NEURO-FUZZY MODELS FOR CONSTRUCTABILITY ANALYSIS

SUBMITTED: February 2004
REVISED: April 2004
PUBLISHED: May 2004 at http://www.itcon.org/2004/4/
EDITOR: B.-C. Björk

S V Barai, Assistant Professor
Department of Civil Engineering, Indian Institute of Technology, India
email: skbarai@civil.iitkgp.ernet.in http://barai.sudhir.tripod.com

Rajeev S Nair, Former post graduate scholar
Department of Civil Engineering, Indian Institute of Technology, India

SUMMARY: *With the emergence of the new computer science areas of artificial intelligence and neural networks, researchers have applied them in the construction industry successfully. This paper presents comparative studies of two machine learning models namely backpropagation (BP) and Fuzzy ARTMAP based neuro-fuzzy models for handling qualitative fuzzy information of constructability evaluation. These models not only perform like traditional machine algorithms, but also handle missing information with better accuracy. Performance evaluation of the network has been carried out using traditional statistical tests. From the study, it was found that the Fuzzy ARTMAP model performs much better than the BP model.*

KEYWORDS: *constructability, fuzzy logic, neural networks*

1. INTRODUCTION

Construction as a sub-discipline of civil engineering is perhaps at the forefront of testing the applicability of Decision Support Systems (DSS) technology. Tasks like design of construction methods, concrete mixing and placement, constructability analysis, project planning, scheduling and control, construction quality control, etc. are receiving attention as application domains. Among them, developing DSS for constructability analysis is a challenging task (Tatum 1987, Eldin, 1988). Constructability is the optimum use of construction knowledge and experience to achieve overall project objectives and its success. The constructability analysis will minimize change orders, claims and disputes. Maximum benefits from the constructability analysis result when constructors, engineers and managers with the requisite experience and knowledge become involved at the very beginning of the project. By analyzing the project early on, one can pinpoint areas of potential conflict or concern. These points can then be taken care during the design phase versus during the actual construction.

Computerizing constructability analysis steps is a key to effective construction process automation. However, it is impossible to develop DSS for such analysis without acquiring formal constructability knowledge for various structural systems. This constructability knowledge cannot be manually acquired because of the complexity of problems involved. Also, the domain experts find it extremely difficult, if not impossible, to articulate complicated relationship among many design decisions regarding structural system and its constructability. Conventional Machine Learning (ML) approach has been applied, but due to its inherent limitations such as, inability to learn implicit knowledge, failures in complete information handling etc., there was a need to explore new ML model for knowledge acquisition and subsequently to make it an integral part of DSS.

Chua et al (1997a, 1997b) and Kog et al. (1999) have presented interesting study on identifying key management factors that could affect budget performance in a project. They have used field data of project performance to build the budget performance model. They have developed a neuro model based on the key determining factors related to the project manager, project team and planning and control efforts. The factors were number of organizational levels between project manager and craftsmen, project manager experience on similar technical scope, detailed design complete at start of construction, constructability program, project team turnover rate, frequency of control meetings during construction, frequency of budget updates, and control system budget. They found that model could give better performance in the events of either unseen data or incomplete information on the key factors.

Yu and Skibniewski (1999a) presented a multi-criteria model for qualitative constructability analysis based on neuro-fuzzy knowledge based system. They developed multiplayer information aggregation network to incorporate the manager's subjective preference information. A systematic approach was demonstrated for constructability problem detection and constructability improvement.

Yu and Skibniewski (1999b) proposed neuro-fuzzy approach for constructability knowledge acquisition for construction technology evaluation. Fuzzy logic based knowledge representation space is developed for neuro-fuzzy models. They demonstrated learning ability of neuro-fuzzy model combining genetic algorithms (GA) for automatic acquisition of constructability knowledge from training examples. Their proposed approach provided a mechanism to track back factors causing unsatisfactory construction performance and feedback to construction engineers for technology innovation.

Recently, Ugwu et al. (2004) presented cognitive model study on the acquisition of knowledge, elicitation of problems that are associated with managing constructability design knowledge, and understanding organizational constructability planning and problem solving methods in the steel structures domain. The selection of an example of portal frame was intended to understand construction domain but they stated that the concepts could be extended to other infrastructure projects.

The main objectives of the paper are to explore different neuro-fuzzy models for constructability analysis, demonstrate their feasibility for the domain example, and demonstrate the performance of neuro-fuzzy model in the event of missing information. In the present work, constructability of a beam in reinforced concrete frames is investigated. For this problem, constructability data is acquired from the existing literature (Skibniewski et al., 1997). The backpropagation (Haykin, 2000) and the Fuzzy ARTMAP (Carpenter et al. 1992) based neuro-fuzzy models are explored. These models used the collected data during learning and testing. The reliability of the models is also checked using existing statistical evaluation methods. The incomplete information handling by Neuro-fuzzy model is studied. The observations of this systematic study is used to propose future development of a larger scale DSS. Next section discusses about problem definition and data collection of constructability evaluation.

2. PROBLEM DEFINITION AND CONSTRUCTABILITY DATA COLLECTION

There is a potential in developing DSS for constructability evaluation. Some important issues include analysis of the constructability of designs, choice of construction material, selection of the best design-function-cost combination, choice between prefabricated and in-situ construction, and feedback into the design process. In developing DSS, an important issue of knowledge acquisition and implementation comes into picture. Artificial Neural Networks (ANN) can be an excellent learning model for DSS. For exploring the possibility of ANN application to constructability analysis, the objectives of the current study can be divided into four major issues. They are as follows and will be discussed in Sections 3-5:

- Qualitative constructability analysis data modeling using Neuro-fuzzy models
- Comparison of Neuro-fuzzy Model with ML generated rules
- Performance reliability of the Neuro-fuzzy model
- Missing data handling in Neuro-fuzzy model for constructability analysis

Data collection: The relevant data for the current study is obtained from literature (Skibniewski et al., 1997). In this collection of 31 data examples, each example represents a structural design concept evaluated from the point of view of its constructability. This data will be used to train the ANN models. The typical data set (Table 1) describes the constructability of a beam in a 12-storied building. There are seven independent attributes describing the beam design. The dependent attribute is constructability evaluation (ConEva). It is a measure of the constructability of a given design and has been attributed three values: *Poor, Good and Excellent.*

TABLE 1: Typical Constructability data for the beam design problem (Skibniewski et al., 1997)

No	ReRa	CoBeRa1	CoBeRa2	NoSla	NoWall	BeCha1	BeCha2	ConEva
1	High	High	High	DiffTwo	SameTwo	WDchange	AllChgeReinf	Poor
2	Average	High	Average	DiffTwo	One	SliChgereinf	WDchange	Poor
3	Average	Average	High	DiffTwo	DiffTwo	AllChgeReinf	None	Poor
4	High	Average	High	SameTwo	One	AllChange	SliChgereinf	Poor
5	Average	High	Average	SameTwo	DiffTwo	AllChgeReinf	AllChange	Poor

The independent attributes are described below;
ReRa represents reinforcement ratio of the beam, and three values are attributed: *Low, Average and High.*
CoBeRa1 represents the first beam to column connection with the values attributed as: Low*, Average and High*.
CoBeRa2 represents the second beam to column connection with values attributed as: Low*, Average and High.*

NoSla represents the number of slabs attached to the beam. The attributed values could be *None, One, SameTwo and DiffTwo*. Value *One* means one slab is attached to the beam. Value *None* means no slab is attached to the beam. Value *SameTwo* means two identical slabs are attached to the beam and value *DiffTwo* means that the two slabs attached are not identical.

NoWall represents the number of walls attached to the beam. The attributed values could be *None, One, SameTwo* and *DiffTwo*. Value *One* means one wall is attached to the beam. Value *None* means no wall is attached to the beam. Value *SameTwo* means two identical walls are attached to the beam and value *DiffTwo* means that the two walls attached are not identical.

BeCha1 represents the changes in steel reinforcements and size of the beam on the left or first side of the considered beam. Five values are used: *None, SliChgereinf, AllChgeReinf, WDchange, AllChange*. Value *None* means that the beam on the left side of the column has exactly the same shapes and size as that of the beam on the right side. Value *SliChgereinf* means that the beam on the left side has same size but slightly different reinforcement as the beam on the right hand side. Value *AllChgeReinf* means that the beam on the left side has same size but different reinforcement as the beam on the right hand side. Value *WDchange* means that the width or depth of the beam on the left side has changed but the reinforcement has not. Value *AllChange* means that the two beams are entirely different in size and reinforcement.

BeCha2 represents the changes to the beam on the right side as in the previous attribute regarding the beam on the left.

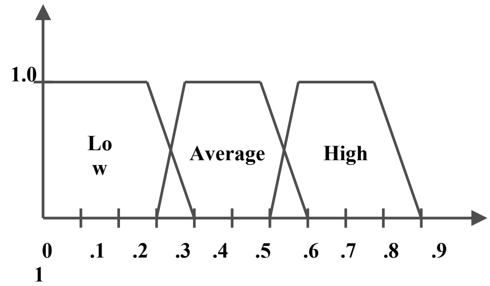
Skibniewski et al. (1997) generated rules using machine learning approach to incorporate in DSS. The examples given by Skibnewski et al. (1997) were analysed using the learning system INLEN based on AQ15 algorithm. The algorithm was developed to learn classification rules from a collection of examples. In this collection, each example is described by a number of independent attributes and their values and a single independent attribute and one of its values. Produced decision rules are the relationships between various groups of independent attributes and their dependent attribute. These rules can be used to classify unknown examples to one of the categories of the decision attribute. Here two types of learning modes were used, the specialisation and the generalisation modes. In the generalisation mode, the learning system induces rules as general as possible, i.e., they involve the minimum number of attributes, each with the maximum number of attributed values. In the specialisation mode, the learning system generates rules as specific as possible, i.e., with the maximum number of attributes and minimum number of attribute values. Next section discusses about Neuro-fuzzy Modeling.

3. NEURO-FUZZY MACHINE LEARNING MODELS

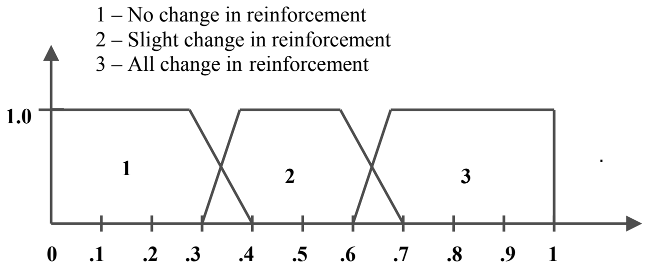
While fuzzy logic performs an inference mechanism under cognitive uncertainty (Zadeh, 1988), computational neural networks offer exciting advantages, such as learning, adaptation, fault-tolerance, parallelism and generalization (Wasserman, 1989). To enable a system to deal with cognitive uncertainties in a manner more like humans, one may incorporate the concept of fuzzy logic into neural networks. The resulting *hybrid system* is called fuzzy neural, neural fuzzy, Neuro-fuzzy or fuzzy-neuro network. Neural networks are used to *tune* membership functions of fuzzy systems that are employed as decision-making systems for constructability evaluation. Although fuzzy logic can encode expert knowledge directly using rules with linguistic labels, it usually takes a lot of time to design and tune the membership functions, which quantitatively define these linguistic labels. Neural network learning techniques can automate this process and substantially reduce development time and cost while improving performance (Kosko, 1996). Hence, in hybrid form they can provide a perfect platform to take into account changing knowledge. In theory, neural networks and fuzzy systems are equivalent in that they are convertible, yet in practice each has its own advantages and disadvantages. For neural networks, the backpropagation algorithm automatically acquires knowledge. Fuzzy systems are more favorable in that their behavior can be explained based on fuzzy rules and thus their performance can be adjusted by tuning the rules. But since, in general, knowledge acquisition is difficult and also the universe of discourse of each input variable needs to be divided into several intervals, applications of fuzzy systems are restricted to the fields where expert knowledge is available and the number of input variables is small.

3.1 Data Modeling

Fuzzy Information Presentation: The qualitative data set of constructability evaluation discussed in Section 2 has to be modeled into quantitative values in order to use in ANN models. The independent and dependent attributes were modeled as trapezoidal fuzzy functions and also discrete binary values. Various previous works has shown that trapezoidal fuzzy functions can closely imitate the modeled parameters. The modeling of dependent and independent attributes (Table 2) for the typical data is given in Fig. 1 and Fig. 2 (Nair, 2000).



(a) Fuzzy represenantion of ReRa, CoBeRe1 and CoBeRe2 variable



No width depth change	Width depth change
0 0	1 1

(b) Fuzzy/binary representation of BeCha1 and BeCha2 variables

None	One	Same	DiffTwo
0 0	0 1	1 0	1 1

(c) Binary representation of NoSla and NoWall variables

FIG. 1: Fuzzy/Binary representation of input variables of neuro-fuzzy models

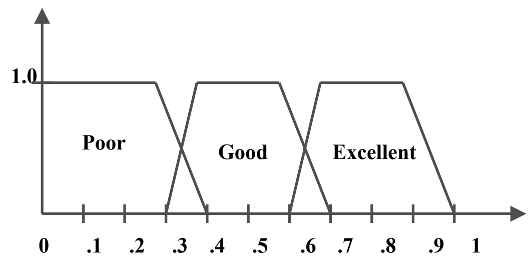


FIG. 2: Fuzzy representation of output variable of neuro-fuzzy models

TABLE 2: Identified variables for Neuro-fuzzy model input and output parameters

Variable Name	Description	Fuzzy Variables
ReRa	Reinforcement Ratio	Low, Average, High
CoBeRe1	First Beam to Column Connection	Low, Average, High
CoBeRe2	Second Beam to Column Connection	Low, Average, High
BeCha1	Change in Steel Reinforcements and Size of the Beam on the Left or First Side of the Beam	None, SliChgeReinf, AllChgeReinf, WDchange, AllChange
BeCha2	Change in Steel Reinforcement and Size of the Beam on the Right or Second Side of the Beam	None, SliChgeReinf, AllChgeReinf, WDchange, AllChange
NoSla	Number of Slabs Attached to Beam	None, One, SameTwo, DiffTwo
NoWall	Number of Walls Attached to Beam	None, One, SameTwo, DiffTwo
ConEva	Constructability Evaluation	Poor, Good, Excellent

Incomplete/Missing Data Modeling: Skibniewski et al. (1997) generated constructability decision rules using machine learning approach. Now to implement these rules for ANN model one has to take care of the combination of two or more values (e.g. Average or Poor) and empty field(s) of the Tables given by Skibniewski et al. (1997). The data in which the ‘or’ estimations were given was modeled by taking the average of the attributed values. The empty fields in Tables of Skibniewski et al. (1997) will be treated as missing data for ANN modeling purpose. Missing data is considered as the average of all the attribute values associated with that attribute data used for training the network. Missing data values are modeled as trapezoidal fuzzy functions and the final representation of the missing values are given in Nair (2000).

3.2 Neuro-fuzzy Modeling

Different types of neural networks models are available in literature (Haykin, 2000). A study of two of them is carried out in the present investigation.

- The backpropagation based multilayer perceptron was selected owing to its recognized ability to perform regression and classification (Haykin, 2000). The neural network architecture had two hidden layers with 48 hidden units (in each layer) having sigmoid activation function in each layer. The program was implemented using improved backpropagation in C programming Language in Turbo C Environment. After several exercises, keeping compromise between accuracy and computational time, sum square error was selected as 0.0001, learning rate as 0.7 and momentum rate as 0.9. Note that no optimization of the architecture or training parameters was performed. In contrast, common architecture and parameters were selected so that the time consuming exercise will take reasonable time (Nair, 2000).
- Grossberg established a new principle of self-organization known as adaptive resonance theory - ART (Carpenter et al., 1992). Basically, the theory involves a bottom-up recognition layer and a top-down generative layer. When the input pattern and learned feedback pattern match, a dynamic state generated is called 'adaptive resonance'. The fuzzy ARTMAP is a synthesis of fuzzy logic and ART. Various supervised ART algorithms are named with the suffix MAP. These algorithms cluster both input and outputs, and associate the two sets of cluster. For the present study, we have implemented the Fuzzy ART algorithm (Carpenter et al., 1992) in C programming Language in Turbo C Environment (Fig. 3). The input vector “a” is applied to ART_a and its correct prediction “b” is applied to ART_b module. Now when the prediction of ART_a is disconfirmed at ART_b, inhibition of mapfield activation induces a match tracking process. Match tracking raises the ART_a

vigilance parameter ρ_a to just above $F1_a/F0_a$ match ratio. This triggers to a ART_a search which leads to the activation of a ART_a category that correctly predicts "b". For this study, choice parameter (α) and Learning rate parameter (β) were 0.001 and 1 respectively. The vigilance parameters considered for the study are given in Table 3.

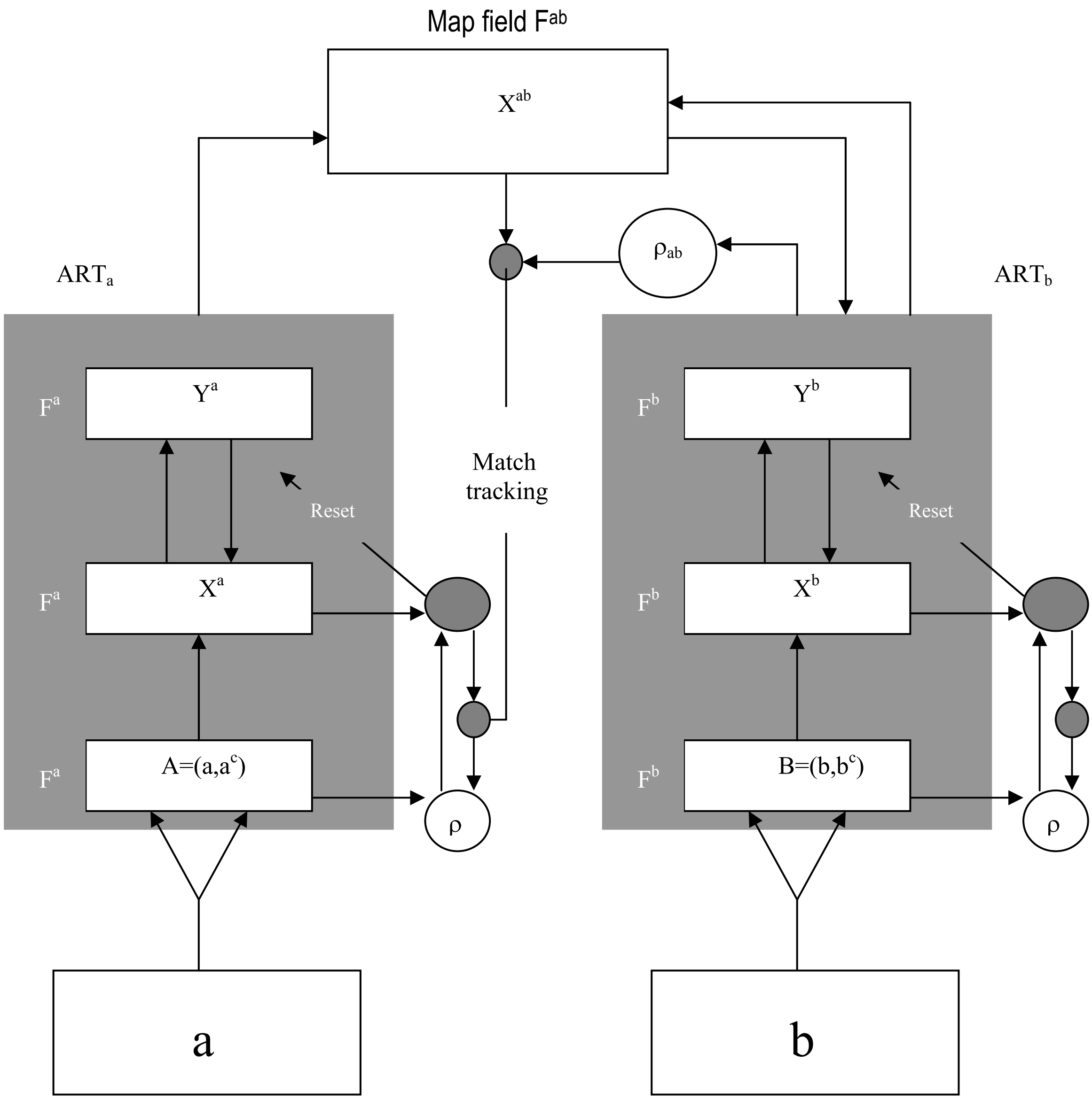


FIG. 3: Fuzzy ARTMAP Architecture

TABLE 3: Fuzzy ARTMAP Vigilance parameter.

ART_a vigilance parameter (ρ_a)	0.3	0.35	0.4	0.7	0.75
ART_b vigilance parameter (ρ_b)	0.2	0.25	0.3	0.6	0.65

If the vigilance parameter is too high, most examples will fail to match those in storage and network will create a new neuron for each one of them. This will lead to poor generalization, as minor variations of the same example become separate categories. Conversely, if the vigilance parameter is too low, totally different decisions will be grouped together, distorting the stored example until it bears little resemblance to any of them.

4. MODELS EVALUATION

The use of ANN in engineering applications has increased dramatically over the last few years. However, by and large, the development of such application or their report lacks proper evaluation. The evaluation methods used to estimate the performance of ML models in general as well as ANN models in particular are: Resubstitution (R), Hold out (H), cross-validation such as leave–one–out (L) or 10-fold cross - validation (K). There is a growing interest in the ML community in understanding the properties of these tests. A general discussion on evaluation that includes ML models for understanding can be found elsewhere (Reich and Barai, 1999). Using these evaluation methods, Neuro-fuzzy models are evaluated and the results obtained during these exercises are discussed in next section.

5. RESULTS AND DISCUSSION

The various model evaluation techniques discussed in previous section were used to check the performance of models. Here the results are discussed in terms of *degree of response*. *Degree of response* is defined as the ratio of examples correctly predicted to number of examples tested.

5.1 Performance evaluation of BP based Neuro-fuzzy model

Fig. 4 shows performance of Neuro-fuzzy model for constructability analysis.

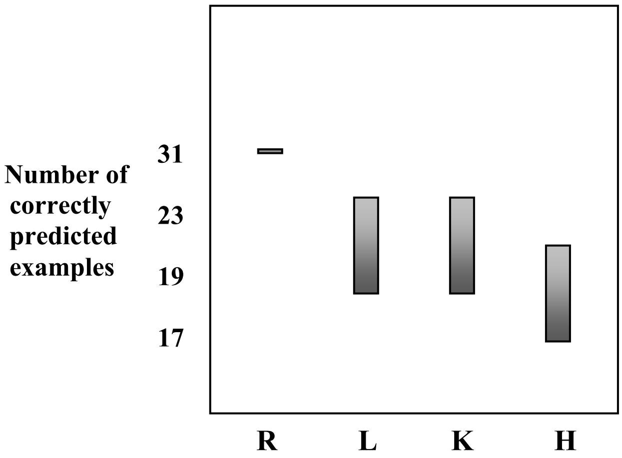


FIG. 4: Backpropagation neuro-fuzzy model evaluation

Resubstitution (R): The resubstitution exercise gave a degree of response equal to 1. All the examples were mapped perfectly as expected. But no model can be recommended as the reliable model by doing the resubstitution exercise as discussed by Reich and Barai (1999); it is a highly optimistic method.

Cross Validation - Leave-one-out (L) & K-fold (K): Cross-validation exercises were carried out 10 times on randomly generated data. These exercises gave a mean degree of response equal to .68. The deviation of results was from .74 to .61 as shown in Fig. 4. The uniformity of the results of these two exercises indicates that the performance of the model is reliable.

Holdout (H): Holdout exercise was carried out 10 times, over randomly generated data. This exercise gave a mean degree of response equal to 0.64. The results deviated from .59 to .69. This exercise gave a low degree of response, as this is a highly pessimistic approach to check the performance of the model.

Efficiency of the model: The efficiency of a model can be defined as the degree of response to missing data. The ML approach generated 22 rules (Skibniewski et al., 1997). After modeling these data, according to Section 3, the performance of backpropagation model was found to be satisfactory. Out of the 22 (10+12) tested examples, 18 examples mapped correctly. The degree of response is equal to 0.82, i.e., 82% of the total data set had been correctly mapped. This of course is a high degree of response of the model, but considering the fact that these 22 examples are generalized version of the initial 31 examples, a result of 100% mapping could not be achieved due to approximation in the qualitative data handling.

5.2 Performance evaluation of the Fuzzy ARTMAP based Neuro-fuzzy model

As discussed in previous section, the Fuzzy ARTMAP model performances were checked. The Fuzyy ARTMAP for resubstitution evaluation considering vigilance parameter (Table 3) gave degree of response 1. Further using the same dataset, the L, K, and H exercises were carried out for various vigilance parameters given in Table 3. It was found that the Fuzzy ARTMAP gave a degree of response of 1. From this exercise, it was found that the Fuzzy ARTMAP could run in a very short time and correctly.

Efficiency of the Model: The efficiency of the model against the missing data was the prime issue for the Fuzzy ARTMAP. It was interesting to observe from Table 4 that the Fuzzy ARTMAP could predict all cases correctly when the vigilance parameters, ρ_a and ρ_b were 0.3 and 0.2 respectively. This observation is very much useful from the point of view of missing data handling.

TABLE 4: Performance of the Fuzzy ARTMAP for missing input data

Vigilance parameter (ρ_a)/ (ρ_b)	0.3/0.2	0.35/0.25	0.4/0.3	0.7/0.6	0.75/0.65
Degree of response	1	0.77	0.82	0.82	0.77

6. DISCUSSION

- The backpropagation (BP) and the Fuzzy ARTMAP based Neuro-fuzzy models were used in ANN modeling for constructability analysis. It was found that backpropagation could be trained and tested for the data set given in literature (Skibniewski et al., 1997). In addition, Neuro-fuzzy model could perform well against hybrid data set (Fuzzy and Binary) of constructability analysis. The Fuzzy ARTMAP produced same performance characteristic.
- In general, Fuzzy ARTMAP model performed much better than BP based model, as it was observed that Fuzzy ARTMAP gave degree of response 1 in all situations when Vigilance parameters (ρ_a) and (ρ_b) were 0.3 and 0.2 respectively.
- During evaluation stage cross-validation (K-fold and Leave-one-out) gave the probable distribution error rates based on data characteristics. This helped in getting a picture of error rate variability for constructability analysis data set. Further, it is an indicator of model robustness under changing knowledge.
- Overall, it was found that, proper care at different stages of data modeling and ANN parameter selection gave reliable learning.
- Even though reliable neuro-fuzzy models were obtained, the network architecture, or the learning parameters were not optimal. This observed characteristic is attributed to the quality of constructability analysis data
- Neuro-fuzzy models indeed captures the rules generated by ML model. Hence such models can be replaced in DSS.
- Validation of Neuro-fuzzy models for larger size of constructability evaluation of data is still unexplored and further studies need to be carried out.

7. CONCLUSION

Applications of BP and Fuzzy ARTMAP based neuro-fuzzy models for constructability analysis has been demonstrated in the present study. The paper addressed the issues related to qualitative constructability analysis data modeling using neuro-fuzzy models; comparison of results of neuro-fuzzy models with ML generated rules, performance study of the neuro-fuzzy models and also missing data handling in neuro-fuzzy models for problem domain. The Fuzzy ARTMAP based neuro-fuzzy model showed better performance in comparison to BP based model. The study showed that there is a scope of Fuzzy ARTMAP based model and there is a need to check the validity of such model for larger size of data sets.

8. REFERENCES

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H. and Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, IEEE Transaction on Neural Networks, 3(5), 698-713.

Chua, D. K. H., Loh, P. K., Kog, Y. C., and Jaselskis, E. J. (1997a). Neural networks for construction project success, Expert Systems with Applications, 13(4), 317-328.

Chua, D. K. H., Kog, Y. C., Loh, P. K., and Jaselskis, E. J. (1997b). Model for construction budget performance – Neural networks Approach, J. Constr. Engrg. and Mgmt. ASCE, 123(3), 214-222.

Kog, Y. C., Chua, D. K. H., Loh, P. K., and Jaselskis, E. J. (1999). Key determinants for construction schedule performance, International Journal of Project Management, 17(6), 351-359.

Eldin, N.N. (1988). Constructability improvement of project design, J. Constr. Engrg. and Mgmt. ASCE, 114(4), 631-640.

Haykin, S. (2000). Neural networks - A comprehensive foundation, Pearson Education Asia, Second edition, India

Kosko, B. (1996). Neural networks and fuzzy systems, Prentice-Hall International, Englewood Cliffs, N.J., U.S.A., 3 rd edition.

Nair, R. S. (2000). Neuro-fuzzy Learning Models for Constructability Analysis, M.Tech dissertation submitted to Department of Civil Engineering, IIT, Kharagpur, India.

Reich, Y., and Barai, S. V. (1999). Evaluating machine learning models for engineering problems, AI in Engrg., 13(3), 1257 - 272.

Skibniewski, M., Arciszewski, T., Lueprasert, K. (1997). Constructability analysis: Machine learning approach, J. of Computing in civil engrg ASCE, 10(3), 6-16.

Tatum, C. B. (1987). Improving constructability during conceptual planning, J. Constr. Engrg. And Mgmt., ASCE, 113(2), 191-207.

Ugwu, O. O. Anumba, C. J. and Thorpe, A. (2004). The development of cognitive models for constructability assessment in steel frame structures, Advances in Engineering Software, (in press).

Wasserman, P.D. (1989). Neural Computing, Van nostrand Reinhold, New York.

Yu, W. and Skibniewski, M. J. (1999a). Quantitative constructability analysis with a neuro-fuzzy knowledge-based multi-criterion decision support system, Automation in Construction, 8, 553-565.

Yu, W. and Skibniewski, M. J. (1999b). A neuro-fuzzy computational approach to constructability knowledge acquisition for construction technology evaluation, Automation in Construction, 8, 539-552.

Zadeh, L. A. (1988). Fuzzy logic, IEEE Computer.,1988, April, 83-92