

Development of Software Reliability Growth Models for Industrial Applications Using Fuzzy Logic

Sultan Aljahdali
Department of Computer Science,
College of Computers and Information Technology,
Taif University, Taif, Saudi Arabia

Abstract: Problem statement: The utilization of Software Reliability Growth Models (SRGM) plays a major role in monitoring progress, accurately predicting the number of faults in the software during both development and testing processes; define the release date of a software product, helps in allocating resources and estimating the cost for software maintenance. This leads to achieving the required reliability level of a software product. **Approach:** We investigated the use of fuzzy logic on building SRGM to estimate the expected software faults during testing process. **Results:** The proposed fuzzy model consists of a collection of linear sub-models, based on the Takagi-Sugeno technique and attached efficiently using fuzzy membership functions to represent the expected software faults as a function of historical measured faults. A data set provided by John Musa of bell telephone laboratories (i.e., real time control, military and operating system applications) was used to show the potential of using fuzzy logic in solving the software reliability modeling problem. **Conclusion:** The developed models provided high performance modeling capabilities.

Key words: Software Reliability Growth Models (SRGM), Takagi-Sugeno technique, Fuzzy Logic (FL), Artificial Neural Net-works (ANN), Genetic Programming (GP), model structure, linear regression model, NASA space

INTRODUCTION

Machine Learning (ML) and Soft Computing techniques, such as Genetic Algorithms (GAs), Genetic Programming (GP), Evolutionary Strategies (ESs), Artificial Neural Net-works (ANN), Fuzzy Logic (FL) and Particle Swarm Optimization (POS), to solve software engineering problems expanded in the recent years. Sheta (2007), author explored the use of Particle Swarm Optimization (PSO) algorithm to estimate SRGM parameters. The proposed method shows significant advantages in handling variety of modeling problems such as the Exponential Model (EXPM), power model (POWM) and Delayed S-Shaped Model (DSSM). Parameter Estimation of Hyper-Geometric Distribution Software Reliability Growth Model by Genetic Algorithms was presented Minohara and Tohma (1995). Predicting accumulated faults during the software testing process using parametric and non-parametric models were explored in many articles (Aljahdali *et al.*, 2001; Sheta, 2006). Zeng and Rine (2004), author provided a strategic solution for estimating software defect fix effort using self-organizing neural network. Genetic Programming (GP) was successfully used to find a

model that fits the given data points without making any assumptions about the model structure (Afzal and Torkar, 2008; Paramasivam, 2009). GP found to be a powerful technique in developing software reliability growth modeling.

Fuzzy logic has been successfully used to solve variety of problems in system identification, signal processing and control (White and Sofge, Wang and Mendel, 1993; Wang and Mendel, 1993; Brown and Harris, 1994). Fuzzy modeling has been regarded as one of the key problems in fuzzy systems research (Wang, 1992; Dubois and Prade, 1992). In the past years, research focused on the development of fuzzy systems from both theoretical and applications oriented prospective were presented Yager and Filev (1994), Lotfi (2002), Lotfi and Garibaldi (2004).

A fuzzy model structure can be represented by a set of fuzzy If-Then rules (Kosko, 1998). A fuzzy rule has two parts the antecedent and the consequence. The antecedent variables reflect information about the process operating conditions. The consequent of the rule is usually a linear regression model which is valid around the given operating condition (Babuska, 1996; Babuska *et al.*, 1996; Huang *et al.*, 1999; Sheta *et al.*, 2009; Abdelrahman *et al.*, 2009).

MATERIALS AND METHODS

In this study, we explore the use of fuzzy logic to predict faults during the software testing process using software historical faults data. The proposed fuzzy model structure is presented in section III. Detailed information about the data set and the experiments developed are provided in sections IV, VI.

Software reliability growth models: In the past three decades, hundreds of models were introduced to estimate the reliability of software systems (Musa, 1975; Xie, 2002). The issue of building growth models was the subject of many research work (Lyu, 1996; Musa, 2004) which helps in estimating the reliability of a software system before its release to the market. Serious application such as weapon systems and NASA space shuttle applications were explored (Carnes, 1997; Keller and Schneidewind, 1997; 1992).

Faults may be bump into market released software. This is a challenge for software companies. It might affect their reputation and finance. Software reliability growth models were significantly used to help in computing the number of faults which is still resides in the software (Teng and Pham, 2002). Thus, it is important to specify the effort required to fix faults, the time required before software can be released and the cost of repair. Software reliability growth models employ system experimental data for testing to predict the number of defects remaining in the software.

Software reliability models can be classified to two types of models according to prediction style either from: (1) the design parameters thus called “defect density” models (2) the test data thus “software reliability growth” models. Some known SRGM are Logarithmic, Exponential, Power, S-Shaped and Inverse Polynomial models (Farr, 1996; Jones *et al.*, 2001). They are typical analytical models. They normally describe the fault process as a function of execution time (or calendar time) and a set of unknown parameters. The model parameters normally estimated using least-square estimation or maximum likelihood techniques (Lyu, 1996).

RESULTS AND DISCUSSION

Proposed fuzzy model structure: Our objective is to approximation the dynamics of the fault measurements during the testing process and instead of representing it in a single nonlinear model we can extend it by a set of local linear models. Each local model should be able to represents the relationship between the historical faults $y(k-1)$, $y(k-2)$, $y(k-3)$, $y(k-4)$ and the current fault $y(k)$ in a certain range of operating conditions. Such a proposed fuzzy model structure can be successfully

represented by means of fuzzy If-Then rules. The proposed model equation is given as follows:

$$y(k) = FM(y(k-1), y(k-2), y(k-3), y(k-4)) \quad (1)$$

Using membership functions and the antecedent of the rule we can define the fuzzy region in the product space. The antecedent variables give the condition of the process status now. The consequent of the rule is typically a local linear regression model which relates $y(k)$ with $y(k-1)$, $y(k-2)$, $y(k-3)$, $y(k-4)$.

A rule-based fuzzy model requires the identification of the following: (1) the antecedent (2) the consequent structure, (3) the type of the membership functions for different operating regions and 4) the estimation of the consequent parameters. The developed fuzzy models implemented based the Takagi-Sugeno technique (Babuska, 1997; Babuska *et al.*, 1996). The proposed technique does not require any a prior knowledge about the operating regimes. If a sufficiently number of measurements are collected which reflects the operating ranges of interest, the developed fuzzy model will be an efficient one.

The software reliability data: John Musa of Bell Telephone Laboratories compiled a software reliability database Musa. His objective was to collect fault interval data to assist software managers in monitoring test status, predicting schedules and to assist software researchers in validating software reliability models. These models are applied in the discipline of software reliability engineering. The dataset consists of software fault data on 16 projects. Careful controls were employed during data collection to ensure that the data would be of high quality. The data was collected throughout the mid 1970s. It represents projects from a variety of applications including real time command and control, word processing, commercial and military applications.

In our case, we used data from three different projects. They are Real Time Control, Military and Operating System. A MATLAB toolbox for modeling of fuzzy systems (Babuska, 1998) was used to implement the following results. The routines of the toolbox contain the Gustanfson-Kessel (GK) clustering algorithm, whose implementation is given Gustafson and Kessel (1979).

Validation criteria: In order to check the performance of the developed model, we compute the Variance-Accounted-For (VAF) performance criterion to measure how close the measured values to the values developed using the fuzzy models. The VAF is computed as:

$$VAF = [1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}] \times 100\% \quad (2)$$

where, y , \hat{y} are the actual measured faults and the estimated fuzzy model faults, respectively.

CONCLUSION

We run the Fuzzy Model Identification Toolbox (Babuska, 1998) along with three membership functions. The data set was split into two parts: 1) 70% of the collected data for training and 2) 30% for testing (i.e., validation). The set of rules which describe the three software projects (i.e., Real Time Control, Military and Operating Systems) are presented in Tables 1-3.

In Fig. 1, we show the membership function for the real time control application. We used three clusters to build the fuzzy model. Figure 2 show the actual and predicted faults over the training and testing data for the real time control applications. The fuzzy membership functions for the military application and operating systems are shown in Figs. 3 and 5, respectively. In Figs. 4 and 6, we show the actual and predicted faults over the training, testing data for the military and operating systems applications.

The VAF criteria and reported in Table 4. It can be seen that applications. A fuzzy nonlinear regression models were developed for predicting the accumulated faults of software engineering applications. The developed fuzzy models implemented based the Takagi-Sugeno technique. The developed fuzzy models were tested using three types of applications. They are real-time control, military and operating systems applications. The data set was developed by John Musa of Bell Telephone Laboratories. The results were promising.

Table 1: Fuzzy rules for real time and control applications

If $y(k-1)$ is A11 and $y(k-2)$ is A12 and $y(k-3)$ is A13 and $y(k-4)$ is A14 then $\hat{y}(k) = 1.56 y(k-1) - 0.889 y(k-2) - 0.0643 y(k-3) + 0.210 y(k-4) + 3.82$ If $y(k-1)$ is A21 and $y(k-2)$ is A22 and $y(k-3)$ is A23 and $y(k-4)$ is A24 then $\hat{y}(k) = 1.56 y(k-1) - 0.0819 y(k-2) - 1.16 y(k-3) + 0.5.87 y(k-4) + 5.38$ If $y(k-1)$ is A31 and $y(k-2)$ is A32 and $y(k-3)$ is A33 and $y(k-4)$ is A34 then $\hat{y}(k) = 1.49 y(k-1) - 0.6.63 y(k-2) - 0.0674 y(k-3) + 0.235 y(k-4) + 0.612$

Table 2: Fuzzy rules for military applications

If $y(k-1)$ is A11 and $y(k-2)$ is A12 and $y(k-3)$ is A13 and $y(k-4)$ is A14 then $\hat{y}(k) = 1.53 y(k-1) - 0.676 y(k-2) + 0.687 y(k-3) - 0.492 y(k-4) - 1.96$ If $y(k-1)$ is A21 and $y(k-2)$ is A22 and $y(k-3)$ is A23 and $y(k-4)$ is A24 then $\hat{y}(k) = 2.36 y(k-1) - 0.59 y(k-2) - 1.84 y(k-3) + 1.07 y(k-4) - 1.27$ If $y(k-1)$ is A31 and $y(k-2)$ is A32 and $y(k-3)$ is A33 and $y(k-4)$ is A34 then $\hat{y}(k) = 0.41 y(k-1) + 0.0793 y(k-2) + 0.138 y(k-3) + 0.347 y(k-4) + 8.41$
--

Table 3: Fuzzy rules for operating system applications

If $y(k-1)$ is A11 and $y(k-2)$ is A12 and $y(k-3)$ is A13 and $y(k-4)$ is A14 then $\hat{y}(k) = 1.78 y(k-1) - 0.496 y(k-2) - 0.619 y(k-3) + 0.322 y(k-4) + 0.283$ If $y(k-1)$ is A21 and $y(k-2)$ is A22 and $y(k-3)$ is A23 and $y(k-4)$ is A24 then $\hat{y}(k) = 1.53 y(k-1) - 0.757 y(k-2) + 0.0933 y(k-3) + 0.123 y(k-4) + 2.52$ If $y(k-1)$ is A31 and $y(k-2)$ is A32 and $y(k-3)$ is A33 and $y(k-4)$ is A34 then $\hat{y}(k) = 1.93 y(k-1) - 1.11 y(k-2) - 0.00867 y(k-3) + 0.213 y(k-4) - 4.25$

Table 4: VAF for the Srgm using fuzzy logic

Project name	Training (VAF)	Testing (VAF)
Military	99.93 %	99.06 %
Real time control	99.86 %	95.28 %
Operating system	99.98 %	99.90 %

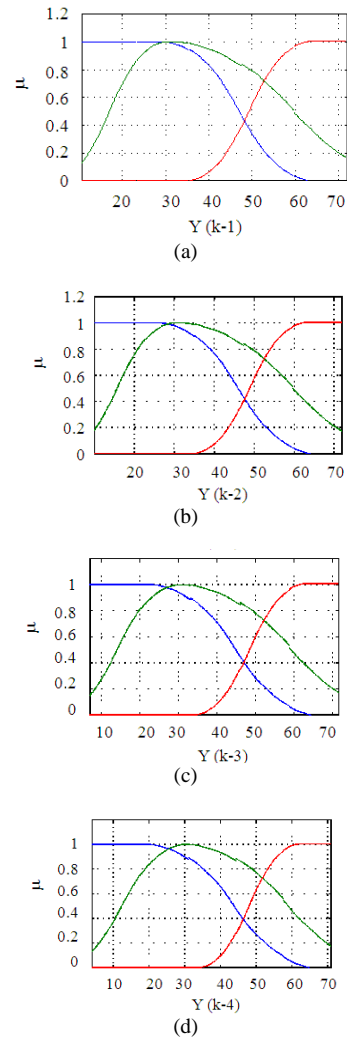


Fig. 1: Membership functions for the real-time control applications

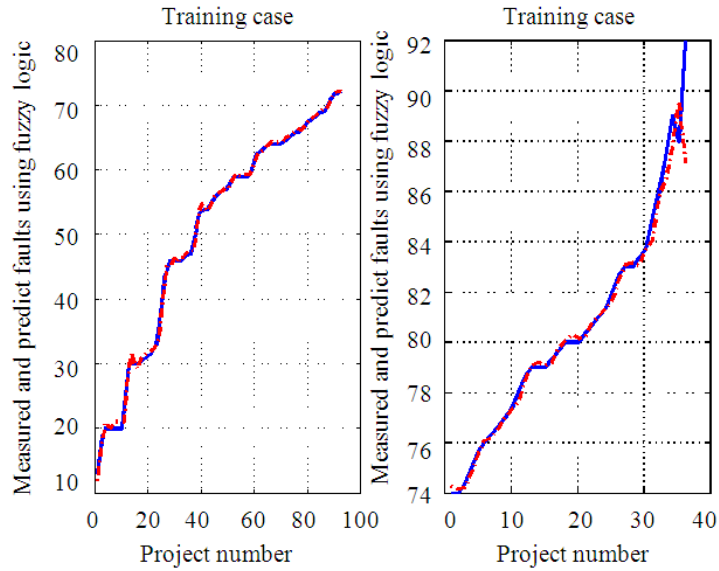


Fig. 2: Actual and estimated faults for Real-Time Control Applications

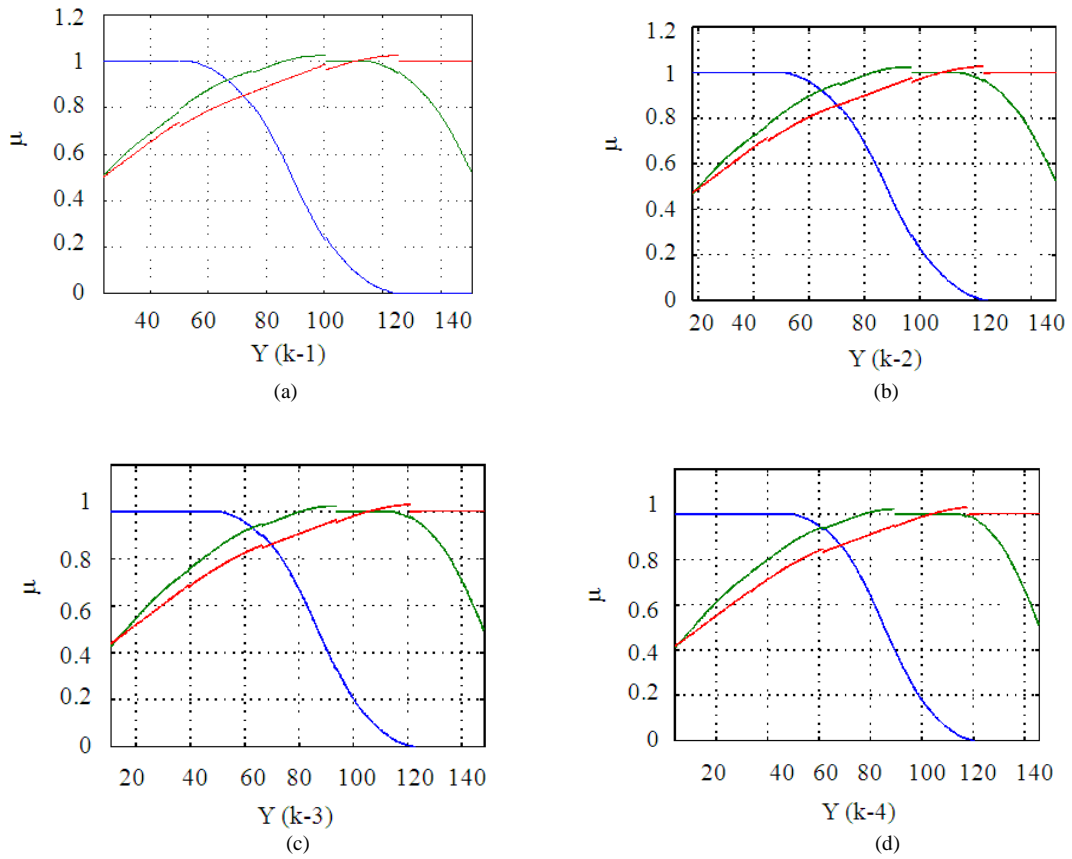


Fig. 3: Membership functions for the military applications

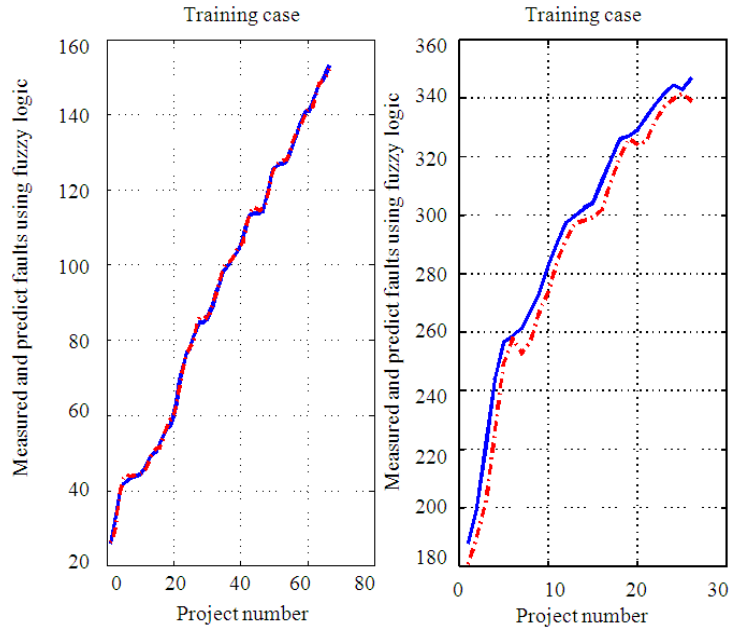


Fig. 4: Actual and estimated faults for military applications

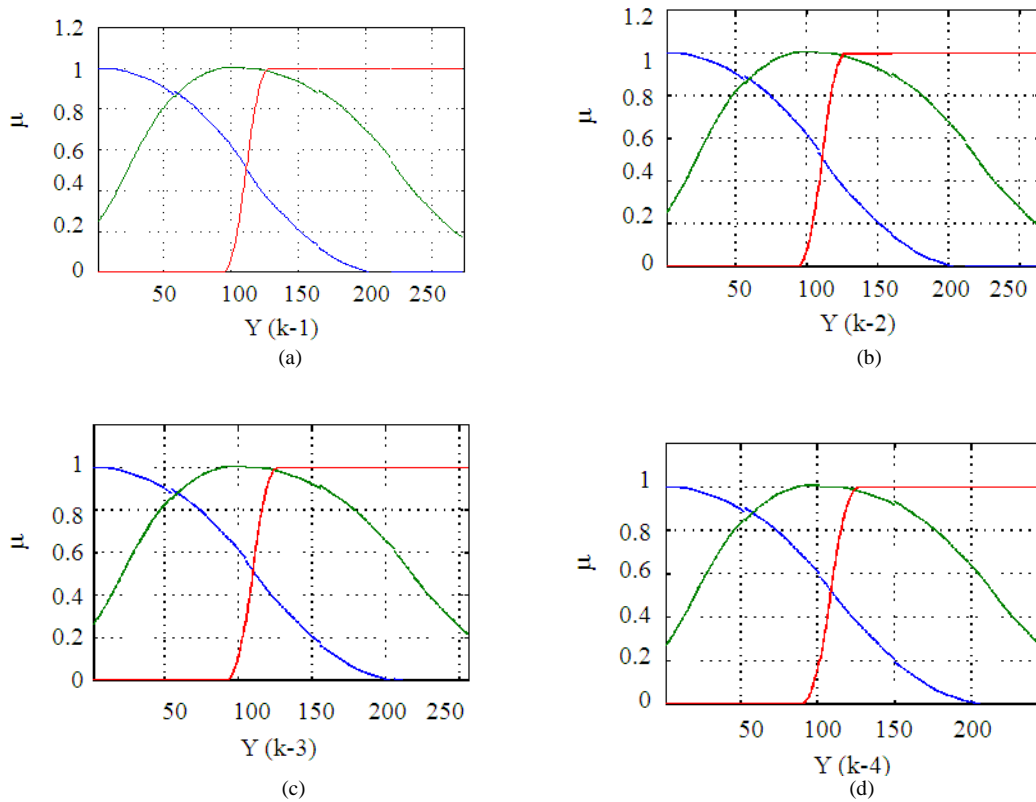


Fig. 5: Membership functions for the operating system applications

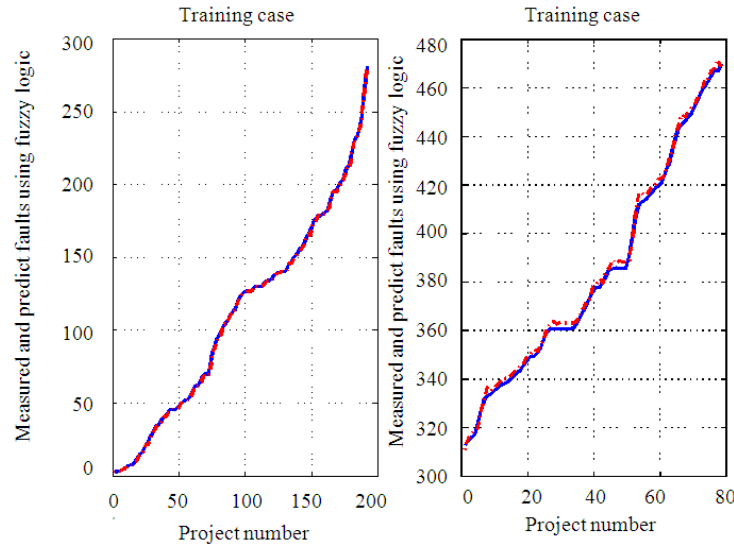


Fig. 6: Actual and estimated faults for operating system applications

ACKNOWLEDGMENTS

Authors would like to thank Prof. Babuska for providing the FMID Matlab Toolbox.

REFERENCES

- Abdelrahman, M., A. Sheta and W. Deabes, 2009. Fuzzy mathematical modeling for reconstructing images in ECT of manufacturing processes. Proceedings of the International Conference on Computer Engineering and Systems, Dec. 14-16, IEEE Xplore Press, Cairo, pp: 461-468. DOI: 10.1109/ICCES.2009.5383221
- Afzal, W. and R. Torkar, 2008. Suitability of genetic programming for software reliability growth modeling. Proceedings of the International Symposium on Computer Science and its Applications, Oct, 13-15, IEEE Xplore Press, Hobart, ACT, pp: 114-117. DOI: 10.1109/CSA.2008.13
- Aljehdali, S., D. Rine and A. Sheta, 2001. Prediction of software reliability: A comparison between regression and neural network non-parametric models. Proceeding of the ACS/IEEE International Conference on Computer Systems and Applications, June 25-29. IEEE Xplore Press, Beirut, Lebanon, pp: 470-473. DOI: 10.1109/AICCSA.2001.934046
- Babuska, H.A., R. Braake, A.J. Krijgsman and H.B. Verbruggen, 1996. Comparison of intelligent control schemes for real-time pressure control. Control Engin. Practice, 4: 1585-1592. DOI: 10.1016/0967-0661(96)00174-8
- Babuska, R., 1997. Fuzzy Modeling and Identification. 1st Edn., Delft, Techn. Univ., ISBN: 9090101535, pp: 294.
- Babuska, R., 1998. Fuzzy Modeling for Control. 1st Edn., Springer, Boston, ISBN: 0792381548, pp: 260.
- Brown, M. and C.J. Harris, 1994. Neurofuzzy Adaptive Modeling and Control. 1st Edn., Prentice Hall, New York, ISBN: 0131344536, pp: 508.
- Carnes, P., 1997. Software reliability in weapon systems. Proceeding of the 8th International Symposium on Software Reliability Engineering, Nov. 2-5, IEEE Xplore Press, Albuquerque, NM, USA, pp: 95-100. DOI: 10.1109/CSSRE.1997.637851
- Dubois, D. and H. Prade, 1992. Fuzzy sets in approximate reasoning: part 1. Fuzzy Sets Syst., 40: 143-202. DOI: 10.1016/0165-0114(91)90050-Z
- Farr, W.B., 1996. Software Reliability Modeling Surve. In: Hand-book of Software Reliability Engineering, Lyu, M.R., (Ed.), IEEE Computer Society Press, Los Alamitos, Calif. ISBN: 0070394008, pp: 850-850.
- Gustafson, D.E. and W.C. Kessel, 1979. Fuzzy clustering with a fuzzy covariance matrix. Elect. Electr. Eng. 86: 761-766. DOI: 10.1016/j.chemolab.2006.08.015
- Huang, C.Y., J.H. Lo, S.Y. Kuo and M.R. Lyu, 1999. Software Reliability Modeling and Cost Estimation Incorporating Testing-Effort and Efficiency. Proceedings of the 10th International Symposium on Software Reliability Engineering, Nov. 1-4, IEEE Computer Society Washington, DC, USA, pp: 62-62.

- Jones, E., L. Jones and A.J. Rembert, 2001. A simulation based trainer for software reliability modeling. Proceeding of the International Symposium on Software Reliability Engineering, Nov. 27-30, IEEE Xplore Press, USA, pp: 160-165. DOI: 10.1109/ISSRE.2001.989469
- Keller, T. and N. Schneidewind, 1997. Successful application of software reliability engineering for the NASA space shuttle. Proceeding of the 8th International Symposium on Software Reliability Engineering, Nov. 2-5, IEEE Xplore Press, Albuquerque, NM, pp: 112-113. DOI: 10.1109/ISSRE.1997.630854
- Kosko, B., 1998. Fuzzy systems as universal approximators. *Trans. Comput.*, 43: 1329-1333. DOI: 10.1109/12.324566
- Lotfi and J.M. Garibaldi, 2004. *Applications and Science in Soft Computing*. 1st Edn., Springer, Berlin, New York, ISBN: 3540408568, pp: 346.
- Lotfi, A., 2002. *Application of Learning Fuzzy Inference Systems in Electricity Load Forecast*. Nottingham Trent University.
- Lyu, M.R., 1996. *Handbook of Software Reliability Engineering*. 1st Edn., IEEE Computer Society Press, Los Alamitos, Calif, ISBN: 0070394008, pp: 850.
- Minohara, T. and Y. Tohma, 1995. Parameter estimation of hyper-geometric distribution software reliability growth model by genetic algorithms. Proceedings of the 6th International Symposium on Software Reliability Engineering, pp: 324-329. DOI: 10.1109/ISSRE.1995.497673
- Musa, J., 1975. A theory of software reliability and its application. *IEEE Trans. Software Engineering*, 1: 312-327.
- Musa, J., 2004. *Software Reliability Engineering: More Reliable Software, Faster and Cheaper*. 1st Edn., Tata McGraw-Hill Education, Bloomington, ISBN: 0070603197, pp: 608.
- Paramasivam, S. and K.M., 2009. Evaluation of GP model for software reliability. Proceeding of the International Conference on Signal Processing Systems, May 15-17, IEEE Xplore Press, Singapore, pp: 758-761. DOI: 10.1109/ICSPS.2009.104
- Schneidewind, N.F. and T.W. Keller, 1992. Applying reliability models to the space shuttle. *IEEE Trans. Software Engin.*, 9: 28-33. DOI: 10.1109/52.143099
- Sheta, A., 2006. Reliability growth modeling for software fault detection using particle swarm optimization. Proceedings of the IEEE Congress on Evolutionary Computation, (CEC 2006), IEEE Xplore Press, Vancouver, BC, pp: 3071-2078. DOI: 10.1109/CEC.2006.1688697
- Sheta, A., 2007. Parameter estimation of software reliability growth models by particle swarm optimization. *ICGST*, 7: 55-31.
- Sheta, A., E. Oznergiz, M. Abdelrahman and R. Babuska, 2009. Modeling of hot rolling industrial process using fuzzy logic. Proceedings of the ISCA 22nd International Conference on Computer Applications in Industry and Engineering (CAINE-2009), Nov. 4-6, San Francisco, CA, USA, pp: 81-86.
- Teng, X. and H. Pham, 2002. A software-reliability growth model for N-version programming systems. *IEEE Trans. Reliability*, 51: 311-321. DOI: 10.1109/TR.2002.801853
- Wang, L.X. and J.M. Mendel, 1993. Fuzzy adaptive filter: With application to nonlinear channel equalization. *Trans. Fuzzy Syst.*, 1: 161-170. DOI: 10.1109/91.236549
- Wang, L.X., 1992. Fuzzy systems are universal approximators. Proceedings of IEE International Conference on Fuzzy Systems, Mar. 8-12, IEEE Xplore Press, San Diego, CA, USA., pp: 1163-1170. DOI: 10.1109/FUZZY.1992.258721
- White, D.A. and D.A. Sofge, 1992. *Handbook of Intelligent Control, Neural, Fuzzy and Adaptive Approaches*. 1st Edn., Van Nostrand Reinhold, New York, ISBN: 0442308574, pp: 568.
- Xie, M., 2002. Software reliability models-past, present and future. In: *Recent Advances in Reliability Theory: Methodology, Practice and Inference*, Limnios, N. and M. Nikulin, (Eds.), Birkhäuser, Boston, ISBN: 0817641351, pp: 484-521.
- Yager, R.R. and D.P. Filev, 1994. *Essentials of Fuzzy Modeling and Control*. 1st Edn., John Wiley, New York, ISBN: 0471017612, pp: 388.
- Zeng, H. and D. Rine, 2004. A neural network approach for software defects fix effort estimation. George Mason University.