MACHINE LEARNING TECHNIQUES FOR THE EVALUATION OF EFFICIENCY OF THE SOFTWARE RELIABILITY GROWTH MODELS

Omal Sahar[1], Muhammad Ahsan Latif[1] & Muhammad Imran[2]

[1]Department of Computer Science, Agriculture University, Faisalabad, Pakistan
[2]Department of Industrial and Management Engineering Hanyang University Ansan, South Korea

ABSTRACT

Efficiency is a vital factor in the domain of software. Several different approaches had been used for this purpose, but no one completely assessed the efficiency and parameters of the software reliability. In this paper, a genetic alogorithm based approach proposed, for evaluating the efficiency of the SRGMs (Software reliability growth models). Genetic algorithm (GA) is a technique in artificial intelligence for optimization and problem solving with the help of selection, crossover and mutation. The experiments were conducted on four real data sets and four different traditional models.Comparing GA based approach with other approches, i.e., simulated annealing and multiple objective optimizations using genetic algorithm, the results shows that GA based approch provides very efficient results of SRGMs as compared to other selected techniques.

Keywords: *Software Reliability, Growth Models, Genetic Algorithm, Simulated Annealing, Multi-objective Optimization*

INTRODUCTION

Now a day's software reliability is a very vital research area of the software industry. To boost the performance of the programming products and the development process of software, we need to evaluate the reliability.The software reliability can describe the possibility of the software system to achieve its identified functions properly for a long period of time. A vital aspect is to measure the quality of the softwares and calculate their testing time period. Many SRGMS has suggested to assess, to evaluate and to calculate the Software reliability and time period (Liu, Zhang & Zhang, 2015). Software reliability model is the most suitable to combat the numerical problematic evaluation in software reliability. Software reliability evaluation utilizes non-homogeneous Poisson distribution based on the software reliability model. It compels a numerical methodology to select the best model with respect to reasonable aptitude. It is a powerful tool to find out the model parameters which enables to detect the flawed data (Okamura & Dohi, 2015).

Efficiency is the most important part of any software, model or system. When we estimate the efficiency of any product; we check the performance and capability of the product. The important question of this research is: Which technique is efficient to estimate the Growth models? How the efficiency of the model can be measured? Keeping in view all the above mentioned factors, we proposed the genetic algorithm based approch, then estimated and checked the efficiency of different SRGMs models and applied the Genetic Algorithm (GA) based approach and other machine learning approaches. A fitness function was created and applied on a mathematical model (software reliability growth models). The population, selection, crossover and mutation operators, and values by default were used. To evaluate this approach, four real data sets were applied on four traditional SRGMS, Goel–Okumoto GO) Model, Generalized-Goel (GG) model, Inflection S-shaped (INFS) model and Yamada delayed S-shaped (YDSS) Model.

Our suggested technique compares numerical method with other techniques simulating annealing and Multi-objective Optimization using Genetic Algorithm. The outcomes show that the GA is best and consistent for estimating the efficiency of SRGMs. The next step are detailed in the following passage. Related Work section define the models and techniques which have been researched earlier. The next section throws light on the concept of Algorithms and SRGMS. In the following section, Outcomes and discussion are elucidated. In the last one, the conclusions of this paper are explained.

LITERATURE REVIEW

Sheta and Raouf (2016) used optimization technique "gray wolf optimization algorithm (GWO)" to measure parameters of SRGMs and forecasting software reliability throughout testing stage. That processes were very lengthy and time consuming and more efforts required to enhance performance. Hanagal and Bhalerao (2016) suggested the new (GIW) generalized inverse Weibull SRGM and used the decay, the average function value of finite collapses, non-homogeneous Poisson procedure to allow both increase/decrease in the error rate. Compared the new model to the various existing models and perceived that new model achieves better result as compared to another, but required to more increase the success of the model and also plot the mean value function (MVF) according to time/ against time period. Jin and Jin (2016) applied that quantum particle swarm optimization (QPSO) to improve the SRGMs parameters through "S-shaped TEF" and it helped, flexible and used to define the real expenditure pattern more authentically through the software development process. The drawback of this method has not required the supposition of the software failure data.

Fang and Yeh (2015) SRE (software reliability estimation) process that utilized software development engineer to build the CI (confidence Interval) of m(t) of growth models and modification of CI was expected to develop the detection rate of the errors. This technique supported the manager to find out the ideal delivery time period of the software at a suitable confidence stage keeping in view of the SDP. Alweshah, Ahmed and Aldabbas (2015) highly- developed SRGM used the two approaches; "genetic programming evolutionary models" and common "auto-regression simple linear model". These approaches used to forecast the anticipated errors in the existing example basis of earlier calculate errors. This form of the study supports in forecasting the reliability of the software in different programs (applications) and support the supervisor to plan its resources and delivery day of software. (Kim, Lee and Baik (2015) they employed the actual value GA to measure the software reliability growth models and two operators was used "heuristic crossover and non-uniform mutation".

Wang, Wu, Shu and Zhang (2015) proposed the incomplete debug model that considers a log-logistic distribution fault content function and could nab characteristics of variations of error starter or primary rate per error. They used previous fault datasets to certify the performance and working of the suggested model. The CI (confidence interval) to evaluate the sensitivity was also performed. (Li & Liu 2014) Used a new combination technique of SRM, which combine the SVMR (software vector machine regression) model and BP (Back Propagation) model. This technique developed a model through combining the GA (genetic algorithm), PSO (particle swarm optimization) and BP (Back propagation), find out the weight algorithm to select the best parameters of single model and used to forecast the failure data of the software. Finally, experiments were carried out.

Hsu and Huang (2014) combine structure of "SRGMs" that increase the forecasting correctness of the assessment of software reliability and not force to use exact type of the developed system. This model was more elastic and defines the different procedure and action of the software development. Various models were also used as combinational model such as (GG, GO, YDSS and INFS models) but when we increase the combinational models then the assessment of the parameter of SRGM is more difficult. However, more calculations could be completely automatic. More accuracy and forecasting reliability were necessary for the large organization and additional energy and calculation can simply justify and recognized. Amin, Grunske and Colman (2013) recognized that the forecasting technique depend on the time series Autoregressive Integrated Moving Average (ARIMA) technique which helps in better prediction and forecasting of SRGMs issues and software efficiency.

MATERIAL AND METHODS

We have used the three algorithms to measure the efficiency and reliability of the different traditional growth models. We created a fitness function in which define our problem; and collected four data sets to perform experiments. We applied numerical methods "Mean Square Error" for comparison purpose.

Data Collection

DACS datasets obtained from CSIAC. DACS for software provide the data that are in numeric form and data available for different projects such as time sharing system, operating system and military system. These datasets are broadly utilized in the software reliability field in numerous studies. From the Table 1, shows the data sets.

Table 1: Data and Analysis Center of Software (DACS) Datasets

Data Sets	Application Type	Number of Failure	Number of Weeks
SSIB	Operating System	375	95
40	Military	101	50
SS2	Time Sharing System	192	81
SSIA	Operating System	112	55

Estimation of SRGMs (Software Reliability Growth Models)

The SRGMs is the most important skills for measuring reliability of the software, perform a vital role for creating highly optimized software system. It is the mathematical model that tells us how we can repair and enhance the reliability of the software. It is used to find out the level of reliability and various SRGMs although generally used "S-shaped model, Mo model, such model, Go model and JM" (Aggarwal, 2014). Software reliability growth models have two attributes, one is that have norms to specify precise environment and second, they have specific parameters that have physical explanations as well as total number of failure and failure finding rate. SRGMs, the average function value is denoted by m (t) and calculate number of failures in time (0,t). Subsequently the illustration Goel–Okumoto model equality are by way of following:

$$m\ (t) = a\ (1 - e{-}bt),\ a > 0,\ b > 0$$

Parameter "a" represents whole quantity of failure, "b" parameter also represents finding error ratio. Since inexact valuation to the factors could reason of stay issue period and cost swarming for continuing developments, this is significant towards assess these factors correctly.

Table 2: Traditional Software reliability growth models

Serial	**Model**	**Equation**	**Parameters**
1	Goel–Okumoto (GO) Model (Goel & Okumoto,1979)	$a\ (1 - e{-}bt),\ a > 0,\ b > 0$	2
2	Yamada delayed S-shaped (YDSS) Model (Yamada, Ohaba & Osaki,1983)	$m\ (t) = a\ (1 - (1 + bt)\ e{-}bt),\ a > 0,\ b > 0$	2
3	Generalized-Goel (GG) model (Goel,1985)	$m\ (t) = a\ (1 - e{-}btc),\ a > 0,\ b > 0,\ c > 0$	3
4	Inflection S-shaped (INFS) Model (Ohba,1984a)	$m(t) = a(1{-}e{-}bt)\ 1{+}\emptyset e{-}bt,\ \ a > 0,\ b > 0,\ \emptyset > 0$	3

Genetic Algorithm

The genetic algorithm is used for making selection in case of problems having specific constraining factors and to deduce optimized solutions. Natural selection is better example in this regard. GA chooses some members of specific population at a random scale and generate their offsprings. As the process goes on, these offsprings act as a population and the next generation, move towards optimal characteristics or solutions. GA based approach is used to solve complex and mathematical problems and this approach frequently varies population of separate descriptions. Firstly, we create a random population and then selects the individual group in this population. Secondly crossover, combining the pair of the parents in the population and thirdly, mutation changes the genes randomly in individual parents. In each phase, this approach, selecting the different for the parent's population, usage to create the child for the upcoming generation.

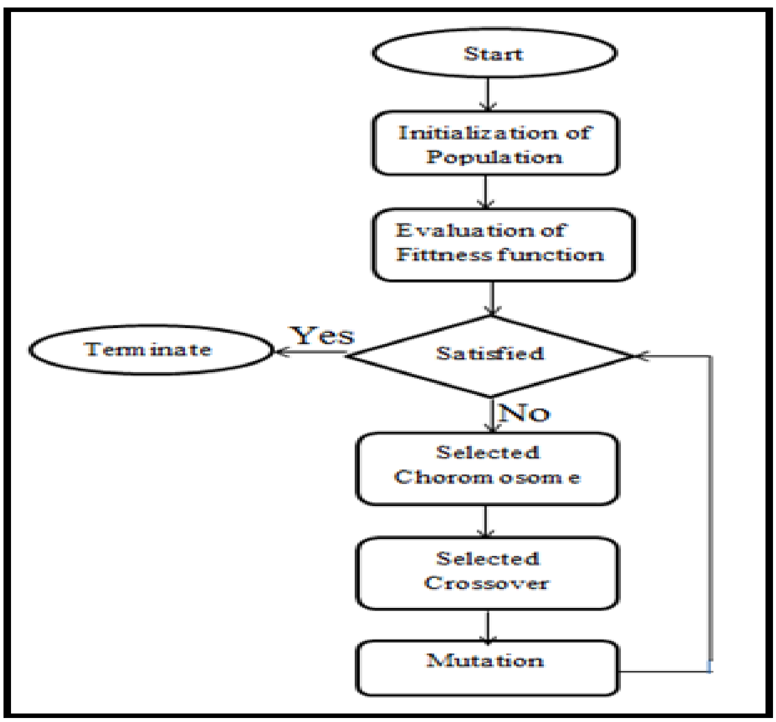

Figure 1: Genetic Algorithm

Multi-objective optimization using Genetic Algorithm

MOGA (multi-objective optimization using genetic algorithm) is the part of the several standards choices of creation or production that is used mathematical problem optimization include more than one objective function to be enhanced concurrently. This algorithm is used in different areas of fields such as science involving economics, engineering and logistics all over optimum choices need to be occupied in occurrence of trade-offs among two or more contradictory objectives. Using the multiple objective genetic algorithm of forecasting the software reliability and also define the results of this technique(Aljahdali & Sheta, 2001).

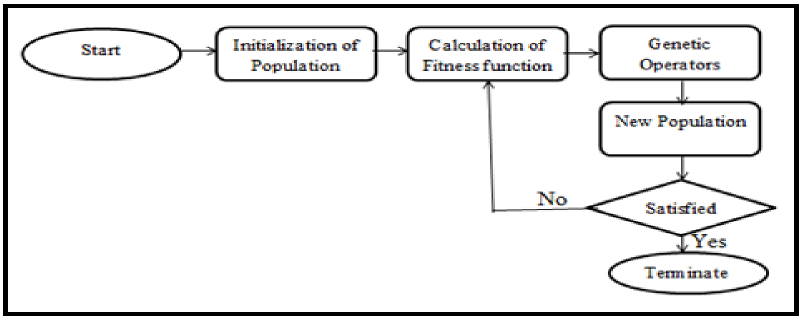

Figure 2: Multi-objective Optimization

Simulated Annealing Algorithm

The SA (Simulated Annealing) is the probabilistic approach for almost the comprehensive optimization of a specified purpose. Specially, this is the meta-heuristic (meta-heuristic is the high level process and structure to determine and generate the best solution of the problem). It is frequently utilized when the space of the search is distinct. The SA understands the slow conserving as a slow reduction in the possibility of accepting inferior solution as it explains the solution. Accept the inferior solution is the important stuff of meta-heuristic that's why it permits for a very wide search for the best solution of the problem. The SA (Simulated Annealing) technique is choosing the parameters of support vector machine Model and used in the software reliability prediction. It is a very good technique in the prediction of reliability of the software (Pai & Hong, 2006).

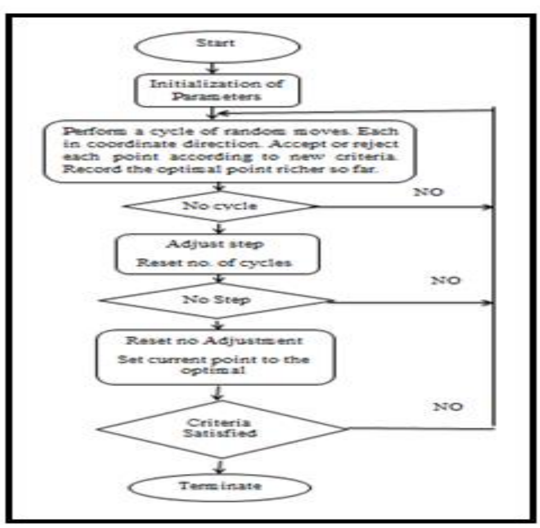

Figure 3: Simulated Annealing

Assessment of Software reliability growth models

Four basic software reliability growth models are used in experiments, GO Model, YDSS Model, GG Model and INFS Model. From the Table 2, show the value of mean function. They have two parameter or factors of GO and YDDS model and also three parameters of GG and INFS models.

Criteria of Comparison

For the criteria of Comparison, the MSE that represents total value of the error among perceive the real faulted and evaluated faulted data from the SRGMs issued.Mean square formula is as below (Hsu & Huang, 2010):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(m_i - m(t_i)\right)^2$$

In which mi are the perceived actual failure and m (ti) assessed failure by software reliability growth models. We compared all the techniques. In this, technique is the best one which MSE value is smaller

RESULT AND DISCUSSION

GA VS other Optimization Techniques

To calculate the efficiency and achievement of proposed GA based approach, we compare outcomes of the genetic algorithm with other optimization techniques SA and MOGA. For the last condition of genetic algorithm, quantity of maximum repetitions is established 200. Every technique applies for every data set, and those the values of median and minimum or results are utilize by this comparison.

Table 3: Comparison of Mean Square Error values of SA, MOGA and GA

Datasets	Model	MSE	SA	MOGA	GA
FC 40	GO Model	Median	68.23	98.24	29.63
		Min	43.71	93.27	28.39
	YDSS Model	Median	40.09	87.37	17.60
		Min	21.01	50.78	15.60
	GG Model	Median	110.54	98.72	9.92
		Min	103.45	56.49	4.11
	INFS Model	Median	114.19	91.81	14.86
		Min	93.91	81.00	11.38
FC SS2	GO Model	Median	353.92	319.23	249.32
		Min	344.75	313.28	246.60
	YDSS Model	Median	293.16	317.98	242.35
		Min	224.92	311.32	239.65
	GG Model	Median	351.76	324.49	220.59
		Min	309.53	324.36	215.10
	INFS Model	Median	344.21	311.49	246.99
		Min	259.92	288.85	240.97

FC SSIA	GO Model	Median	266.70	248.07	168.54
		Min	194.76	239.70	163.38
	YDSS Model	Median	201.83	230.45	250.44
		Min	192.51	221.76	248.41
	GG Model	Median	217.13	226.00	251.29
		Min	183.76	180.96	248.80
	INFS Model	Median	208.00	232.07	161.96
		Min	188.05	227.88	153.63
FC SSIB	GO Model	Median	306.35	267.16	205.40
		Min	296.21	252.31	199.81
	YDSS Model	Median	305.52	271.81	194.87
		Min	303.57	269.39	191.94
	GG Model	Median	315.19	278.36	215.25
		Min	313.90	263.27	205.64
	INFS Model	Median	311.62	267.13	216.05
		Min	303.68	255.68	214.29

From Table 3, Defines the mean square error values among optimization techniques for every model. Compare the three techniques, all the median and minimum values of the genetic algorithm (GA) based approch are smaller than results of other techniques except in the case of the median and minimum values for YDDS and GG Model of SSIA datasets. The median and minimum value of GA based approch in the YDDS and GG Model values of SSIA data sets is greater than other techniques, the difference are not big. The overall results define that the best efficiency of SRGMs and superiority of genetic algorithm based approach.

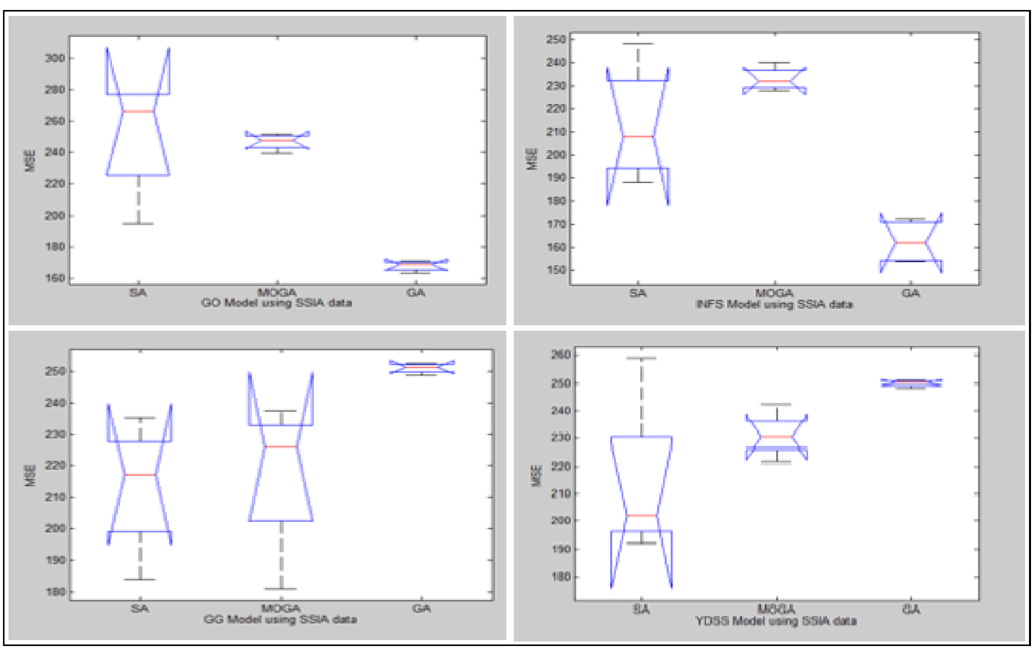

Figure 4: Plots the graphs of the result of SSIA data set

From the point choosing the best model, In the SSIA data set, the YDSS and GG model is choosing to an ideal and optimum model when using the SA and MOGA. However, the GO model and INFS model are choosing the optimum model when using the genetic algorithm (GA) based approch and mean square error value is minimum in both models; When using GA is smaller than that of YDSS and GG model when using SA and MOGA. This means if these SA and MOGA techniques are used in parameter estimation and estimate the efficiency of SRGMS models, the manager of the project could not measure the correct reliability of the present project that could loss the cost and incorrect forecasting of the delivery time or shipping time of the project. Because all data sets and Models, the median and minimum MSE (mean square error) values of GA are smaller than SA and MOGA. Plots the graph of the results of SSIA dataset shown in Fig 4. That define the GA based approache result is better and smaller than other approaches.

In the ANOVA test is to compare the significant difference among the techniques. The "Ho" the null hypothesis and "H1" alternative hypothesis are defined as below:

$$Ho: SA = MOGA = GA$$
$$H1: SA \neq MOGA \neq GA$$

Table 4: Results of ANOVA Test

Data Set	Algorithm	GO Model	YDSS Model	GG Model	INFS Model
		Avg P-value	**Avg P-value**	**Avg P-value**	**Avg P-value**
	SA	74.00	49.95	116.02	115.20
FC40					
	MOGA	**[0.0025]**	**0.0127]**	**[1.89988e-05]**	**[2.89802e-05]**
	GA	394.99	80.82	90.25	91.88
		30.37	17.98	9.14	21.31
	SA	354.12	287.65	343.98	327.81
FCSS2					
	MOGA	**[2.4673e-09]**	**[0.015]**	**[1.9821e-06]**	**[0.0086]**
	GA	318.64	317.83	325.07	309.66
		250.51	243.18	220.89	246.71
	SA	251.36	213.74	213.35	213.10
FCSSIA					
	MOGA	**[0.0009]**	**[0.057]**	**[0.0417]**	**[0.0005]**
	GA	246.98	231.27	217.64	182.98
		167.76	250.1	250.96	162.48
	SA	305.70	306.69	314.99	310.39
FCSSIB					
	MOGA	**[5.89549e-08]**	**[2.12193e-09]**	**[4.77235e-09]**	**[4.05186e-07]**
	GA	264.49	272.82	275.03	267.93
		205.26	197.90	213.66	218.01

From the Table 4, the average values of MSE and ANOVA test results are described. For the FC40 dataset, the p values of GO Model are 0.0025, 0.0127 for YDSS model, 1.89988e-05 for GG Model and 2.89802e-05 for INFS model. These values of P are smaller than the level of significant 0.05. So, alternative hypothesis are accepted. In addition, in all the data sets the values of p are lesser than significant values. So that we discard the null assumption and there is a discrete alteration among the three technique SA, MOGA, GA.

CONCLUSION

In this paper, we suggested a very efficient and suitable technique to calculate the efficiency of the SRGMs using GA based approch. We proposed the genetic algorithm based approach to apply to the evaluation of the parameters of the SRGMs. Three operators were used in GA based approach i.e., Selection, crossover and mutation. To evaluate the performance of suggested technique, experiments were performed in the four real data sets and four traditional models; and also compared the efficiency and performance of our technique to other optimization techniques. The experimental outcomes show that GA based approach could determine the best solution more efficiently and frequently than other meta-heuristics or optimization techniques like Simulating Annealing and multi-objective optimization. In the future, we will carry out the experimental studies with other meta- heuristic techniques and optimization techniques.

Reference

Aggarwal, G., & Gupta, V. K. (2014). Software reliability growth model. I*nternational Journal of Advanced Research in Computer Science and Software Engineering,* 4 (1): 475-479.

Aljahdali, S. H., Sheta, A., & Rine, D. (2001). Prediction of software reliability: A comparison between regression and neural network non-parametric models. In *Computer Systems and Applications, ACS/IEEE International Conference*, 470-473.

Alweshah, M., Ahmed, W., & Aldabbas, H. (2015). Evolution of Software Reliability Growth Models: A Comparison of Auto-Regression and Genetic Programming Models. *International Journal of Computer Applications*, 125(3).

Amin, A., Grunske, L., & Colman, A. (2013). An approach to software reliability prediction based on time series modeling. *Journal of Systems and Software*, 86 (7): 1923-1932.

Fang, C. C., & Yeh, C. W. (2016). Effective confidence interval estimation of fault-detection process of software reliability growth models. *International Journal of Systems Science*, 47(12): 2878-2892.

Goel,A.L. (1985).Software reliability models: assumptions limitations and applicability. *IEEE Transaction Software Engneering*, 11:1411–1423.

Goel, A. L., & Okumoto, K. (1979). Time-dependent error detection rate model for software reliability and other performance measures. IEEE Transaction of Reliability, 28:206–211.

Hanagal, D. D., & Bhalerao, N. N. (2016). Modeling and Statistical Inference on Generalized Inverse Weibull Software Reliability Growth Model. *Journal of the Indian Society for Probability and Statistics*, 17(2):145-160.

Hsu, C. J., & Huang, C. Y. (2010). A Study on the applicability of modified genetic algorithms for the parameter estimation of software reliability modeling. In *2010 IEEE 34th Annual Computer Software and Applications Conference*, 531-540.

Hsu, C. J., & Huang, C. Y. (2014). Optimal Weighted Combinational Models for Software Reliability Estimation and Analysis. *IEEE Transactions on Reliability*, 63(3), 731–749.

Jin, C., & Jin, S. W. (2016). Parameter optimization of software reliability growth model with S-shaped testing-effort function using improved swarm intelligent optimization. *Applied Soft Computing*, 40, 283-291.

Kim, T., Lee, K., & Baik, J. (2015). An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm. *Journal of Systems and Software*, 102, 134-144..

Li, K., & Liu, H. (2014). New combination approach of software reliability model based on time-series data. *Journal of Computer Applications,* S2.

Liu, G., Zhang, D., & Zhang, T. (2015). Software Reliability Forecasting: Singular Spectrum Analysis and ARIMA Hybrid Model. In *Theoretical Aspects of Software Engineering (TASE), International Symposium,* 111-118.

Ohba, M. (1984a). Inflection S shaped software reliability growth model ,Stochastic Models in Reliability Theory. *Springer,* 144–162.

Okamura, H., & Dohi, T. (2015). Phase-type software reliability model: parameter estimation algorithms with grouped data. *Annals of Operations Research*, 1-32.

Pai, P. F., & Hong, W. C. (2006). Software reliability forecasting by support vector machines with simulated annealing algorithms. *Journal of Systems and Software*, 79(6):747-755.

Sheta, A. F., & Raouf, A. (2016). Estimating the Parameters of Software Reliability Growth Models Using the Grey Wolf Optimization Algorithm. *International Journal of Advanced Computer Science & Applications*, 1(7):499-505.

Wang, J., Wu, Z., Shu, Y., & Zhang, Z. (2015). An imperfect software debugging model considering log-logistic distribution fault content function. *Journal of Systems and Software*, 100:167-181.

Yamada, S., Ohba, M., & Osaki, S. (1983). S-shaped reliability growth modeling for software error detection, *IEEE Transaction of Reliability,* 32:475–484.