I.J. Modern Education and Computer Science, **2015, 10, 27-32**
Published Online October 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2015.10.04

Measuring of Software Maintainability Using Adaptive Fuzzy Neural Network

Mohammad Zavvar
Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
E-mail: zavvar.developer@gmail.com

Farhad Ramezani
Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
E-mail: ramezani.farhad@iausari.ac.ir

Abstract—Software maintenance mainly refers to the process of correcting software after delivery. Maintenance process is usually a high percentage of Organizational effort to the whole process of software programs. As a result, the effectiveness of the entire production process and customer satisfaction is dependent on the effectiveness of maintenance activities. Because many factors including type of service, type of product and human factors is dependent on the maintenance process, And the imprecise nature of qualitative factors and sub-criteria leading software maintenance, accurate assessment can be maintained in order to measure the effectiveness of programs seem highly desirable. In this paper, using adaptive fuzzy neural network to provide a method for evaluating the capability of software maintenance conducted after the tests, the root mean square error of the proposed method was equal to 0.34331. The results show that the method is based on adaptive fuzzy neural, maintainability software performance evaluation is appropriate.

Index Terms—Adaptive Fuzzy Neural Network, maintenance, Software, Measuring.

I. Introduction

The software quality is a grade that the software is a combination of desirable properties.[1] The software system quality depends on the functional and non functional requirements. Needs and characteristics play a major role in defining the quality of software. Software quality is a vital force in the development of the software industry. By this time, events that have occurred because of the quality of software, are common events. Especially heavy losses that have occurred in the areas of application, such as the fields of military, economic and national security, human shock has arrived.[2] The software quality is still a concern. Entering the 21st century, the development of human society to the network and the data has been entered. The scale is bigger and bigger investments. The complexity and the difficulty grows day by day. The risks are higher.

Decades of experience show that the quality control software in a short period due to the unique characteristics of software products, the general can not be resolved. It is essential that new methods and techniques and quality control software continuously developed in terms of long-term research and operational software engineering and software quality has gradually improved.[3] The relationship between needs and characteristics that Commenting on the quality of a software product that might be possible. In the definition of the characteristics and features, there is a quality. The quality characteristics which the products more attractive, more useful, faster and produced more credible.[4]

Table 1. Outstanding Qualitative Models with Features [5-9]

Models					Parameters
Dromey	**ISO 9126**	**FURPS**	**Boehm**	**McCall**	
✓	✓			✓	Reusability
				✓	Flexibility
✓	✓		✓	✓	Efficiency
			✓		Utility
		✓			Performance
✓	✓				Maturity
		✓			Supportability
✓	✓		✓	✓	Portability
✓	✓	✓			Functionality
				✓	Correctness
✓	✓	✓	✓	✓	Reliability
✓	✓	✓	✓	✓	Usability
				✓	Integrity
✓	✓	✓	✓	✓	Maintainability

The following features and different features to define the quality of a software product from researchers, organizations and various institutions have been suggested that many of these features and the features of different models of quality are the same. In Table1, the number of outstanding qualitative models with features or characteristics they'll pay more. Looking at the Table1 above and the features that each quality model to evaluate the quality of a system, capable of holding one of the most widely used software features, we find The well-known and popular models of quality, great position and it is important to address.

Software maintainability, one of the most important topics in the field of software quality.

Software maintenance, software product is able to be changed. Changes could include modifications, enhancements or software adaptation to changing environments and needs, and operational characteristics.[10] There are two major phases in the software life cycle: Phase of maintenance& phase of the software development increase. Keeping a software system, although apparently the final phase of the product development cycle, but Yip showed that almost 66% of the total cost of the software life cycle to be spent on software maintenance phase.[11]

Because of the high cost of maintenance software, a software client demand systems that are easier to maintain, so the client software is often less potential maintenance software tools that are learning it.[12]

Maintenance activities are generally carried out four objectives:[13, 14]

- **Corrective maintenance**: changes to the software to fix the system errors discovered after delivery is carried out includes all changes that eliminate faulty operation of the software.
- **Adaptive maintenance**: changes to the software after delivery, can be used to keep the program in place have been changed or are changing. Include environmental changes needed to implement the system with the system should be operated. For example, changes to its work on a new hardware platform, the new operating system.
- **Perfective maintenance:** changes to the software product after delivery to improve performance or the repair is done. Includes changes that organized by the user's request. For example, such as add, delete and modify functions, rewriting documentation, improve efficiency or improve ease of use noted. Ideally this should repair operations reliability and reduce repair or disrupt the structure of the system components. Otherwise, changes are coming gradually more difficult and the cost of their implementation will be more and more expensive.
- **Preventive maintenance:** includes all the improvements that makes a piece of software to be repaired.

However, due to the imprecise nature of the requirements in the early phase of production, evaluate the effectiveness of the production process and customer satisfaction can significantly to activities related maintenance phase Due to the imprecise nature of some of the engineering systems, including the ability to assess the maintenance of the angles of analysis, variability, the ability to retrieve and test-taking ability is very important.[15, 16]

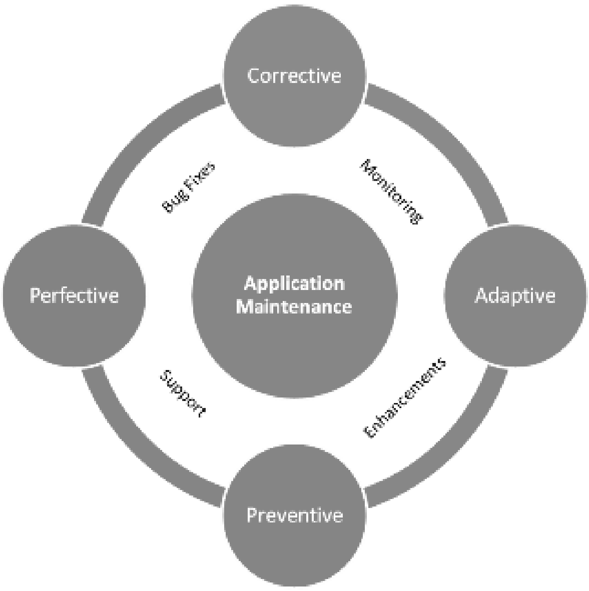

Fig.1. Objectives of Maintenance

It is also not uncommon in the software development organizations that exert control over their developed software maintenance. In addition, although the evaluation software maintenance software developers manage them, reluctant to share this information with their clients. From the perspective of an organization's software development resources to develop software runs under the maintenance easier, the first cost-effective best practices, But by providing a workable development costs for software development and maintenance easier, more income in the hands of the maintenance phase. The maintenance can be performed by a number of factors, including the characteristics of the code, enforcement of maintenance conditions, tools used for maintenance and affected. Besides the fact that technology is constantly changing, meaning that the new standards for architecture and design software systems are constantly being established. But may be one of the standards on all factors that can affect the maintenance of software systems, not scientific.

Evaluation of predictive maintenance is a software system based on existing information systems and effort is spent on maintenance. The strengths and weaknesses of experts, formal methods and compounds that have been studied in the context of estimating software. Results show that a combination of professional evaluation and formal methods usually provide the best results.[17]

However, significant differences between the estimate and assess the maintenance software development efforts there, but there are similarities that suggest future studies on the role of vocational assessment, it can also be useful in the field of software maintenance.

Software maintenance is often costly. The strategy to assess the full maintenance of software systems is very important. Therefore, given the importance of assessing the ability of software maintenance, this article provides a method based on adaptive fuzzy neural network to evaluate the effectiveness of software maintenance.

II. Related Works

In this section some works and studies in the field of software maintenance pay.

Bandini and colleagues[18], three factors independent of the complexity of the task of maintenance and the ability to predict the performance of maintenance programmer object-oriented systems examined. To scale factor design complexity, size, interface, interaction and complexity of the operation were selected argument. The second factor is the duty of maintenance; two types of corrective maintenance were selected.

Several problems by Kajko and colleagues[19] in relation to the maintenance of software systems were discussed and maintenance models for product and process were presented separately. Product features include the usual features, features variable maturity level tracking features, etc., while in feature process, responsibilities for maintenance management through the entire life cycle of software.

The [20] fuzzy model to measure the amount of maintenance software is used by Aggarwal et al. writers consider four influential factors , namely, average number of lived variables, average life expectancy of variables, average complexity rotating . All inputs into fuzzy values of low, medium and high were classified, and in turn, keep the amount of very good, good, fair, poor and very poor classified. All inputs and all 81 of the model are presented. However, this model is not valid on complex projects in the real world.

III. The Proposed Method

A neuro fuzzy inference system, a neural network that is based on Takagi-Sugeno fuzzy system. This method was created in the early 1990s. Because of this system unify neural network and fuzzy logic of the concepts with each other,Both features can be used in a frame.[21] View of ANFIS structure shown in Fig.2.

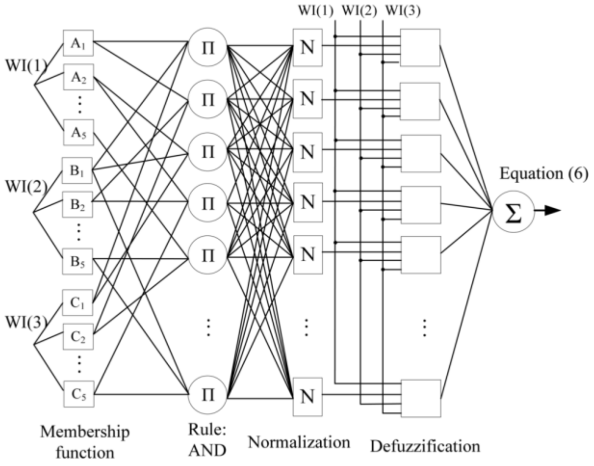

Fig.2. ANFIS Structure

In the first layer, all the nodes are adaptive and output layer is the degree of input fuzzy membership that relationship (1) is:

$$O_i^1 = \mu_{A_i}(x) \quad \text{i=1, 2}$$

$$O_i^1 = \mu_{B_{i-2}}(x) \quad \text{i=3, 4} \tag{1}$$

$\mu_{A_i}(x)$ & $\mu_{B_{i-2}}(x)$ can be a fuzzy membership function. For example, if the Gaussian membership function used in equation (2) is calculated.

$$\mu_{A_i}(x) = e^{\frac{-(x-c_i)^2}{2\sigma_i^2}} \tag{2}$$

σ_i and ci are the parameters of membership function accordingly shaped Gaussian function.

In the second layer, nodes which are fixed and expressed as a simple multiplier operate. This layer can be output (3) stated:

$$O_i^2 = \varpi_i = \mu_{A_i}(x)\mu_{B_i}(y) \quad \text{i=1, 2} \tag{3}$$

This can be interpreted as the power to fire regulations. The third layer is fixed and with N nodes shown and the normal role of the fire power of the previous layer acts. This layer has an output connection (4) is:

$$O_i^2 = \varpi_i = \frac{\varpi_i}{\varpi_1 + \varpi_2} \quad \text{i=1, 2} \tag{4}$$

This layer can be interpreted normalized firepower. The fourth layer, nodes, and output nodes comparative product of the normal fire power and first-degree polynomial model is primarily Sugeno. The output of this layer can be in the form of equation (5) stated:

$$\begin{aligned} O_i^4 &= \varpi_i f_i = \\ &\varpi_i (p_i x + q_i y + r_i) \end{aligned} \quad \text{i=1, 2} \tag{5}$$

The fifth layer is a fixed node labeled S is the sum of the output signal input is expressed as Equation (6).

$$\overline{O_i^5} = \sum_{i=1}^{2} \varpi_i f_i = \frac{\sum_{i=1}^{2} \varpi_i f}{\varpi_1 + \varpi_2} \tag{6}$$

The task of learning algorithms for the ANFIS, the parameters can be modified {ci and σ_i } and {r_i and p_i and q_i} output ANFIS to coordinate with the training data. When the above parameters{c_i and σ_i }are fixed output ANFIS model can be expressed as Equation (7):

$$f=\frac{\varpi_1}{\varpi_1+\varpi_2}f_1+\frac{\varpi_2}{\varpi_1+\varpi_2}f_2 \qquad (7)$$

By putting the equation (3) in equation (7) to (8) we see:

$$f=\varpi_1 f_1+\varpi_2 f_2 \qquad (8)$$

And put the fuzzy rules in the equation (8) to (9) will be:

$$\begin{aligned}f&=(\varpi_1 x)p_1+(\varpi_1 y)q_1+(\varpi_1)r_1\\&+(\varpi_2 x)p_2+(\varpi_2 y)q_2+(\varpi_2)r_2\end{aligned} \qquad (9)$$

Delayed a linear order of the parameters are {r_i and p_i and q_i} to identify the optimal values of the parameters of the least squares method is used. Least square method to optimize parameters delayed with fixed parameters used before. Once the optimal parameters were found posterior immediately reverse course begins. Descending error reduction method for setting optimal parameters related priority fuzzy sets used in the input range.

ANFIS output using the parameters found in the path forward will be calculated later. Output error to set the parameters preceded by standard error propagation algorithm is used.

We intend to continue using ANFIS to assess the maintenance of their software. For training and testing the network of data collection software projects are used. This dataset contains 6 inputs and one output.[22] The output value equal to the amount assessed for the ability to maintain software is based aspects. The input data set are as follows:

- **Modularization:** the libertine that shows how an independent software component is formed; So that changes in one component does not have much effect on others.
- **Variability:** in the amount that can be as efficient and effective system of change that this change does not have a devastating impact on system performance. In fact, this criterion amount of user effort, backup and system to implement reform by the measure.
- **Validation and Testing:** This is the amount of effort the user, backup and system test time correction system or not, is measured.
- **Ability to read:** the amount of internal and external documentation
- **Return on investment:** the amount of work value due to the return on investment
- **The size of the project team:** the size of the team and cycles

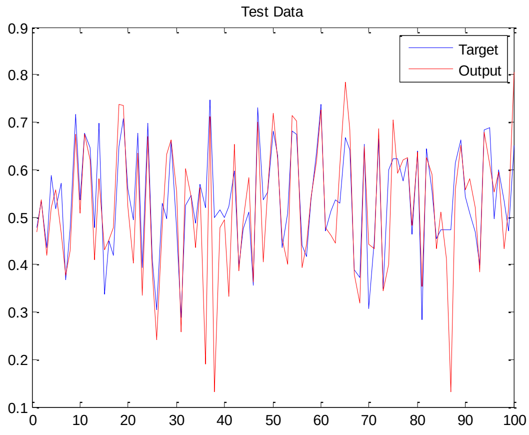

Fig.3. Ability to Maintain Application Performance Assessment Model

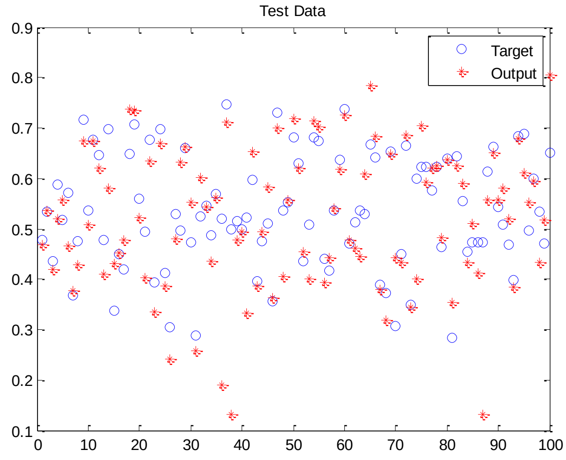

Fig.4. Ability to Maintain Application Performance Assessment Model

One of the criteria used in the evaluation model, root mean square error between the predicted and measured values. RMSE is a useful tool for comparing the forecasting errors. This value is calculated using the following equation:

$$RMSE=\sqrt{\frac{\sum_{i=1}^{n}(y_i-y)^2}{n}} \qquad (10)$$

This index first calculates the individual differences observed with predicted values by the model and the power of the heart. The square root of the mean difference and the number of offers is the same RMSE.[23] This index is a measure of accuracy. If the model is generally consistent with the data better , the amount is less.[23] The diagram in Fig5, root mean square error for the model shown.

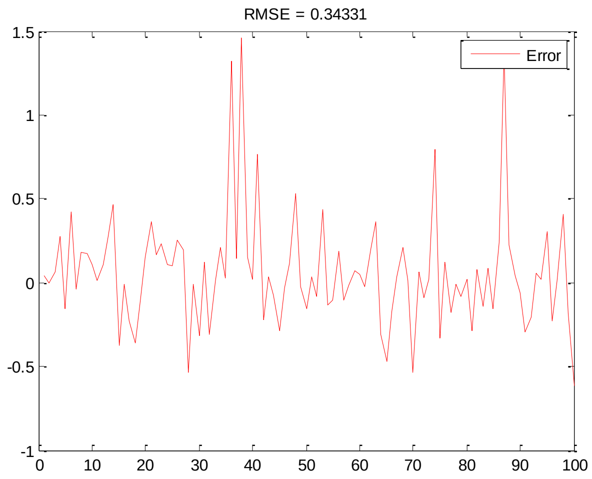

Fig.5. Error Diagram for Model

As can be seen in Fig5, the square root of the sum of squared errors of the proposed method is equivalent to 0.34331 Good value for this standard. Fig6 shows the mean value and standard deviation of error in the model with histograms. STD is one of the indicators that show the distribution of data on average how much the average value away.[24] If the standard deviation of the data set is close to zero, it means that the data are close to the average and dispersion are small, while large standard deviation indicates a significant distribution data. Standard deviation equal to the square root of the variance.[25] As can be seen in Fig6, STD of the proposed method is equivalent to 0.33972

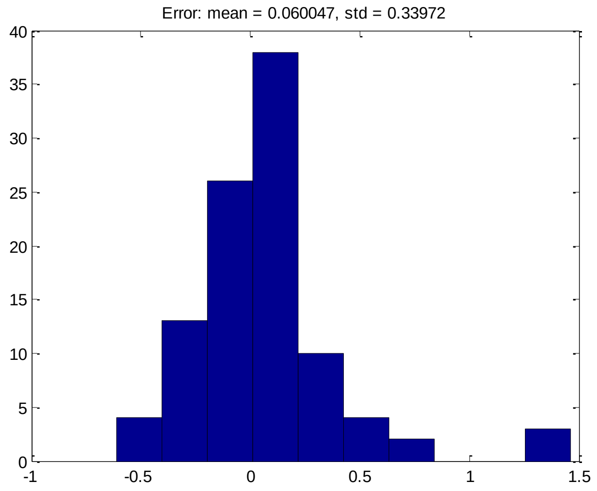

Fig.6. The Mean Value and Standard Deviation of Error with the Proposed Model Charts Histfit

IV. Conclusion

In this paper, the evaluation of the importance of maintenance software, to provide a method based on adaptive fuzzy neural network to assess the ability of maintenance software. More than 70 percent data and 30 percent training data we use to test the network. In this system, the root mean square error was equal to 0.34331 with respect to the content, accuracy and performance of the proposed method is suitable for evaluating the capability of software maintenance.

Refrences

[1] EEE, I., Standard G lossary of SoftwareEngineering Terminology. IEEE Software Engineering Standards & oll ecti o n. I EEE, 1990: p. 610.12-190.

[2] Mordal, K., et al., Software quality metrics aggregation in industry. Journal of Software: Evolution and Process, 2013. 25(10): p. 1117-1135.

[3] Ampatzoglou, A., G. Frantzeskou, and I. Stamelos, A methodology to assess the impact of design patterns on software quality. Information and Software Technology, 2012. 54(4): p. 331-346.

[4] Robertson, S. and J. Robertson, Mastering the requirements process: Getting requirements right. 2012: Addison-wesley.

[5] McCall, J.A., P.K. Richards, and G.F. Walters, Factors in software quality. volume i. concepts and definitions of software quality. 1977, DTIC Document.

[6] Boehm, B.W., J.R. Brown, and H. Kaspar, Characteristics of software quality. 1978.

[7] Grady, R.B. and D.L. Caswell, Software metrics: establishing a company-wide program. 1987.

[8] Kan, S.H., Metrics and models in software quality engineering. 2002: Addison-Wesley Longman Publishing Co., Inc.

[9] Dromey, G.R., A model for software product quality. Software Engineering, IEEE Transactions on, 1995. **21**(2): p. 146-162.

[10] Al-Qutaish, R.E., Measuring the Software Product Quality during the Software Development Life-Cycle: An International Organization for Standardization Standards Perspective. Journal of Computer Science 5 (5), 2009.

[11] Yip, S.W. and T. Lam. A software maintenance survey. in Software Engineering Conference, 1994. Proceedings., 1994 First Asia-Pacific. 1994. IEEE.

[12] Bennett, K.H. and V.T. Rajlich. Software maintenance and evolution: a roadmap. in Proceedings of the Conference on the Future of Software Engineering. 2000. ACM.

[13] Grant, S., J.R. Cordy, and D.B. Skillicorn. Using topic models to support software maintenance. in Software Maintenance and Reengineering (CSMR), 2012 16th European Conference on. 2012. IEEE.

[14] Sjøberg, D.I., B. Anda, and A. Mockus. Questioning software maintenance metrics: a comparative case study. in Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement. 2012. ACM.

[15] Anquetil, N., et al., Software maintenance seen as a knowledge management issue. Information and Software Technology, 2007. 49(5): p. 515-529.

[16] Webster, K.P.B., K.M. de Oliveira, and N. Anquetil. A risk taxonomy proposal for software maintenance. in Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on. 2005. IEEE.

[17] Rajlich, V. Software evolution and maintenance. in Proceedings of the on Future of Software Engineering. 2014. ACM.

[18] Bandini, S., Paoli, F. D., Manzoni, S., Mereghetti, P., A support system to COTS based software development for business services. Proceedings of the 14th International

Conference on Software Engineering and Know ledge Engineering, Ischia, Italy, 2002. Vol. 27: p. 307–314.

[19] Kajko-Mattsson, M., et al., A model of maintainability–suggestion for future research. 2006.

[20] Aggarwal, K., et al., Measurement of software maintainability using a fuzzy model. Journal of Computer Science, 2005. **1**(4): p. 538.

[21] Wai, R.-J. and Z.-W. Yang, Adaptive fuzzy neural network control design via a T–S fuzzy model for a robot manipulator including actuator dynamics. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2008. 38(5): p. 1326-1346.

[22] Dit, B., et al. A dataset from change history to support evaluation of software maintenance tasks. in Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on. 2013. IEEE.

[23] Chai, T. and R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE)? Geoscientific Model Development Discussions, 2014. **7**: p. 1525-1534.

[24] Oseni, O.F., et al., Comparative Analysis of Received Signal Strength Prediction Models for Radio Network Planning of GSM 900 MHz in Ilorin, Nigeria. 2014.

[25] Ronquist, F., et al., MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. Systematic biology, 2012. **61**(3): p. 539-542.

Authors' Profiles

Mohammad Zavvar has got M.S.c in Software Engineering from the Sari Azad University. And now, to research and teaching in the field of programming, image processing, fuzzy systems, neural networks, data mining, software engineering and algorithm optimization. He also has published numerous articles on various topics in the field of software engineering and information technology in conferences and international journals.

Farhad Ramezani was born in 1982 in Babol, I.R.Iran. He studied computer Engineering at the Iran University of Science & Technology where he received the B.Sc. degree in 2006. He received the M.Sc. degree in artificial intelligence in 2010 under the direction of Prof. M. Yaghobi and Prof. A.A.Ghangherme with a thesis on Caspian Sea Level Prediction by Auto Fuzzy Regression from the Islamic Azad University Mashhad Branch, Mashhad. He is currently PhD Candidate at Islamic Azad University, Science and Research Branch, and Faculty Member of Computer Engineering Department of Islamic Azad University Sari Branch, Sari, Iran. His primary research interests include the area of intelligent systems, fuzzy set theory and Image processing.