

A Comparative Analysis of Fuzzy, Neuro-Fuzzy and Fuzzy-GA Based Approaches for Software Reusability Evaluation

Parvinder Singh Sandhu, Dalwinder Singh Salaria, and Hardeep Singh

Abstract—Software Reusability is primary attribute of software quality. There are metrics for identifying the quality of reusable components but the function that makes use of these metrics to find reusability of software components is still not clear. These metrics if identified in the design phase or even in the coding phase can help us to reduce the rework by improving quality of reuse of the component and hence improve the productivity due to probabilistic increase in the reuse level. In this paper, we have devised the framework of metrics that uses McCabe's Cyclometric Complexity Measure for Complexity measurement, Regularity Metric, Halstead Software Science Indicator for Volume indication, Reuse Frequency metric and Coupling Metric values of the software component as input attributes and calculated reusability of the software component. Here, comparative analysis of the fuzzy, Neuro-fuzzy and Fuzzy-GA approaches is performed to evaluate the reusability of software components and Fuzzy-GA results outperform the other used approaches. The developed reusability model has produced high precision results as expected by the human experts.

Keywords—Software Reusability, Software Metrics, Neural Networks, Genetic Algorithm, Fuzzy Logic.

I. INTRODUCTION

THE demand for new software applications is currently increasing at the exponential rate, as is the cost to develop them. The numbers of qualified and experienced professionals required for this extra work are not increasing commensurably [1]. Software professionals have recognized reuse as a powerful means of potentially overcoming the above said software crisis [2], [3] and it promises significant improvements in software productivity and quality [4], [5].

There are two approaches for reuse of code: develop the reusable code from scratch or identify and extract the reusable code from already developed code. The organization that has experience in developing software, but not yet used the software reuse concept, there exists extra cost to develop the

reusable components from scratch to build and strengthen their reusable software reservoir [4]. The cost of developing the software from scratch can be saved by identifying and extracting the reusable components from already developed and existing software systems or legacy systems [6]. But the issue of how to identify reusable components from existing systems has remained relatively unexplored. In both the cases, whether we are developing software from scratch or reusing code from already developed projects, there is a need of evaluating the quality of the potentially reusable piece of software. The contribution of metrics to the overall objective of the software quality is understood and recognized [7]-[9]. But how these metrics collectively determine reusability of a software component is still at its naïve stage. A neural Network approach could serve as an economical, automatic tool to generate reusability ranking of software [10]. But, when one designs with Neural Networks alone, the network is a black box that needs to be defined, which is a highly compute-intensive process. One must develop a good sense, after extensive experimentation and practice, of the complexity of the network and the learning algorithm to be used. Fuzzy systems, on the other hand, require a thorough understanding of the fuzzy variables and membership functions, of the input-output relationships, as well as the good judgment to select the fuzzy rules that contribute the most to the solution of the application. As for the Fuzzy inference system there is a need of membership rules for fuzzy categories. It is difficult to deduce these membership rules with a given set of complex data. Neural nets and fuzzy systems, although very different, have close relationship: they work with impression in a space that is not defined by crisp, deterministic boundaries [11]. Neural network can be used to define fuzzy rules for the fuzzy inference system. A neural network is good at discovering relationships and pattern in the data, so neural network can be used to preprocess data in the fuzzy system. Furthermore, neural network that can learn new relationships with new input data can be used to refine fuzzy rules to create fuzzy adaptive system. Initially treated with skepticism, the flexibility and power of fuzzy systems is now well recognized. One major application of fuzzy systems has been in controlling manufacturing processes and various appliances such as air conditioners and video cameras. Increasingly fuzzy logic is being combined with other intelligent system methodologies to develop hybrid fuzzy-expert, neuro-fuzzy, or fuzzy-GA systems. In this paper, we

Parvinder S. Sandhu is Professor with Computer Science & Engineering Department, Rayat & Bahra Institute of Engineering & Bio-Technology, Sahauran, Distt. Mohali, Punjab-140104, India (phone: +91-98555-32004; e-mail: parvinder.sandhu@gmail.com).

Dalwinder Singh Salaria is Lecturer with Computer Science & Engineering Department, Lovly Institute of Engineering & Technology, Jalandhar, Punjab-India.

Hardeep Singh is Professor with Computer Science & Engineering Department, Guru Nanak Dev University, Amritsar(Punjab)-India.

have devised the framework of metrics that uses McCabe's Cyclometric Complexity Measure for Complexity measurement, Regularity Metric, Halstead Software Science Indicator for Volume indication, Reuse Frequency metric and Coupling Metric values of the software component as input attributes and calculated reusability of the software component. Here, comparative analysis of the fuzzy, Neuro-fuzzy and Fuzzy-GA approaches is performed.

II. METHODOLOGY USED

Reusability evaluation System for function Based Software Components can be framed using following steps:

1. A framework of metrics is proposed for structural analysis of function-based components [12]. The set of metrics are able to target all the essential attributes of function-based software as mentioned by Selby [13] in his latest findings, so we analyzed, refined and used the following metrics of to explore different structural dimensions of a component:

- Cyclometric Complexity Using Mc Cabe's Measure [14],
- Regularity Metric[15]
- Halstead Software Science Indicator[16]
- Reuse Frequency Metric[15]
- Lack of Coupling Metric[12].

2. Parse the software to generate the Meta information related to the above Software Metrics. .
3. Evaluate the following approaches for the Modeling of the reusability data provided in the step 2.

a) Fuzzy Based approach: The steps of the fuzzy based approach are:

- Deciding the Membership functions
- optimized selection of initial rule-base for the Fuzzy Inference System (FIS)
- Perform the Inference
- Calculate the % accuracy of the results

b) Neuro-fuzzy based approach: The steps for the Neuro-Fuzzy based are as follows:

- Build Fuzzy Inference System as discussed in fuzzy based approach
- Load the Training data NF system with the training data
- Training of the NF based system is performed using a hybrid learning algorithm using both least squares method and back-propagation. In the forward pass the consequent parameters are identified using least squares and in the backward pass the premise parameters are identified using back-propagation.
- After traing pahse the trained NF system is tested with the testing data
- In the testing phase the results are expressed in terms of % accuracy

c) Hybrid fuzzy-GA based approach: The steps of the hybrid fuzzy-GA algorithm are:

- Read the input as the metric values
- Find the nearest match with Example data using Euclidean Distance

- Calculate the Output of the fuzzy Inference System corresponding to the Input set
- Treat FIS value and the Nearest Match Value as Chromosome and convert the values into Binary after multiplying the values with 100
- Perform the cross over of the Values at a Particular Point
- Compare the results

4. Select the best approach from the step 3 and make reusability evaluation model for the function based software components.

III. RESULTS AND DISCUSSION

The proposed Fuzzy and Neuro-fuzzy based methodology is implemented in MATLAB 7.2. MATLAB (Matrix Laboratory) environment is one such facility which lends a high performance language for technical computing. The Fuzzy-GA implementation is done in C programming language. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made. Following figure gives the snap shot of MATLAB window while using FIS editor for 5 inputs and 1 output.

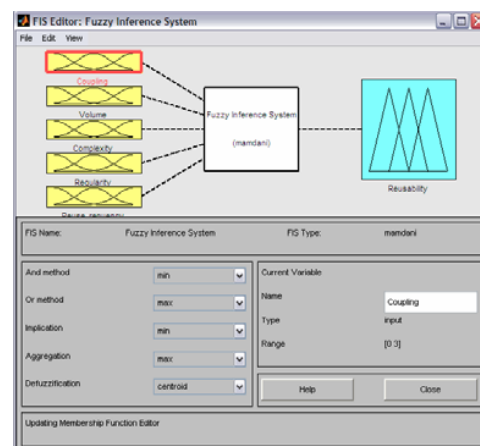


Fig. 1 Fuzzy Inference System with 5 input and 1 output

Using a given input/output data set, the toolbox function `anfis` constructs a fuzzy inference system whose membership function parameters are tuned or adjusted using either a back-propagation algorithm alone, or in combination with a least squares type of method. This allows the fuzzy systems to learn from the data they are modeling.

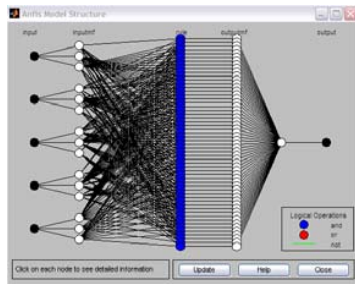


Fig. 2 ANFIS model structure

Fig. 2 shows the overall ANFIS model structure. This model shows five inputs, three membership functions for each input, 56 rules, output membership functions and one output i.e. Reusability.

TABLE I
TRUTH TABLE FOR PERCENTAGE ERROR FOR NF AND FUZZY SYSTEM

Cp	Vol	Cm	Reg	R-F	Reu	Fo	% error	Nf	% Error
3	1	1	1	1	1	1.5	0.5	1.49	0.49
3	3	3	2	1	2	1.5	0.25	1.98	0.01
3	3	1	2	1	2	1.5	0.25	1.98	0.01
3	3	2	2	1	2	1.5	0.25	1.98	0.01
3	1	2	2	1	3	2.49	0.17	2.76	0.08
2	3	2	2	1	3	1.99	0.336	2.92	0.267
2	1	1	2	1	3	3.00	0.00	3.10	0.03
2	2	2	2	1	3	2.5	0.166	3.58	0.193
1	2	2	2	1	4	3.0	0.25	3.85	0.037
1	2	2	3	2	2	3.51	0.755	5.15	1.575
1	2	2	3	3	5	4.99	0.002	5.15	0.03
3	1	2	2	1	3	2.49	0.17	3.1	0.03
2	3	2	1	3	6	3.51	0.415	4.10	0.317
1	3	3	2	3	4	1.5	0.625	3.10	0.225

Table I shows the 15 data values taken from the data provided by the different companies to calculate the percentage error and to discuss about the accuracy of the fuzzy logic output and the Neuro-fuzzy output. The fuzzy based inference system and Neuro-fuzzy system are evaluated on the basis of percentage error. The calculations for average output

errors of both fuzzy-logic and neuro-fuzzy are as follows:

Sum of percentage error of fuzzy logic output = 4.13

Average fuzzy logic output error = $4.13/14 = 0.295$

Sum of percentage error of neuro-fuzzy output = 3.285

Average neuro-fuzzy logic output error = $3.285/14 = 0.2346$

In Table I, Cp, Vol, Cm, Reg, R-F, Reu, Fo and N-f notations are used for Coupling, Volume, Complexity, Regularity, Reuse- Frequency, Reusability, Fuzzy logic output, and Neuro-fuzzy output respectively.

In the Table II, the results of Fuzzy-GA implementation are shown. Cp, Vol, Cm, Reg, R-F, Reu, Fo, F-G1 and F-G2 notations are used for Coupling, Volume, Complexity, Regularity, Reuse- Frequency, Reusability, Fuzzy logic output, Fuzzy-GA output1 and Fuzzy-GA output2 respectively.

The overall comparison of the three approaches used is shown in Table III. After analyzing the table it is deduced that the hybrid Fuzzy-GA approach produces less % error in evaluating a software component.

TABLE II
TRUTH TABLE FOR PERCENTAGE ERROR FOR FUZZY-GA SYSTEM

Cp	Vol	Cm	Reg	R-F	Reu	Fo	% error	F-G1	F-G2
3	1	1	1	1	1	1.5	0.5	2.3	1.48
3	3	3	2	1	2	1.5	0.25	2.02	1.48
3	3	1	2	1	2	1.5	0.25	2.02	1.48
3	3	2	2	1	2	1.5	0.25	2.02	1.48
3	1	2	2	1	3	2.49	0.17	0.45	2.48
2	3	2	2	1	3	1.99	0.336	0.47	1.96
2	1	1	2	1	3	3.00	0.00	3.00	3.00
2	2	2	2	1	3	2.5	0.166	0.46	2.48
1	2	2	2	1	4	3.0	0.25	4.0	3.0
1	2	2	3	2	2	3.51	0.755	4.58	3.48
1	2	2	3	3	5	4.99	0.002	5.02	4.96
3	1	2	2	1	3	2.49	0.17	0.45	2.48
2	3	2	1	3	6	3.51	0.415	0.9	3.48
1	3	3	2	3	4	1.5	0.625	1.46	1.48

TABLE III
COMPARISON ON THE BASIS OF PERCENTAGE ERROR

% Error (fuzzy logic output)	%Error (Neuro-fuzzy logic output)	%error (fuzzy-GA)
0.5	0.49	0.48
0.25	0.01	0.01
0.25	0.01	0.01
0.25	0.01	0.01
0.17	0.08	0.01
0.3367	0.267	0.01
0.00	0.03	0.0
0.1667	0.193	0.173
0.25	0.0375	0.00
0.755	1.575	0.74
0.002	0.03	0.004
0.17	0.03	0.17
0.415	0.317	0.42
0.625	0.225	0.63

IV. CONCLUSION

Fuzzy-GA hybrid algorithm is proved to be best as compared to the other algorithms considered in this work. In such data search application the design and developed fuzzy GA code has shown its superiority because it includes the advantages of fuzzy as well as genetic algorithms. Fuzzy provides a robust inference mechanism with no learning and adaptability while on the other hand, the genetic algorithms provide an efficient data modification in the wake of optimization objectives of given application. Neuro-fuzzy algorithm is definitely superior to fuzzy algorithm as it inherits adaptability and learning but seriously lacks optimal nature. From the simulation and the result obtained, it has been shown that the percentage average error is least in the case of fuzzy-GA algorithms and maximum in the case of fuzzy algorithms. Neuro-fuzzy algorithm has yielded accuracy lying between the accuracy levels as in the case of fuzzy and fuzzy-GA algorithms. It is concluded that for non linear and complex

engineering applications involving control, inference and analysis by and large fuzzy-GA is an efficient technique.

REFERENCES

- [1] E. Smith, A. Al-Yasiri, and M. Merabti, "A Multi-Tiered Classification Scheme for Component Retrieval," Proc. Euromicro Conference, 1998, 24th Volume 2, 25-27 Aug. 1998, pp. 882 – 889.
- [2] V.R. Basili, "Software Development: A Paradigm for the Future," Proc. COMPAC '89, Los Alamitos, Calif.: IEEE CS Press, 1989, pp. 471- 485.
- [3] B. W. Boehm and R. Ross, "Theory-W Software Project Management: Principles and Examples," IEEE Trans. Software Eng., vol.15, no. 7, 1989, pp. 902.
- [4] W. Lim, "Effects of Reuse on Quality, Productivity, and Economics," IEEE Software, vol. 11, no. 5, Oct. 1994, pp. 23-30.
- [5] H. Mili, F. Mili and A. Mili, "Reusing Software: Issues and Research Directions," IEEE Transactions on Software Engineering, Volume 21, Issue 6, June 1995, pp. 528 - 562.
- [6] G. Caldiera and V. R. Basili, "Identifying and Qualifying Reusable Software Components", IEEE Computer, February 1991, pp. 61-70.
- [7] W. Humphrey, Managing the Software Process, SEI Series in Software Engineering, Addison-Wesley, 1989.
- [8] L. Sommerville, Software Engineering, Addison-Wesley, 4th Edition, 1992.
- [9] R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill Publications, 5th edition, 2005.
- [10] G. Boetticher and D. Eichmann, "A Neural Network Paradigm for Characterising Reusable Software," Proceedings of the 1st Australian Conference on Software Metrics, 18-19 November 1993.
- [11] S. V. Kartalopoulos, Understanding Neural Networks and Fuzzy Logic-Basic Concepts and Applications, IEEE Press, 1996, pp. 153-160.
- [12] Parvinder Singh Sandhu and Hardeep Singh, "Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach", International Journal of Information Technology, vol. 3, no. 3, 2006, pp. 209-214.
- [13] Richard W. Selby, "Enabling Reuse-Based Software Development of Large-Scale Systems", IEEE Trans. Software Eng., 31(6), (June 2005), pp. 495-510.
- [14] T. McCabe, "A Software Complexity measure", IEEE Trans. Software Eng., vol. SE-2 (December 1976), pp. 308-320.
- [15] G. Caldiera and V. R. Basili, Identifying and Qualifying Reusable Software Components, IEEE Computer, (1991), pp. 61-70.
- [16] Maurice H. Halstead, Elements of Software Science (Elsevier North-Holland, New York, 1977).