International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Application of Bayesian Regularization Algorithm for Evaluation of Performance Software Complexity Prediction Model Based on Requirement

Wartika, Ford Lumban Gaol , Ariadi Nugroho , Bahtiar Shaleh Abbas

Abstract: ***Model performance evaluation is a method and process of evaluating the model that has been built. The model that will be evaluated is software complexity prediction model based on requirement. This model allows measuring software complexity before actual design and implementation. The experiment used three datasets namely training dataset, validation data set , and test dataset. For performance evaluation using Mean squared error. Mean squared error is very good at giving a description of how consistent the model is built. By minimizing the value of mean squared error, it means minimizing model variants. Models that have small variants are able to give relatively more consistent results for all input data compared to models with large variants. This research proposes the application of the Bayesian regularization algorithm for evaluating the performance of software complexity prediction model based on requirement. With this research it is expected to know how much the performance of the model that has been built.***

Keywords: Algorithm, prediction, software, complexity

I. INTRODUCTION

Quality assurance, customer satisfaction and reliable products are the immediate needs of the software industry today [1]. [2] in his research wrote that software development projects have high uncertainty. . [3] states that the six main causes of software project failures are incomplete requirements, less user involvement, lack of resources, excessive expectations, lack of support from the executive, changes in requirements and specifications.

Along with the increasing implementation of information technology in various fields, then the complexity of the software also increases, The increased complexity is related to various increased requirements. [4] In his research stated that complexity is the main driver of cost, reliability, software performance and one of the most important factors that can affect quality.

Manuscript published on 30 September 2019
* Correspondence Author
Wartika*, Program Doctor Of Computer Science, Binus Graduate Program, Binus University
Ford Lumban Gaol , Reseacher In The Field Of Computer Science, Lecturer At Program Doctor Of Computer Science, Binus University
Ariadi Nugroho, Practitioners In The Field Of Information Technology, Work At BTPN Bank, Reseacher In The Field Of Computer Science
Bahtiar Saleh Abbas, Reseacher In The Field Of Computer Science, Lecturer At Program Doctor Of Computer Science, Binus University

Measuring and controlling the complexity of software is an important aspect in every software development paradigm. So far, most research has tried to identify and measure complexity in the design and coding phases. However, in this phase to control complexity is too late [4]. To develop research [5] and [4] , then in this study a proposed software complexity prediction model based on requirement.
Then the model is evaluated for its performance by using Bayesian regularization algorithm. With this evaluation it is expected to know how much the performance of the model that has been built.

II. LITERATURE REVIEW

A. Software Complexity

According to [6] complexity is one of the main problems in software engineering. Software complexity is a term that includes various properties of a piece of software that all affect internal interactions. [7] Propose that software complexity refers to characteristics software that affects the level of resources used by someone in carrying out the tasks assigned. According to [8] software complexity is the level of difficulty in analyzing, designing, modifying, maintaining, and testing software.

B. Requirement

Requirements are a description of the services that must be provided by the software system and the limitations on which the services must operate . The types of requirements consist of : (1) User requirements, are services provided by the system and operational constraints are written to the customer. (2) System requirements, are structured documents that describe detailed descriptions of system services, written as contracts between clients and contractors. (3) Software specification, is a detailed software description that can serve as a basis for design or implementation, written for developers.

C. Bayesian Regularization Backpropagation

Bayesian regularization is an artificial neural network training algorithm which corrects the weight and refraction values based on the Levenberg-Marquardt optimization [9]. This algorithm minimizes the combination of error squares and weights, then determine the correct combination so as to produce a good network [9]. This process is called Bayesian regularization.

Bayesian regularization neural network introduce network weights into the objective function of training. The objective function of the training is notated as follows [9]

$$F(\omega) = \alpha E\omega + \beta ED \quad (1)$$

Where $E\omega$ is the sum of squares of the network weight, and ED the sum of squares of network errors. The values α and β are parameters of the objective function. In the Bayesian process flow, network weights are seen as random variables, then the previous distribution of network weights and training is considered a Gaussian distribution follows [9]. The following applies the Bayesian rule to optimize the objective function parameters α and β.

$$P(\alpha, \beta|D, M) = \frac{P(D|\alpha, \beta, M)P(\alpha, \beta|M)}{P(D|M)} \quad (2)$$

The principle of Bayesian Regularization artificial neural network training algorithm is shown in Figure 1.

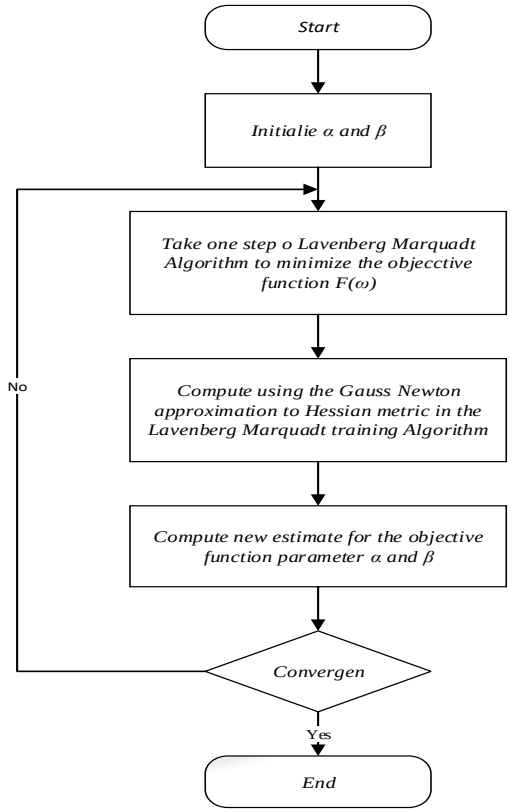

Fig. 1. Principles of Bayesian Regularization Algorithm [9]

III. RESEARCH METHOD

This section contains the methods used in research according to the type of research. According to the method, this research is an experimental research that is research that seeks to find the influence of certain variables on other variables under controlled conditions. According to the type of data and the analysis of this study, including quantitative research, namely research whose data is quantitative data so that the data analysis uses quantitative analysis (inference).

According to the level of explanation (explanation) this study includes descriptive research that is research that collects data to test hypotheses or answer research questions regarding the final status of research subjects. Descriptive research seeks to obtain a complete and accurate description of a situation.

The steps in this research method are preceded by concepts and theories which are an attempt to obtain answers to the problem formulation. Theoretical arguments are obtained from theoretical studies or relevant research results. This is done as a guide or direction for conducting research. The second step is to set the independent variable and the dependent variable. The third step is to establish relationships between variables. Based on the relationship between variables that will be used as a basis for making a hypothesis then a regression analysis is performed and model performance analysis. the final step is to draw conclusions.

The steps in the research method can be seen in Figure 2 .

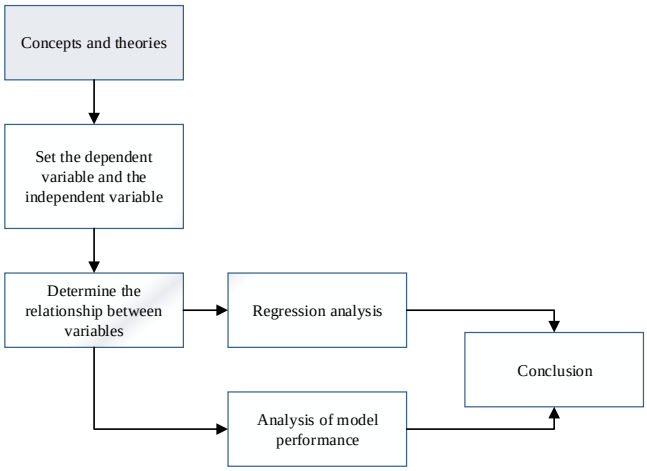

Fig. 2. The steps in the research method

IV. DISCUSSION

The variables used in this study refer to research [10] and [5]. The following are the independent variables and the dependent variables used in the study.

Table-I : Independent Variable and Dependent Variable

Variable Independent		
Description	Data Types	Data Value
Basic factor	Nominal	1 ≤ value ≤ 50
Requirement factor	Nominal	1 ≤ value ≤ 50
User terminal factor	Nominal	1 ≤ value ≤ 50
Function point	Nominal	1 ≤ value ≤ 50
Variable Dependent		
Software complexity	nominal	

The independent variable is used as the input variable, and the dependent variable is used as the target (output) variable. After determining the dependent variable and independent variables then do data collection. The data used consisted of 44 data from different companies.

The following will describe the effect of basic factors, requirements factors, user terminal factors, and function points on software complexity using regression analysis.

Table-II : Variabel Entered/Removed

Variables Entered/Removed[a]			
Model	Variables Entered	Variables Removed	Method
1	VAR4, VAR2, VAR3, VAR1[b]	.	Enter
a. Dependent Variable: VAR5			
b. All requested variables entered.			

The output variables enteres / removed table above provides information about the research variables and the methods used in the regression analysis. The independent variables used in this analysis are the basic factors, requirements factors, user terminal factors and function points. While the dependent variable is software complexity. Regression analysis using the enter method. There are no variables removed so the variables removed column does not have a number or is empty.

Table-III : Model Summary

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.928[a]	.860	.846	6444.84289
a. Predictors: (Constant), VAR4, VAR2, VAR3, VAR1				

The summary table provides information about the value of the coefficient of determination, namely the contribution or contribution of the influence of the variable basic factor, requirement factor, user terminal factor and function point simultaneously (together) to software complexity.

The meaning of the coefficient of determination (R square) in multiple linear regression analysis or symbolized by R2 which is meaningful as contribution of influence given the independent variable on the dependent variable, or in other words, the value of the coefficient of determination or R square is useful to predict and see how much the contribution of the influence of variable x given simultaneously (together) to the variable y.

Based on the output model summary table above, it is known that the coefficient of determination or R square is 0.860. R square value of 0.860 is derived from the squaring of the correlation coefficient value of R. which is 0.928 x 0.928 = 0.860.

The magnitude of the coefficient of determination (R square) is 0.860 or equal to 86.0%. This number implies that the variable basic factors, requirements factors, user terminal factors, and function points simultaneously (together) affect the software complexity of 86.0%. While the rest (100% - 86.0% = 14%) is influenced by other variables outside this regression equation or variables not examined.

Table-IV : Anova

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	9985906821.000	4	2496476705.000	60.104	.000[b]
	Residual	1619903994.000	39	41535999.850		
	Total	11605810820.000	43			
a. Dependent Variable: VAR5						
b. Predictors: (Constant), VAR4, VAR2, VAR3, VAR1						

Based on the significance value (sig) and anova output, it is known that the sig value is 0,000. Because the sig value of 0.000 <0.05, according to the basis of decision making in the F test, it can be concluded that the hypothesis is accepted or in other words the variable basic factors, requirements factors, user terminal factors, and function points simultaneously (together) affect the software complexity.

Based on the output table, the calculated F value is 60,104. Because F count 60.104> F table 2.61, as a basis for decision making in the F test it can be concluded that the hypothesis is accepted.

Table-V : Coeficients

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-27892.364	3278.111		-8.509	.000
	VAR1	1630.430	182.618	.673	8.928	.000
	VAR2	1887.962	296.211	.388	6.374	.000
	VAR3	2795.192	654.645	.276	4.270	.000
	VAR4	19.638	29.060	.053	.676	.503
a. Dependent Variable: VAR5						

VAR1 = basic factor = X_1
VAR2 = requirement factor = X_2
VAR3 = user terminal factor = X_3
VAR4 = function point = X_4
VAR5 = software complexity = Y

The resulting regression formula is:

$$Y = -27892.364 + 1630.430X_1 + 1887.962X_2 + 2795.192X_3 + 19.638\ X_4 \quad (3)$$

The experiment used three datasets namely training dataset, validation data set , and test dataset. Training dataset is a part of a dataset that is trained to make predictions or other machine learning algorithms according to their respective goals.

We provide instructions through an algorithm so that trained machines can find their own correlation. . The training dataset is a collection of examples used for learning, i.e. to fit parameters.

The validation dataset is an example dataset used to set hyperparameters (eg architecture) of the classifier. In artificial neural networks, hyperparameter for example, is the number of hidden units.

The test dataset is the part of the dataset that we test to see its accuracy, or in other words to see its performance. The test dataset is a dataset that does not depend on the training dataset, but follows the same probability distribution as the training dataset. If the model matches the training dataset also matches the test dataset well, minimal overfitting has occurred.

To evaluate the performance of the model built in this study the Bayesian regularization prediction algorithm is used. Regularization has the role of increasing the generalization process by limiting the size of the weight of a network.

With regularization, a large network simplified must be able to represent the actual function. In the classic backpropagation algorithm aims to minimize the function F = Ed, where [9] :

$$\text{Ed} = \sum_{i=1}^{n}(ti - ai)^2 \qquad (4)$$

In this case n is the number of inputs in the training set, ti is the target value in data to i and αi is output for the data to i which is obtained as a neural network response. The regularization method changes the performance of function errors by adding a standard deviation of weights and biases, where [9] namely:

$$F = \beta E_d + \alpha E_w \qquad (5)$$

αβ is a reguralisation parameter, and Ew is defined as:

$$E_w = \frac{1}{n}\sum_{i=1}^{n}(wi)^2 \qquad (6)$$

wi is a weight or threshold. Using the equation to change the error performance function allows the network to get the lowest weight and threshold , but cannot determine the weight and effective network threshold. Conventional methods are often difficult to determine parameter size. [11] propose networks that can adjust the size of adaptive parameters using the Bayesian theory framework, and allow optimal performance to be achieved.

The formula for determining the regularization parameter is where [9] namely:
:

$$\alpha = \frac{\gamma}{2Ew} \qquad (7)$$

$$\beta = \frac{n-\gamma}{2ED} \qquad (8)$$

The following will describe the experiments carried out. In this experiment using the Bayesian Regularization prediction algorithm with 4 neurons in the input layer, hidden layers used 1 to 5 hidden layers, and 1 neuron in the output layer. Experimental input is the basic factor, requirements factor, user terminal factor, and function point, while the output is software complexity.

Random data distribution. Training using scaled conjugate gradient means training using supervised learning algorithms for feedforward neural networks, and is a member of the conjugate gradient method class.

In this experiment the performance uses Mean Squared Error (MSE). Mean Squared Error (MSE) is a method for evaluating forecasting methods. Each error or remainder is squared. Then added up and added to the number of observations.

This approach manages large forecasting errors because they are squared. If the error between one iteration and the next iteration is small enough, the calculation process stops. MSE can be defined as [12] :

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(fi - yi)^2 \qquad (9)$$

Where fi is the value of the forecast result, yi is the true value, and n is the amount of data.

A smaller error value will be smaller and a large error value will be even greater. MSE can also be analogous to the variant plus the quadratic bias of the model. [12]

$$\text{MSE}(\hat{\theta}) = \text{E}[(\hat{\theta} - \theta)^2] = \text{var}(\hat{\theta}) + (\text{bias}(\hat{\theta}, \theta)^2) \qquad (10)$$

Where is $\hat{\theta}$ the model being evaluated and θ is the actual model. If there is no bias between the model built and the actual model (unbiased model), then MSE is commensurate with the variant of the model. MSE is very good at giving a description of how consistent the model is built. minimizing the value of MSE, it means minimizing model variants. Models that have small variants are able to provide relatively more consistent results for all input data compared to models with large variants (large MSE).

In this experiment, one of the training methods, backpropagation, uses the term epoch because when one iteration is done with back propagation. Epoch is when the entire dataset has been through the training process on neural netwok until it is returned to the beginning for one round, because one epoch is too large to be fed (feeding) into a computer so we need to divide it into small units (batches).

Batch Size is the number of sample data distributed to a neural network. Iterations are the number of batches needed to complete one epoch. Gradient is the slope taken from the current position compared to the previous epoch and iteration. If it's still large the difference means it's still tilted. If it's still tilted it still moves down, while if it's not so tilted, , and although it hasn't stopped, in computing it's been stopped for efficiency.

Gradient or slope of a line is the amount of angle formed by the line with respect to the horizontal line. The magnitude of the slope starts from infinite negative and reaches to infinity. Sum squared is a statistical technique used in regression analysis to determine the dispersion of data points.

Sum squared is used as a mathematical way to find the most suitable (least variable) function of the data. For a set of x of n items. Sum square is defined as the equation where [9]

Sum of squared = $\sum_{i=0}^{n}(Xi - \bar{X})^2$ (11)

Where :

Xi = item i from set

$\bar{x}$ = mean of all items in the set

$(Xi - \bar{x})$ = deviation of each item in the mean

Here are the experimental data using the Bayesian regularization prediction algorithm :

Table-VI : Experimental Results Data

Hidden layer	*Epoch*	*Time*	*Performance*	*Gradient*
1	62	0:00:08	9.89e-23	4.67e-07
2	51	0:00:15	6.27e-24	6.93e-08
3	65	0:00:15	9.713-18	0.000293
4	79	0:00:22	1.85e-17	8.69e-05
5	30	0:00:11	1.30e-23	2.00e-07

From the table-VI the performance with the smallest MSE is to use hidden layer 2 with the following details:

Table-VII : Experimental Data With The Smallest MSE

Data division	**Random**	
Training	Scaled conjugate gradient	
Performance	Mean Squared Error	
Calculation	MEX	
Epoch	51 iterasi	0 - 1000
Time	0:00:15	
Performance	6.27e-24	3.26e+08 - 0.00
Gradient	6.93e-08	8.68e+08-1.00e-07
Mu :	5.00e+08	0.00500 1.00e+10
Effective #param	222	222 – 0.00
Sum squared param	84.1	52.01 – 0.00
Validation Checks	0	

The experiment randomly divided 94 samples with the following details:

Table-VIII : Experimental Data With The Smallest MSE

Data	**Persentage**	**Number Of Samples**	**MSE**	**R**
Training	70%	66	6.57850e24	9.99999e-1
Validation	15 %	14	0.00000e-0	0.00000e-0
Testing	15 %	14	5.98199e-24	0

Mean squared error is the difference in average squared between output and target. Lower value is better. Zero means there are no mistakes.

Regression value R measures the correlation between output and target. A value of R close to 1 means a close relationship. Value R = 0 means a random relationship.

Using the Bayesian regularization prediction algorithm shows the best validation performance of 6.2657e-24 on Epoch 51 as shown in the following figure 2.

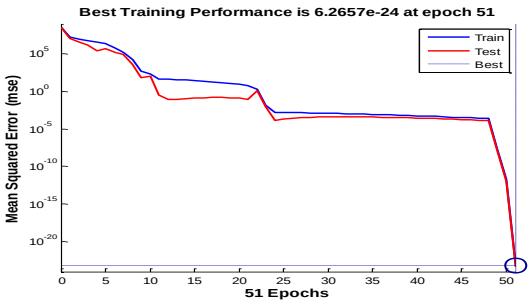

Fig. 3. Performance Algoritma Bayesian Regularization

The resulting regression function is as follows:

Output = 1*target + (-1,6 e-12) (12)

R=1

V. CONCLUSION

From the results of regression analysis and model evaluation using the Bayesian regularization algorithm, conclusions are drawn that requirements consisting of basic factor variables, requirement factors, user terminal factors, and function points can be used as variables to predict software complexity, where these variables 86.0% simultaneously (together) affect the software complexity, while 14% is influenced by other variables outside this regression equation or variables not examined.

The resulting complexity model prediction software is: Y = -27892.364+ 1630.430X_1 + 1887.962X_2 + 2795.192X_3 + 19.638 X_4.

The resulting model is validated using mean squared error. By using the Bayesian regularization prediction algorithm, it shows the best validation performance of 6.2657e-24 on Epoch 51.

REFERENCE

1. X. Liu, “Object-oriented software metrics,” 1999.
2. K. Na, J. T. Simpson, X. Li, T. Singh, and K. Kim, “Software development risk and project performance measurement : Evidence in Korea,” vol. 80, pp. 596–605, 2007.
3. B. Boehm, C. Abts, and A. Brown, “Cost estimation with COCOMO II,” *ed Up. Saddle …*, 2000.
4. G. Keshavarz, N. Modiri, and M. Pedram, “Metric for Early Measurement of Software Complexity,” *Interfaces (Providence).*, 2011.
5. B. Arthi, A. G. Selvarani, and A. Sathya, “Software Complexity Measure from Software Requirements Document and Cost Driver Factors,” vol. 24, pp. 6–12, 2016.
6. A. Jakhar and K. Rajnish, “Measurement Of Complexity And Comprehension Of A Program Through A Cognitive Approach,” *Int. J. Eng. B …*, 2015.
7. B. Sellers, “Ojbect-oriented metrics. measures of complexity,” 1996.
8. S. Nystedt and C. Sandros, “Software Complexity and Project Performance,” *Univ. Gothenbg.*, 1999.
9. F. Burden and D. Winkler, “Bayesian Regularization of Neural Networks,” 2008, pp. 23–42.
10. G. Keshavarz, “Metric for Early Measurement of Software Complexity,” vol. 3, no. 6, pp. 2482–2490, 2011.

11. Z. Yang, L. Xie, C. Z.-I. T. on Signal, and undefined 2013, "Variational Bayesian algorithm for quantized compressed sensing," *ieeexplore.ieee.org*.
12. B. Murphy, T. Gent, and P. Street, "Measuring System and Software Reliability using an Automated Data Collection Process 2 . Changes in System Reliability 2 . 1 Changes in Product Reliability," pp. 1–12, 1980.

AUTHOR PROFILE

Wartika, Program Doctor Of Computer Science, Binus Graduate Program, Binus University

Ford Lumban Gaol , Reseacher In The Field Of Computer Science, Lecturer At Program Doctor Of Computer Science, Binus University

Ariadi Nugroho, Practitioners In The Field Of Information Technology, Work At BTPN Bank, Reseacher In The Field Of Computer Science

Bahtiar Saleh Abbas, Reseacher In The Field Of Computer Science, Lecturer At Program Doctor Of Computer Science, Binus University