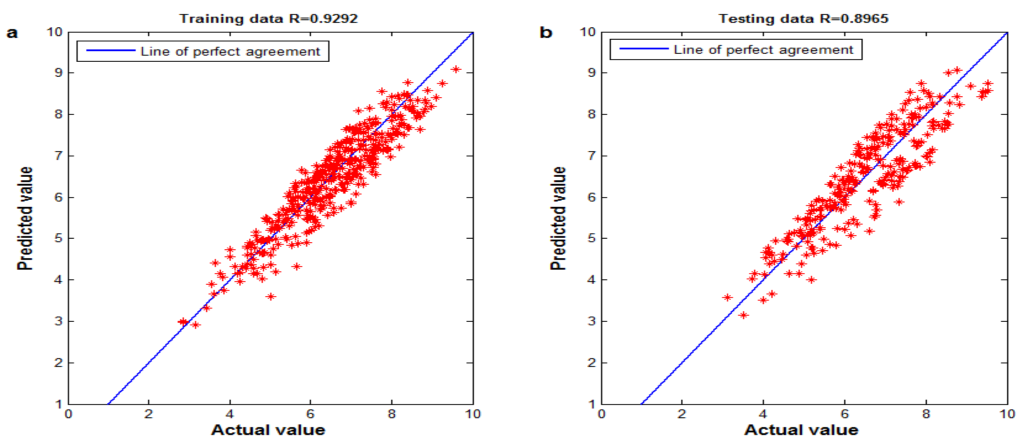

Fig.6: Comparison between actual and predicted values of the ANN model: a) training, b) testing.

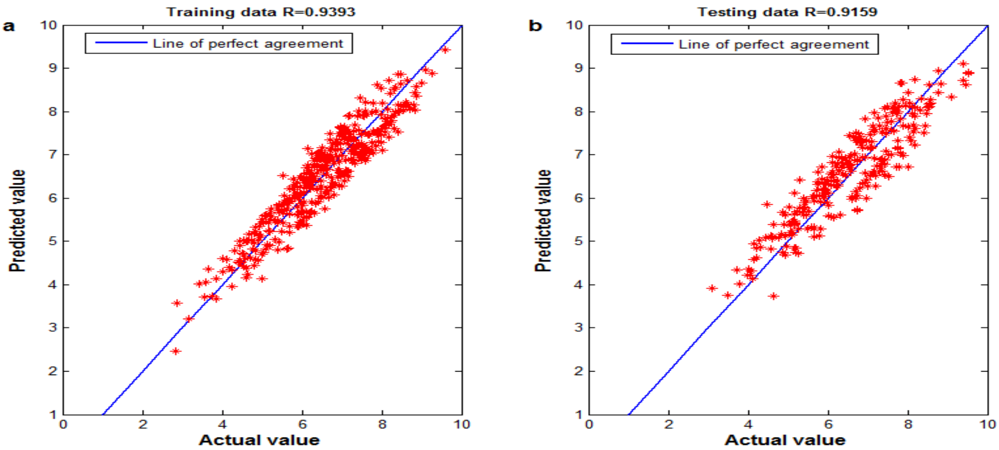

Fig.7: Comparison between actual and predicted values of the hierarchical ANFIS model: a) training, b) testing.

Table 2 represents the performance statistics of the hierarchical ANFIS and ANN model. Theoretically, a prediction model is accepted as ideal when MAPE, RMSE, and MAE are small and *R* is close to 1. As can be seen from Table 2, when compared with results from other studies on prediction, the obtained performance criteria values from the hierarchical ANFIS and ANN models are both satisfactory.

Table 2: Comparison of the ANN and the hierarchical ANFIS models

Model	Training data				Testing data			
	MAPE	RMSE	MAE	R	MAPE	RMSE	MAE	R
ANN	0.0605	0.4658	0.3901	0.9292	0.0751	0.5744	0.4876	0.8965
Hierarchical ANFIS	0.0593	0.4247	0.3763	0.9393	0.0715	0.5196	0.4498	0.9159

It is very clear from Table 2 that the hierarchical ANFIS model has a smaller MAPE, RMSE and MAE as well as a bigger *R* for both the training and testing datasets than those of the ANN model. When the MAPE values are considered, the hierarchical ANFIS model has significantly better accuracy rates than the ANN model in the training data (0.0593 vs. 0.0605) and in testing data (0.0715 vs. 0.0751). Also, RMSE and MAE values of the hierarchical ANFIS model are lower, showing better performance than the ANN model. The correlation coefficients (*R*) of the hierarchical ANFIS model are 0.9393 and 0.9159 for training and testing datasets, respectively; and both are above 0.9. The results indicate that the hierarchical ANFIS model achieves better performances than the ANN model.

Based on the obtained results, it can be concluded that both the hierarchical ANFIS and ANN models can be used to predict student academic performance. However, regarding prediction accuracy, the hierarchical ANFIS is highly appreciated. The hierarchical ANFIS model outperformed the ANN model, and the results show that its prediction outcome is more accurate and reliable. Hence, the hierarchical ANFIS may be acceptable and good enough to serve as a predictor of student academic performance. Moreover, an additional benefit of the hierarchical ANFIS model is the ability to provide the set of rules that can be used for decision making.

6 Conclusions

The accurate prediction of student academic performance is of importance for making admission decisions as well as providing better educational services. In this study, two models, the hierarchical ANFIS and ANN, to predict student academic performance were evaluated. The performance statistics of the hierarchical ANFIS model were compared against those of the ANN model in terms of MAPE, RMSE, MAE, and *R* achieved. The hierarchical ANFIS model was found to have better overall performance in all criteria. The findings demonstrated the remarkable advantage and the potential of the hierarchical ANFIS model in the prediction of student performance. The results of the present study also reinforce the fact that a comparative analysis of different approaches has been always supportive in developing a prediction model. It is expected that this work may be used as a reference for decision making in the admission process and to provide better educational services by offering customized assistance according to student predicted academic performance. For future research, to improve the prediction accuracy, some factors such as character, intelligence, and other psychological factors will be considered. Moreover, we will develop and incorporate our model in a user-friendly software tool for the prediction of student academic performance in order to make this task easier for educators.

Acknowlegement

This research was funded by the National Science Council of Taiwan under Grant No. NSC 102-2221-E-035-040.

References:

[1] C. Romero and S. Ventura, Educational data mining: A survey from 1995 to 2005, *Expert Systems with Applications*, Vol.33, No.1, 2007, pp. 135-146.

[2] G.M. Kanakana and A.O. Olanrewaju, Predicting student performance in engineering education using an artificial neural network at Tshwane university of technology, *Proceedings of the International Conference on Industrial Engineering, Systems Engineering and Engineering Management for Sustainable Global Development*, 2011, pp. 1-7.

[3] I. Lykourentzou, I. Giannoukos, G. Mpardis, V. Nikolopoulos and V. Loumos, Early and Dynamic Student Achievement Prediction in E-Learning Courses Using Neural Networks, *Journal of the American society for information science and technology*, Vol.60, No.2, 2009, pp. 372-380.

[4] V.O. Oladokun, A.T. Adebanjo, and O.E. Charles-Owaba, Predicting Students' Academic Performance using Artificial Neural Network: A Case Study of an Engineering Course, *The Pacific Journal of Science and Technology*, Vol.9, No.1, 2008, pp. 72-79.

[5] M. Buragohain and C. Mahanta, A novel approach for ANFIS modeling based on full factorial design, *Applied Soft Computing*, Vol.8, No.1, 2008, pp. 609-625.

[6] A. Azadeh, M. Saberi, M. Anvari, A. Azaron, and M. Mohammadi, An adaptive network based fuzzy inference system - genetic algorithm clustering ensemble algorithm for performance assessment and improvement of conventional power plants, *Expert Systems with Applications*, Vol.38, No.3, 2011, pp. 2224-2234.

[7] E.H. Metin and H. Murat, Comparative analysis of an evaporative condenser using artificial neural network and adaptive neuro-fuzzy inference system, *International Journal of Refrigeration*, Vol.31, No.8, 2008, pp. 1426-1436.

[8] O. Taylan and B. Karagözoglu, An adaptive neuro-fuzzy model for prediction of student's academic performance, *Computers & Industrial Engineering*, Vol.57, No.3, 2009, pp. 732-741.

[9] Y. Norazah, B.A. Nor, S.O. Mohd, and C.N. Yeap, *A Concise Fuzzy Rule Base to Reason Student Performance Based on Rough-Fuzzy Approach*, *Fuzzy Inference System–Theory and application*, Publisher: InTech, 2010.

[10] K. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks*, Vol.2, No.3, 1989, pp. 183-192.

[11] K. Hornik, M. Stinchcombe, and H. White, Multilayer feed-forward networks are universal approximators, *Neural Networks*, Vol.2, No.5, 1989, pp. 359-366.

[12] M.R.O. Norgaard, N.K. Poulsen, and L. K. Hansen, *Neural networks for modeling and control of dynamic systems, A practitioner's handbook*, Springer, London, 2000.

[13] R. Caruana, S. Lawrence, and C. L. Giles, Over-fitting in neural networks: back propagation, conjugate gradient and early stopping, *Proceedings of Advances Neural Information Processing Systems 13*, USA, 2001, pp. 402-408.

[14] R. Ata and Y. Kocyigit, An adaptive neuro-fuzzy inference system approach for prediction of tip speed ratio in wind turbines, *Expert Systems with Applications*, Vol.37, No.7, 2010, pp. 5454-5460.

[15] J.S. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference, *IEEE Transaction on Systems Man and Cybernetics*, Vol.23, No.3, 1993, pp. 665-685.

[16] K.S. Pal and S. Mitra, *Neuro-fuzzy pattern recognition: Methods in soft computing*, New York: Wiley, 1999.

[17] F.K. Nauck and R. Kruse, *Foundation of Neuro-Fuzzy systems*, John Wiley & sons Ltd, 1997.

[18] R. Singh, R. Kainthola, and T.N. Singh, Estimation of elastic constant of rocks using an ANFIS approach. *Applied Soft Computing*, Vol.12, No.1, 2012, pp. 40-45.

[19] H. Takagi and I. Hayashi, NN-driven fuzzy reasoning, *International Journal Approximate Reasoning*, Vol.5, No.3, 1991, pp. 191-212.

[20] T. Takagi and M. Sugeno, Derivation of fuzzy control rules from human operator's control actions. *IFAC symposium fuzzy information knowledge representation and decision analysis*, 1983, pp. 55-60.

[21] M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Science, Amsterdam, the Netherlands, 1985.

[22] M. Wei, B. Bai, A.H. Sung, Q. Liu, J. Wang, and M.E. Cather, Predicting injection profiles using ANFIS, *Information Sciences,* Vol.177, No.20, 2007, pp. 4445-4461.

[23] J.S. Jang, C.T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*, New Jersey: Matlab Curriculum Series Prentice Hall, 1997.

[24] T.N. Singh, R. Kanchan, A.K. Verma, and K. Saigal, A comparative study of ANN and neuro-fuzzy for the prediction of dynamic constant of rockmass, *Journal of Earth System Science*, Vol.114, No.1, 2005, pp. 75-86.

[25] A.F. Güneri, T. Ertay, and A. Yücel, An approach based on ANFIS input selection and modeling for supplier selection problem, *Expert Systems with Applications*, Vol.38, No.12, 2011, pp. 14907-14917.

[26] M. Brown, K. Bossley, D. Mills, and C. Harris, High dimensional neuro-fuzzy systems: overcoming the curse of dimensionality, *Proceedings of International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, 1995, pp. 2139-2146.

[27] P. Kennedy, M. Condon, and J. Dowling, Torque-ripple minimization in switched reluctant motors using a neuro-fuzzy control strategy, *Proceedings International Conference on Modeling and Simulation*, 2003.

[28] K. Hornik, M. Stinchombe, and H. White, Universal Approximation of an unknown Mapping and its Derivatives Using Multilayer Feed-forward Networks, *Neural Networks*, Vol.3, No.5, 1990, pp. 551-560.

[29] G. Cybenko, Approximation by superposition of a sigmoid function, *Mathematics of Control, Signal, and System*, Vol.2, 1989, pp. 303-314.

[30] U. Bhattacharya and B. B. Chaudhuri, Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed Numerals, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.31, No.3, 2009, pp. 444-457.

[31] Z. Rafik, B. Ridha, and R. Daniel, Levenberg-Marqurdt learning neural network for adaptive pre-distortion for time varying HPA with memory in OFDM systems, *Proceedings of 16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, 2008.