SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 4, ISSUE 1
ISSN 2054-7390

Implications of System Identification Techniques on ANFIS E-learners Activities Models-A Comparative Study

[1]Isiaka Rafiu Mope, [2]Omidiora Elijah Olusayo, [3]Olabiyisi Stephen O., [4]Okediran Oladotun O., and [5]Babatunde Ronke Seyi

[1,5] Department of Computer Science, College of Information and Communication Technology, Kwara State University, Malete, Ilorin, Nigeria;

[2,3,4] Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria;

abdulrafiu.isiaka@kwasu.edu.ng; eoomidiora@lautech.edu.ng; soolabiyisi@lautech.edu.ng; oookediran@lautech.edu.ng; ronke.babatunde@kwasu.edu.ng

ABSTRACT

Efficient e-learners activities model is essential for real time identifications and adaptive responses. Determining the most effective Neuro- Fuzzy model amidst plethora of techniques for structure and parameter identifications is a challenge. This paper illustrates the implication of system identification techniques on the performance of Adaptive Network based Fuzzy Inference System (ANFIS) E-learners Activities models. Expert knowledge and Historical data were used to formulate the system and their performances were compared. Similarly, comparison was made between memberships functions selected for Historical data identification. The efficiencies of the simulated models in MATLAB editor were determined using both classification uncertainty metrics and confusion matrix–based metrics. The classification uncertainty metrics considered are Mean Absolute Error (MAE) and Root Mean-Squared Error (RMSE). The confusion matrix-based metrics used are Accuracy, Precision and Recall. It was discovered that the model based on Experts Knowledge after training outperformed those based on Historical Data. The performances of the Membership Functions after ranking are Sigmoid, Gaussian, Triangular and G-Bells respectively.

Keywords: Neuro Fuzzy Model; E-learners Activities; System Identification Technique; Dataset Normalization.

1 Introduction

Adoption of Artificial Intelligent (AI) techniques for modelling complex problems is gaining prominence over mathematical models and traditional statistics [9]. Also, within the AI contemporary researches, the Soft computing techniques is becoming the most implemented when compared with the Hard computing techniques. Reasons adduced to this phenomenon essentially are the inherent features of the former which include tolerance of imprecision, incomplete or corrupt input data, capability to solve problems through repeated observation and adaptation [4], others are inspiration by natural processes and availability of simulation tools. The Soft computing technique which is also known as the Scruffy

DOI: 10.14738/tmlai.41.1799
Publication Date: 21st February, 2016
URL: http://dx.doi.org/10.14738/tmlai.41.1799

Techniques is less provable but are yielding useful and significant results [8]. The technique is the fusion of Fuzzy Computing, Evolutionary computing, Artificial Neural Networks and Probabilistic computing [15].

The Soft Computing techniques have their individual strengths and weaknesses which could be optimised by hybridization. Neuro Fuzzy techniques for instance integrate the learning capability, generalization capacity and standard architecture of Artificial Neural Networks (ANN) with the linguistic rule base and explicit internal operations of Fuzzy Inference System (FIS) [10]. The Hybrid intelligent systems built on these techniques have proven to be one of the best solutions in data modelling due to their capability to reason and learn in an environment of uncertainty and imprecision [3]. However, being a Scruffy technique, researches are being intensified at determining the most efficient approaches for optimizing the various operational components.

Identification of appropriate structure of ANN that could maximally model a scenario is a challenge, most a times the Trial and Error approach are adopted [1]. It could by incremental approaches [5], using number of rules, number of hidden layers or input clustering techniques [2]. This limitation could be ameliorated with a fusion of the ANN with FIS to form a hybrid Neuro Fuzzy model. The FIS built for the problem in accordance with available human expert knowledge suggests the structure for the ANN. Though the FIS model on its own may not be optimized due to human incapability to identify and or represent all possible instances. Even more that, no standard methods exist for transforming human knowledge or experience into the rule base and database of FIS [14]. Nonetheless, the fused ANN optimizes the system by adjusting the parameters during training. Among other Neuro Fuzzy techniques that have been implemented, ANFIS seems to be the most widely applied [7]. Its application specifically and that of ANN generally covers areas such as Function Fitting, Pattern recognition, Data clustering and Time series analysis.

This work demonstrates the implementation of this approach, the implication of each stage from formulation of FIS, transformation to ANFIS, the training and the effects of varied membership functions on the performance of E-Learners Activities Model.

2 Modelling E-Learner's Concentration and Exploration

E-Learners modelling are commonly meant for performance prediction or adaptive system. This study is a diversion from the common trends, in that it modelled quality of learner's experiences on a course. It was premised on the fact that formal knowledge requires formal experience [13]. Hence, the extent of knowledge acquired could be presumed from the quality and quantity of experiences acquired. The two behaviours considered were the learner's extent of content exploration and level of concentration. These behaviours are composite and of low bandwidth that cannot be measured directly, they can only be deduced from simple and measurable activities [11]. For actualization, the learner's Exploration was measured using the time spent on the course and extent of course content coverage. Other variables used for measuring the Concentration are Participation Index and Diagnosis Assessment Grade.

3 Methodology

3.1 Adoption of Expert Knowledge Techniques for Model Identification

Expert knowledge which is one of the common approaches for system identification in user modelling was employed as the base. The Expert knowledge based fuzzy system was built as described in [11], [6]. There, the number (N) of observable behaviours for modelling was two. Given as:

N = 2 = {N1, N2} = {Exploration, Concentration}, where N1 (Exploration) = {B1, B2} = {total time spent reading, content completion status} and N2 (Concentration) = {B3, B4} = {the Participation Index, diagnosis assessment remarks}.

Model Term set $\mathrm{T} = \{\mathrm{T}(\mathrm{B}_1), \mathrm{T}(\mathrm{B}_2), \mathrm{T}(\mathrm{B}_3), \mathrm{T}(\mathrm{B}_4)\}$ where

T(B1) = {total time spent reading} = {L,M,H} = B1f1 = 3 linguistic values,

T(B2) = {content completion status} = {NA, A} = B2f2 = 2 linguistic values,

T(B3) = {the Participation Index } = {L,M,H} = B3f3 = 3 linguistic values and

T(B4) = {diagnosis assessment remarks} = {U, S} = B4f4 = 2 linguistic values.

Hence, the model's term set is T(N) = {(L,M,H), (NA, A), (L,M,H), (U, S)}.

The Cartesian product of this (3 x 2 x 3 x 2) premises generated 36 rules and 36 consequents.

However, in order to provide training capability for the model such that it could adjust to the subjective decisions of various tutors on students' activities. The fuzzy student model was implemented in a connectionist adaptive network as Hybrid/Fussed Neuro-Fuzzy System. The fuzzed neural network was built on the following principles as in [12].

I. The number of cells in the input layer is equal to the number of input values which is four.
II. The number of cells in the fuzzification layer is equal to the number of fuzzy set which is ten
III. The number of cells in the premise layer, normalization layer and consequent layer is equal to the number of rules which is thirty six
IV. The output layer has a single node for the final inference

The structure of the neural network for implementing the fuzzy student model for learning activities quantitative and qualitative exploration is shown in Figure 1.

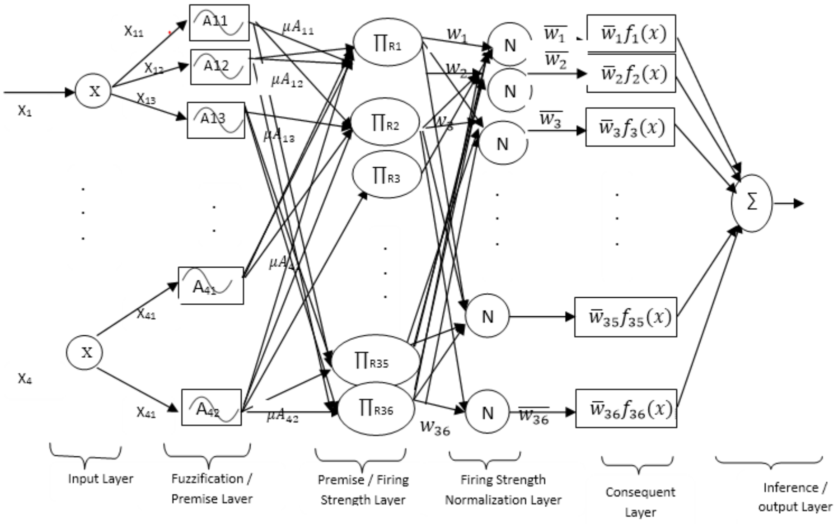


Figure 1: The Neuro-Fuzzy Model for Student Exploration and Concentration

The Neuro Fuzzy model shown is a six layer feed-forward hybrid network. Being an adaptive network, it consists of nodes and directional links. The network is adaptive because some of the nodes have adjustable parameters; such nodes are indicated with circular/oval shape in the network. Other nodes in square /rectangle shape are fixed nodes.

The Input layer (L1) provides the system with the base value for each of the features contributing towards the students evaluation. For instance the total time spent on the course may be based on seconds say 10800s for a 3 hour course. So the input layer send out the external crisp values to the next fuzzification layer without any processing. Hence, the neurons in this layer passively transmit the external input (crisp) signals to the next layer. The output of layer1 is given as:

$$O_i^1 = x_i \tag{1}$$

The total number of neurons in the input layer is four (4), one each for four linguistic variables B1-B4. Hence, for every instance (a student), four crisp values will be available as input for the purpose of determining the status of the student.

The Fuzzification layer (L2) is the first hidden layer; it has node for every categories of expert's description and classification for each of the variables. Neurons in this layer represent antecedent fuzzy sets of the fuzzy rules. This design adopts low, medium, and high fuzzy set, for B1 and B3. It uses Non Adequate (NA), Adequate (A), Unsatisfactory (US) and Satisfactory (S) for B2 and B4 respectively. For each input value entering the system, the membership degree to which such input belongs will be estimated. Common functions are Bell shape, Triangular, Gaussian and Sigmoid. For this work the Gaussian transfer function was used because of its amenability to training via adjustment of the parameters.

$$\text{The Gaussian MF} = \mu_{A_{ij}}\left(x_j; C_{ij}, \sigma_{ij}\right) = e^{\left(\frac{-\left(x_j - c_{ij}\right)^2}{2\sigma_{ij}^2}\right)} \tag{2}$$

Where x is the input, c is the centre and σ is the width, i is the input number and j is the terms number.

The output of the Fuzzification layer is the degree of membership of the input values. It can be given as:

$$O_{ij}^2 = \mu_{A_{ij}}(x_i) \tag{3}$$

The Premise layer (L3) is the second hidden layer. Each neuron in the layer corresponds to a fuzzy rule in the system. It receives signals only from relevant fuzzification neurons and calculates the activation of premises of the fuzzy rules. It uses minimum type t-norm to implement AND operators in each of the units. The t-norm operator for determining firing strength of each rule is given as:

$$O_j^3 = w_j = \prod_{j=1}^{36} \mu_{A_j}(x_i) \tag{4}$$

The fourth layer (L4) is the third hidden layer. Each neuron in this layer receives signals from all rule neurons in layer 3. It then calculate the normalized firing strength for a given rule to a close range [0 1]. It is given as:

$$O_j^4 = \overline{w}_j = O_j^3 / \sum_{j=1}^{36} O_j^3 = \prod_{j=1}^{36} \mu_{A_j}(x_i) / \sum_j^{36} \prod_{j=1}^{36} \mu_{A_j}(x_i) \tag{5}$$

The consequent layer is the defuzzification layer. Each neuron in this layer receives the initial input signals and connected to the respective normalization neuron in the preceding fourth layer. It adaptively multiplies O_j^4 with the consequent parameters as shown in (14).

$$O_j^5 = O_j^4 f_j(x) = \bar{w}_j f_j(x) = \bar{w}_j(p_j x_1 + q_j x_2 + r_j x_3 + s_j x_4 + t) \quad (6)$$

Here, the p, q, r, s, and t for each defuzzification neuron are the consequent parameters; they are to be determined by training. The $x_1, x_2, x_3,$ and x_4 are the initial inputs received.

The sixth layer is the output layer. This is a layer of single neuron that calculates the sum of the weighted consequent value from the defuzzification layer. For our model with 36 rules the overall output is given as:

$$O^6 = \sum_{j=1}^{36} \bar{w}_j f_j(x) = \sum_{j=1}^{36} \bar{w}_j \left(p_j x_1 + q_j x_2 + r_j x_3 + s_j x_4 + t\right) = y \quad (7)$$

3.2 The Network Training

The training techniques used for the network parameter identification is the Hybrid learning algorithm. It is a two pass algorithm; the forward pass and the backward pass. This option is opted for because the use of back propagation (Steepest Descent Method) - a one pass algorithm alone as shown in Table 2 converges with high error estimates. The hybrid learning uses combines Steepest Descent (SD) with Least Squares Estimation (LSE). This learning method is possible because output of the adaptive network is linear in some of the network's parameters as shown in (6) and (7). The two passes of the Hybrid learning algorithm is known as forward propagation and backward propagation respectively. In the forward pass, on the condition that the premise (nonlinear) parameters are fixed, the input vector is propagated through the network layer by layer until the last layer in which the consequent parameters are estimated by linear Least-Square Method (xxx). The sum of squared residuals or the sum of squared errors is given as:

$$\text{The Sum of Squared Error} = \sum_{i=1}^{36}(y_i - y_i^*)^2 \quad (8)$$

The y_i is the desired output and y_i^* is the observed output from the system. The essence of squaring the errors is to prevent cancellation of error value with opposite signs, thereby gives the sum of error made by the network in the course of identifying the training dataset. The network error is a function of the weight (w) of the network. If the error is unsatisfactory, the network weight needs to be adjusted in a manner to minimize the error. This can be achieved by using Gradient descent method. The Gradient descent method is 'a going downhill in small steps' method until the bottom of the error surface is reached. As a back propagation technique, the weight update of Gradient descent is scaled by a learning rate η as show in (9).

$$w_{ji\,\text{new}} = w_{ji\,\text{old}} + \eta(y_j - y_j^*)x_i \quad (9)$$

The w_{ji} represents the synaptic weight to jth neuron in the output layer from the ith neuron in the preceding layer. The parameter η is the speed at which error correction is made. Since changing the weight value could lead to convergence or divergence from the local minima of the curve. It is essential therefore to take steps proportional to the negative of the gradient at the point of estimation.

3.3 Models Implementation

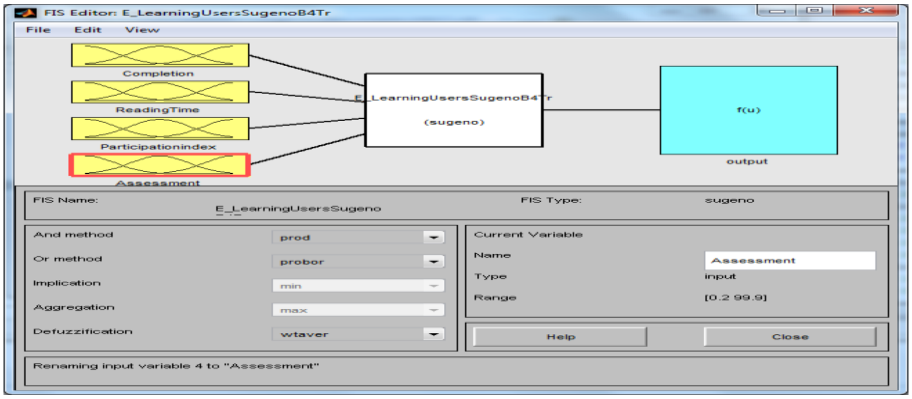
Each of the stages in the Neuro-Fuzzy Network development as discussed where simulated in the MATLAB Version 7 environment. The following are the descriptions of the implementations.

3.3.1 Sugeno Fuzzy Model Simulation

The Sugeno Fuzzy Inference System was produced by a direct transformation of the Mamdani Fuzzy in (Isiaka. Et.al, 2014). The Mamdani Fuzzy model which was developed based on expert knowledge has the same premise as the Sugeno but the consequent of the later is in the form of First Order Polynomial. The MATLAB command used for the transformation is

>> a=readfis('e_learningusers') ; >> sugenoelearningusers=mam2sug(a).

Figure 2 shows the Sugeno Model in MATLAB FIS Editor



FIS Name: E_LearningUsersSugeno
FIS Type: sugeno

And method: prod
Or method: probor
Implication: min
Aggregation: max
Defuzzification: wtaver

Current Variable
Name: Assessment
Type: input
Range: [0.2 99.9]

Renaming input variable 4 to "Assessment"

Figure 2: Formulated Sugeno FIS Model

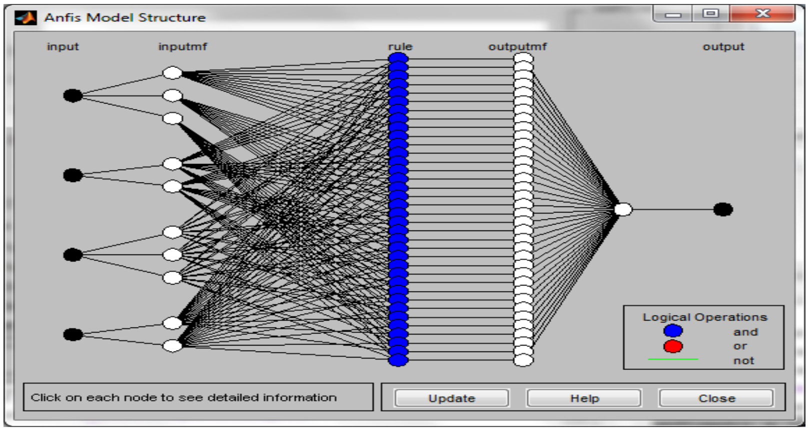
The Editor shows the four input variables, the Sugeno Inferences, the resultant linear output and selected training parameters.

3.3.2 Neuro-Fuzzy Model Simulation

The Sugeno FIS Model was converted to its Adaptive Neuro-Fuzzy Inference System (ANFIS) equivalent with the command:

```
>> anfisedit e_learninguserssugeno
```

The generated Neuro-Fuzzy Structure for the E-learning Users Model is shown in figure 3.



Click on each node to see detailed information

Figure 3: Generated Neuro-Fuzzy Structure E-learning Users Model

The system auto-generated fuzzy inference structure shown in figure 3 is the equivalent of the structure in figure 1. The figure shows the four neurons of the input layer, one for each of the input variables. The second layer shows the ten (10) fuzzification neurons, one for each of the fuzzy-terms. The third or premise layer has the thirty six (3 x 2 x 3 x 2) neurons, one for each of the rules. The blue colour shows that 'AND' is the aggregation operator used. The thirty six normalization neurons and the thirty six consequent neurons in Figure 1 are merged in Figure 3. Finally, the single neuron for the inference or output is shown. Details of the neurons are revealed in figure 1 but they are hidden in figure 3, they can be revealed as a screen tip when mouse is pointed at them.

3.4 Input Data Generation and Dataset Normalization

In the early stage of the model development as described in Isiaka (2014) simulated dataset where used for the model training, validation and testing. However, the actual target dataset for the study was generated in the Rain Semester of 2013-2014 academic sessions. It was the Learning activities of Sixty (60) B.Sc Computer Science students in the Department of Computer, Library and Information Science, Kwara State University, Malete, Nigeria that enrolled for System Analysis and Design (CSC 306) in the session. The lesson which spanned for the period of fourteen weeks entailed utilization of several learning activities and resources online as provided in Moodle LMS. Moodle by defaults logs (stores) users and details of all their activities into corresponding mdl_tables. The mdl_user table stores one record for each user in the system. In all it has sixty five (65) users, five (5) for the Admin and the Teachers, the remaining sixty (60) are students records. The approached used for estimation of the four variables are follows:

3.4.1 Time Variable Estimation

The estimation of the total time spent by the student on the platform throughout the duration is the most challenging of all. The mdl_log table logged every user's actions as far as possible. The mdl_log table for this research had the total of twenty thousand seven hundred and seventy four (20,774) records. The attributes considered in the table are userid, time, ip, course and action. The four patterns of online sessions (usage) identified are Login-logout session, Login-login session, Login_ip - logout_ip session and Login_in - logout_end session.

a. Login-logout session: here the user successfully logged out after series of actions. This is the ideal and expected situations. The time spent in such session is given as Tlogout – Tlogin. The sum of all such time is given as:

$$\sum_{i=1}^{p}(Tlogout - Tlogin) \tag{10}$$

b. Login-login session: here the user could not log out of the session as expected. This situation may be as result of Internet connectivity failure or care-free attitude. In that case, the time of the last valid action that precedes the second login following the first is considered as the logout time for that session. The time spent in such session is given as (Tlogin2-2) – Tlogin1. The sum of all such time is given as:

$$\sum_{j=1}^{q}((\mathrm{Tlogin2} - 2) - \mathrm{Tlogin1}) \tag{11}$$

c. Login_ip - logout_ip session: this is the kind of situation in which a login or a logout is indicated by a change in the internet protocol (ip) addresses. Hence, the end of a previous session (which is also the beginning of a new session) is signified by the difference in the successive ip addresses. The time spent in such session is given as Tlogout_ip – Tlogin_ip. The sum of all such time is given as:

$$\sum_{k=1}^{r}((\mathrm{Tlogout_{ip}} - Tlogin) + (\mathrm{Tlogout_{ip}} - \mathrm{Tlogin_{ip}})) \qquad (12)$$

d. Login_in - logout_end session: this category of session represent a situation when the last action for a student's record set in the mdl_log is not a logout. The time spent in such session is given as Tlogout_end - Tlogin. The sum of all such time is given as:

$$\sum_{l=1}^{s}(\mathrm{Tlogout}_{\mathrm{\ end}} - Tlogin) \qquad (13)$$

The total time spent by the student in relation to the threshold is

$$\mathrm{x_t} = \frac{\sum \mathrm{te}}{\mathrm{T}} \times 100 \qquad (14)$$

where $\sum \mathrm{te}$ is the total time spent by the learner on the lesson and T is the total threshold for the time in seconds. Equation for the total time spent by the student is:

$$\sum \mathrm{te} = \sum_{i=1}^{p}(Tlogout - \ Tlogin) + \sum_{j=1}^{q}((\mathrm{Tlogin2} - 2) - \mathrm{Tlogin1}) + \sum_{k=1}^{r}\left((\mathrm{Tlogout_{ip}} - Tlogin) + (\mathrm{Tlogout_{ip}} - \mathrm{Tlogin_{ip}})\right) + \ \sum_{l=1}^{s}(Tlogout_{end} - Tlogin) \qquad (15)$$

Where p,q,r and s are the upper bounds for their respective type of sessions. The i,j,k and l are the counters for the various sessions.

3.4.2 Completion Variable Estimation

The table that stores the completion state (completed or non_ completed) of all modules specified by the teacher against all registered students is the mdl_course_modules_completion table. The attributes considered in this table are *userid, coursemoduleid* and *completionstate*. The total instance of module completion status in this lesson was four hundred and sixty eight (468) out of this number, four hundred and forty nine (449) while the status of nineteen (19) of them are non_completed. Completed status had status (1) while the status (0) was assigned to non completed status.

The completion value for a student is estimated based on the equation

$$\mathrm{x_c} = \frac{\sum \mathrm{Ce}}{\mathrm{C}} \times 100 \qquad (16)$$

where $\sum \mathrm{Ce}$ is the total number of modules completion records with 'completed status' that is available for the student and the C is the completion threshold set by the teacher.

3.4.3 Participation Index Variable Estimation

The measure for student participation in a lesson is a composite function. In this research, participation is treated as a function of assignment submission, approved project proposal, survey participation, message initiation and reply to message, and forum post.

The mdl_assign_submission is the log table for monitoring student assignment submission. The attributes considered include *userid, assignment_id and the assignment_statusfields*. The assignment_statusfields could either be submitted (1) or draft (0). The total number of submissions in this table was two hundred and fifty seven (257), the number with submitted status was two hundred and fifty six while the status of only one record reads draft.

The mdl_data_records table was used for database. It was used in this work to process the projects topics submitted by the students. The attributes used include *userid, dataid, and approval*. A submission that was approved had its status set to 1 otherwise it was set to 0. The table had fifty seven (57) submissions, fifty three (53) were approved while four (4) were not approved. The total number of students that participated in the survey as shown in the mdl_feedback_tracking table was forty five (45).

Furthermore, the mdl_message_read table was used to track student to students' interactions. The table had three hundred and thirty records (330), two hundred and eighty one (281) of them had notification status 1the remaining forty nine (49) had notification status 0. Those with notification status 1 were the system notifications message they were not included in the estimation of participation index. Only those with status 0 were included in the estimation because they were the messages exchanged between students.

Finally, students' forum posts were included in the participation index. The mdl_forum_post table logs the record of students' forum threads either as parent or as children. The total number of forum threads was thirty four (34). The students that participated in messaging and forum had the points included in their grading. If u,v,w,x,y, is used respectively for tables mdl_assign_submission, mdl_data_records,mdl_feedback_tracking, mdl_message_read and mdl_forum_post. The total participation index for a student x can be estimated using

$$\mathrm{x_p} = \frac{\sum \mathrm{pe}}{\mathrm{P}} \times 100. \tag{17}$$

Where the $\sum \mathrm{pe} = \sum \mathrm{u, v, w, x, y}$ and P is the threshold set by the teacher for Participation Index.

3.4.4 Diagnosis Assessment Estimation

The students Diagnosis Assessment is extracted from the mdl_quiz_grades. The table has the quiz value for all students that participated in the quiz. The overall quiz grade can be estimated using

$$\mathrm{x_d} = \frac{\sum \mathrm{q}}{\mathrm{Q}} \times 100 \tag{18}$$

Where q is the number of correct diagnostic questions, Q is the number of diagnostic questions available.

4 Performance Evaluation Metrics

The efficiency of the model was determined using both classification uncertainty metrics and confusion matrix–based metrics. The classification uncertainty metrics considered are Mean Absolute Error (MAE) and Root Mean-Squared Error (RMSE). The confusion matrix-based metrics used are Accuracy, Precision and Recall. The 70% of the dataset were used for training the models, 15% were used for the model validation and remaining 15% were used for the testing. Details of the evaluation procedures are as follows.

4.1 Evaluation of the Sugeno Fuzzy Models Based On Aggregation and Defuzzification Techniques

The Sugeno Fuzzy Model described in 3.3.1 was evaluated using the Aggregation and Defuzzification Techniques. The two Aggregation techniques (Prob_ Probor and Max_Min) where combined with the Wtsum and Wtaver Defuzzification Techniques. The result is as shown in table 1.

Table 1: Sugeno Fuzzy Model Evaluation Based On Aggregation and Defuzzification Techniques

Aggregation Techniques	Prob_Probor		Max_Min	
Defuzzification Techniques	Wtsum	Wtaver	Wtsum	Wtaver
(MAE)	2190.5	3489.4	1342.8	13936.4
(RMSE)	46.9	59.1	36.7	118.1

Table 1 shows the performance of four types of Sugeno Fuzzy models. In all, the models with Wtsum deffuzification are the best. They have the least MAE and RMSE values when compared with those of Wtaver deffuzification. Attempt was made to convert the best of the models (MAE: 1342.8 and RMSE: 36.7) with Max-Min Aggregation and Wtsum deffuzification to Adaptive Neuro-Fuzzy Inference System (ANFIS) model. However, it was realized that ANFIS support only Sugeno with Wtaver deffuzification. To this end, the Sugeno model with the least value (MAE: 3489.4 and RMSE: 59.1) and Prob_Probor Aggregation Technique made the choice for building the ANFIS models whose performances are further shown in Table 2.

4.2 Evaluation of ANFIS Based on Training Methods

The effect of training and the choice of training methods were determined by taken two copies of the untrained ANFIS model which was the direct transformation of the Sugeno. The first copy was trained using the Back propagation method while the second copy was trained by the Hybrid (Least Square Estimation and Gredient Descent) method. The result is as shown in Table Tb.

Table 2: Effects of Training Methods on ANFIS

ANFIS (Prob_Probor/Wtaver)			
Metrics	Before Training	After Back Propagation Training	After Hybrid Training
(MAE)	2190.5	2074.8	127.4
(RMSE)	46.9	45.6	11.3

As shown, table 2 reveals the performance of three categories of the ANFIS models. The MAE and RMSE of the untrained ANFIS are 2190.5 and 46.9 respectively. This significant improvement over the Sugeno equivalent with MAE (3489.4) and RMSE (59.1) even before training of the model reveals the strength of the ANFIS architecture. The quest for optimized model however necessitated training model using the Back Propagation and the Hybrid Training Techniques. The performance of the Back propagation trained model is 2074.8 for the MAE and 45.6 for the RMSE. These error figures are very high when compared with those obtained when the Hybrid training methods were used. In the Hybrid model which is the best, the MAE was 127.4 and the RMSE was 11.3. This information established the fact that adaptive network trained by the Hybrid algorithm (Forward and Backward – two passes) will outperform the untrained and a one pass trained network of the same architecture. This deduction provides the bases for using the Hybrid training for the ANFIS model built on Expert knowledge and four other ANFIS models built on Historical data (Isiaka, 2014) and default membership functions. The details of the performances of these five (5) models follow.

4.3 Evaluation of ANFIS Models Based on Identification Techniques and Membership Functions

The knowledge discovered in the results of the experiment on the best Sugeno fuzzy model in table 1 and that of table 2 on the appropriate choice of training algorithm came into play in designing equivalent ANFIS models. The models were developed based on Historical data and varied four (4) default membership functions (MF). The membership functions considered are Triangular MF, Gaussian (MF), G-Bells MF and Sigmoid MF. The performances of these models were compared with the developed experts' knowledge based model. The Precision Accuracy of the five models shown in Table 3.

Table 3: Comparison of the Models by Precision Accuracy

Model Types: ANFIS Partitioning Methods: Grid Optimization Techniques: Hybrid			
Membership Functions	MAE	RMSE	Ranking
Developed Model	127.4	11.3	1
Triangle	163.2	12.8	4
Gaussian	160.4	12.7	3
G. Bells	168.0	13.0	5
Sigmoid	157.6	12.6	2

Table 3 shows the MAE and RMSE for the Expert based model and those based on Historical data. The G-Bells model display the worst performance, it was ranked 5th, because its performance estimates for the MAE and the RMSE are 168.0 and 13.0 respectively. The model with Triagular MF was ranked 4th, its MAE is 163.2 and its RMSE is 12.8. The 3rd in the ranking is the Gaussian MF based model. The table shows that the 2nd best model is the Sigmoid MF based model while best of the five models in consideration is the formulated model. The MAE and RMSE for the 2nd best are 157.6 and 12.6 respectively. The MAE and RMSE for the Expert based which is the best of the models are 127.4 and 11.3 respectively. Furthermore, in Table 4, the superiority of the developed model was further demonstrated.

Table 4: Comparison of Neuro-Fuzzy Models Complexity

Model Types: ANFIS Partitioning Methods: Grid Optimization Techniques: Hybrid						
Membership Functions	Training Time (sec)	Epoch	Training Errors	Training Data Errors	Validation Data Errors	Ranking
Developed Model	5	40	10.6538	10.6506	9.6042	1
Triangular	5	40	12.5577	12.5566	13.2108	3
Gaussian	4	40	12.7555	12.7529	13.3295	4
G-Bells	5	40	13.2678	13.2591	13.9856	5
Sigmoid	5	40	12.7382	12.5532	12.6077	2

Table 4 shows that the developed model has the least complexity figures, follow by Sigmoidal MF models. There is a swap of ranking positions between the Triangle MF model and the Gaussian MF model. In complexity the Triangle MF model outperformed the Gaussian. The G-Bells model retained the worth performance ranking of 5th position.

Finally, the Classification Accuracy of the models were determined as shown in table 5 and table 6

Table 5: Measures of Classification Accuracy of the Models

Positive (Proceed) = 67.0 Negative (Repeat) = 83.0 Total = 150.0					
Expected Values	**Develop Model**	**Triangle Model**	**Gaussian Model**	**G-Bells Model**	**Sigmoid Model**
TP	67.0	66.0	67.0	67.0	64.0
TN	74.0	65.0	66.0	68.0	67.0
FP	9.0	18.0	17.0	15.0	16.0
FN	0.0	1.0	0.0	0.0	3.0
Metrics					
Accuracy (TP +TN) / (P+N)	0.9	0.9	1.0	1.0	1.0
Precision TP / (TP+FP)	0.9	0.8	0.8	0.8	0.8
Recall TP / (TP+FN)	0.5	0.5	0.5	0.5	0.5

The analysis in table Table 5 shows that Gaussian, G-Bells and Sigmoidal membership function based model have accuracy value of 1.0 which is higher by 0.1 when compared with the 0.9 of the developed and Triangle MF models. The less accuracy value of the developed model does not discredit it supremacy, this is because accuracy has been adjudged to be a non sensitive metric especially when handling unbalanced dataset (). Secondly and most importantly is the precision value in which the developed model has 0.9 where the value for other models is 0.8. Finally, the developed model is in per with others in the Recall capability. Table 6 provides the compacted view of the classification accuracy of the models in a Confusion Matrix.

Table 6: The Confusion Matrix of Classification Models

Confusion Matrix										
	Developed		Triangle		Gaussian		G-Bells		Sigmoid	
	True Class		True Class		True Class		True Class		True Class	
Predicted Class	P(+ve)	N(-ve)	P(+ve)	N(-ve)	P(+ve)	N(-ve)	P(+ve)	N(-ve)	P(+ve)	N(-ve)
T (True)	TP (67.0)	FP (9.0)	TP (66.0)	FP (18.0)	TP (67.0)	FP (17.0)	TP (67.0)	FP (15.0)	TP (64.0)	FP (16.0)
F (False)	FN (0.0)	TN (74.0)	FN (1.0)	TN (65.0)	FN (0.0)	TN (66.0)	FN (0.0)	TN (68.0)	FN (3.0)	TN (67.0)
Total	P=TP +FN (67.0)	N=FP +TN (83.0)	P=TP +FN (67.0)	N=FP +TN (83.0)	P=TP +FN (67.0)	N=FP +TN (83.0)	P=TP +FN (67.0)	N=FP +TN (83.0)	P=TP +FN (67.0)	N=FP +TN (83.0)

The Confusion Matrix as shown in Table 6 provides another evidence for rating the developed model higher than others. As indicated, the developed model has the best True Positive value of 67.0 which is the highest, though it shares this figure with Gaussian and G.Bell models. However, its strength over these two is vivid from its 9.0 False Positive (FP) value which is very low when compared with 17.0 and 15.0 for Gaussian and G.Bell models respectively. Furthermore, the developed model did not have a single False Negative and its True Negative (74.0) is the best. The implications of this figures among others is that the system shall not in any way disadvantage any student.

5 Conclusion

This work demonstrates an approach to overcome the challenges of system identification by Trial and Error method. It recommends that a FIS of a problem should be created using whatever level of expert knowledge that may be available. Such model should then be transformed to ANFIS via its Sugeno equivalent. The transformed model can trained for better performance if historical data is available. The hybrid training technique is recommended since it has proven to be the most efficient. Moving forward, this approach shall be tested in different domain and on other Neuro Fuzzy Model.

REFERENCES

[1] Ahmed, A., Jun, S., Rami, A., and Jun, Y., *Modeling and simulation of an adaptive neuro-fuzzy inference system (ANFIS) for mobile learning*. IEEE Transactions and Learning Technologies, 2012. 5(3): p. 226 – 237.

[2] Babuska, R., and Verbruggen, H.B., *Fuzzy set methods for local modeling and identification*. In Multi model Approaches to Modeling and Control, Yaylor & Francis. 1997. P. 75 – 100.

[3] Bodyanskiy Y., and Dolotov A., *Methods and Instruments of Artificial Intelligence*. Rzeszow-Sofia, Bulgaria: ITHE, 2010. p. 17–24.

[4] Ebru, A., and Parvinder, S.S., *A soft computing approach for modeling of severity of faults in software systems.* International Journal of Physical Sciences, 2010. 5(2): p. 74-85.

[5] Hoogendoorn, R.G., Van Arem, B., and Hoogendoorn, S.P., *A neeurofuzzy apprach to modeling longitudinal driving behaviour and driving task complexity*. International Journal of Vehicular Technology. 2013. P. 1-12.

[6] Isiaka, R.M., Omidiora, E.O, Olabiyisi, S.O, and Okediran, O.O.*, Mamdani fuzzy model for learning activities evaluation.* International Journal of Applied Information Systems (IJAIS). Foundation of Computer Science FCS. 2014. New York, USA. 7(3): p. 1-8

[7] Jang, J.R., Sun , C.T., and Mizutani, E., *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. 1997. USA: Prentice-Hall,Inc.

[8] Jones, M.T., *Artificial intelligence: A systems approach*. Infinity Science Press LLC, 2008. New Delhi

[9] Lan, T.H., Lo, E.W., Wu, M.S., Hu, T.M., Chou, P., Lan, T.Y., and Chiu, H.J., *Performance of a neuro-fuzzy model in predicting weight changes of chronic schizophrenic patients exposed to antipsychotics*.Molecular Psychiatry, 2008. 13: p. 1129-1137.

[10] Nikam, S.R., Nikumbh P.J., and Kulkarni, S.P.*, Fuzzy logic and neuro-Fuzzy modelling*. Recent Trends in Computing. International Journal of Computer Applications (IJCA), 2012. 22: p. 22-31.

[11] Omidiora, E.O, Olabiyisi, S.O, Okediran, O.O., and Isiaka, R.M., *Learner activities evaluation model: A neuro-fuzzy approach*. International Journal of e-Education, e-Business, e-Management and e-Learning. 2013. 3(5): p. 421-424.

[12] Sevarac, Z., *Neuro fuzzy reasoner for student modeling*. Advanced Learning Technologies, 2006. IEEE Sixth International Conference. P. 740-744.

[13] Sun, Z., and Finnie, G.*, Experience management in knowledge management*. Information Technology Papers. 2005. Paper 103. Retrieved on 23/04/2010 from http://epublications.bond.edu.au/ infotech_pubs/103.

[14] Yuanyuam, C., Limin, J., and Zundong, Z., *Mamdani model based adaptive neuro fuzzy inference system and its application.* World Academy of Science, Engineering and Technology. 2009. 3: p. 743- 750.

[15] Zadeh, L.A., *Outline of a new approach to the analysis of complex systems and decision processes*. IEEE Trans. on systems, Man and Cybernetics, 1999. 3 (1): p. 28-44.