APPLICATION STUDY OF ALGORITHM C4.5, MLP, AND NAIVE BAYES FOR SOFTWARE DEFECT PREDICTION

Aditiya Hermawan

Informatics Engineering, Universitas Buddhi Dharma, Indonesia
aditiya.hermawan@ubd.ac.id

Abstract

Companies and institutions in various fields require software to help their business processes in order to run fast, precise, effective and efficient. Surely the software used must have good quality standards for the purpose of a company or institution can be met. Thus the necessary software that does not have a fault / error (defect). There have been many other researchers doing modeling for software development, which will be used for the development of fatherly making software. The proposed model is proposed is expected to produce quality software is without defect. Of several previous studies there has been no accurate model for prediction of Software Defect due to the number of variables are many and varied which resulted in less accurate predictions. Model is good enough to do software defect prediction is C4.5, MLP and Naïve Bayes. Several other researchers also tried to improve the accuracy of existing by variable selection are used. The research on genetic algorithm will be applied to the selection of variables in methods of C4.5, MLP and Naïve Bayes using data from the NASA dataset. After that will be tested with ROC curves and Confusion Matrix to find which model produces the highest level of accuracy in the prediction of software defects. Accuracy results obtained prove that the Naïve Bayes has a higher degree of accuracy than C4.5 and MLP. Naïve Bayes with Genetic Algorithm produces an accuracy percentage of 88.25% and the value of AUC (Area Under Curve) of 0.772. Thus Naïve Bayes algorithm optimized with a genetic algorithm to predict the Software Defect better.

Keywords: *Software Defect, C4.5, MLP, Naive Bayes, Genetic Algorithm*

1 INTRODUCTION

Almost all companies or institutions in various fields need software to help their business processes run fast, precisely, effectively and efficiently. The software used must have good quality standards for the purpose of the company or institution can be fulfilled. The demand for quality software to support the performance of companies or institutions increases from year to year [10]. The attributes of software quality are reliability, functionality, fault proneness, re usability, and comprehensibility [9] [10]. Among the attributes of software quality, fault proneness is an important issue, as it can be used to assess the final quality of software, predict customer standards and satisfaction [10]. Fault proneness is the probability of error in software. Fault proneness is one of the attributes in assessing the software of concern as it can be a tool of error (defect) [26]. Companies or institutions definitely need software that has little or no errors in order to invest in the field of information technology is not a waste. The number of defects in the software can be used to measure the quality of software developers and manage the software process [32].

From some previous research, there are several methods on some popular datasets for prediction software defect. Some do attribute selection or attribute classification, then compile the method used and get the results of the evaluation. Of all the models that have been studied, there is no model that produces very precise accuracy on the prediction of Software Defect although with different process models, there is still no model that can be a reference for software defect prediction. The datasets they use are not exactly the same. Research by Menzies team and colleagues in 2007 used the NASA dataset from Promise Repository, while research by Stefan Lessmann and colleagues, Khoshgoftaar and colleagues, Qinbao Song and colleagues used NASA datasets from MDP Repository [31]. The use of datasets originating from different repositories can also produce different accuracy.

Based on the results of previous research to predict software defects, there are some fairly accurate data mining algorithms for some datasets, which will be role models in this study. The algorithm used is C4.5, Multilayer Perceptron (MLP) and Naive Bayes. These three algorithms were chosen because of some previous studies having a fairly high degree of accuracy on some datasets [20]. These three algorithms will be searched for accuracy and comparisons to find which algorithm is best for predicting software defects.

To improve the accuracy in this study used the selection of variables or often heard with Feature Selection. One of the most commonly used methods is the Genetic Algorithm (GA) method. The Genetic Algorithm process incorporates a heuristic evaluation methodology [1]. The purpose of selecting variables is to identify the equally important variables in the dataset, then discard other variables whose value is irrelevant and redundant [23]. With the selection of variables make the method faster and more effective because it does not use irrelevant and excessive variables. Moreover, results with variable selection allow to

improve accuracy in data classification which in this case is expected to increase the accuracy of algorithm C4.5, Multilayer Perceptron (MLP) and Naive Bayes in predicting Software Defect.

2 METHODOLOGY AND DISCUSSION

The methodology used in this study using CRISP-DM model (Cross Standard Industries Process for Data Mining), in this method there are 6 stages [19]:

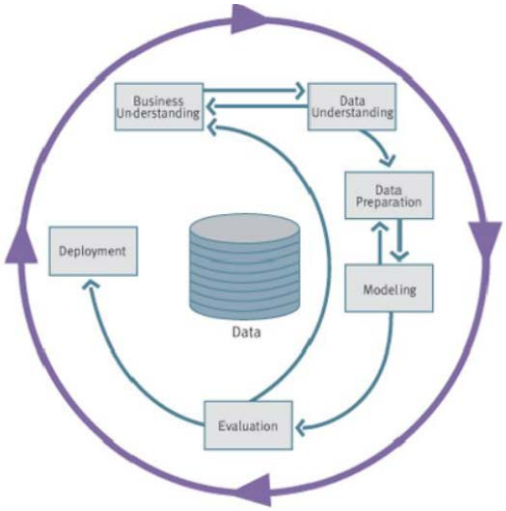

Figure 1. CRISP-DM (Cross Standard Industries Process for Data Mining) [19]

1. Business/Research Understanding Phase

NASA The currently available dataset comes from several Repository (MDP and PROMISE), this research used datasets derived from MDP Repository that have been fixed by another researcher named Martin Shepherd. The dataset has been fixed by deleting data that is null or of no value. Therefore, researchers used the dataset to conduct research in determining the proposed model in the Software Defect prediction.

2. Data Understanding Phase

The dataset obtained is 13 datasets, namely: CM1, JM1, KC1, KC2, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4, and PC5. The dataset fixed by Martin Shepherd, with the following specifications:

Table 1. NASA Dataset

	Programming Language	Number of variables	Number of Data	Number Of Defect	Defect (%)
CM1	C	38	344	42	12.21
JM1	C	22	9593	1759	18.34
KC1	Java	22	2096	325	15.51
KC3	Java	40	200	36	18.00
MC1	C++	39	9277	68	0.73
MC2	C++	40	127	44	34.65
MW1	C	38	264	27	10.23
PC1	C	38	759	61	8.04
PC2	C	37	1585	16	1.01
PC3	C	38	1125	140	12.44
PC4	C	38	1399	178	12.72
PC5	C	39	17001	503	2.96

3. Data Preparation Phase

NASA The dataset obtained by 13, in this study used 12 datasets (without KC2). One of the datasets is not used because the dataset is not relevant for this research.

To do this research, used data that has value in each variable. Data on any dataset that has no value will be removed and not used. This is done so that the data becomes clean and can produce accuracy with the correct value.

Data on datasets that do not have values on any of the variables will be deleted or not used, even if other variables have values. Suppose, a data with 22 variables and one of the variables is null then the data is not used. Data whose variables are null will also not be used.

The amount of data in the dataset will decrease in number from the beginning of the obtained dataset. The data in the dataset will not increase in number because what is done is a reduction by deleting the data.

4. Modeling Phase

The model that will be proposed in this research is to do 5 stages:

1. Dataset separation into training data and data testing
2. Selection of variables with variable selection on training data
3. Using the method / algorithm in the training data that has been selected variables
4. Implementation of methods / algorithms on data testing
5. Compare the results of accuracy and performance results

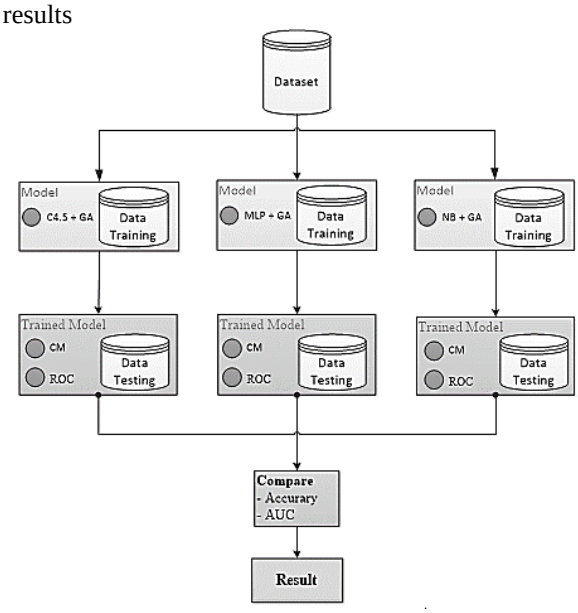

Figure 2 Comparison Proses with optimization techniques

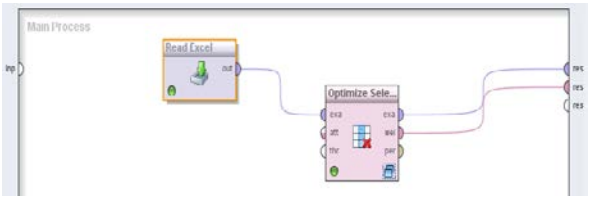

Figure 3. Main Proses - Rapid Miner 5

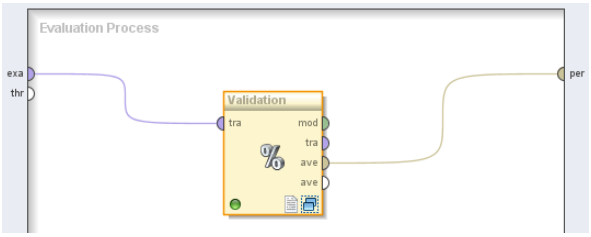

Figure 4. Cross Validation - Rapid Miner 5

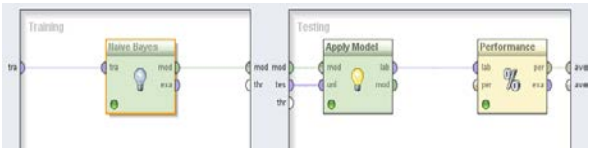

Figure 5. Cross Validation Detail - Rapid Miner 5

5. *Evaluation Phase*

In this phase testing of models aimed to obtain the most accurate model. Evaluation and validation is done by using Confusion Matrix method and ROC curve (Receiver Operating Characteristic).

The dataset is tested using C4.5 method with Genetic Algorithm optimization.

Results obtained from the method C4.5 + GA, i.e. with the value of AUC: 0.504; TP: 1; TN: 296; FP: 41; And FN: 6. Of the 4 values for the confusion matrix generate the following table:

Table 2 Confusion Matrix for CM1 Dataset with C4.5 Algorithm

CM1			
	N	Y	Precision
N	296	41	87.83%
Y	6	1	14.29%
Recall	98.01%	2.38%	

The accuracy value of the confusion matrix is as follows:

$$\text{accuracy} \frac{(TN+TP)}{(TN+FN+TP+FP)} = \frac{(296+1)}{(296+6+11+41)} = 0{,}8634 = \mathbf{86,34}\%$$

ROC Curve for CM1 dataset :

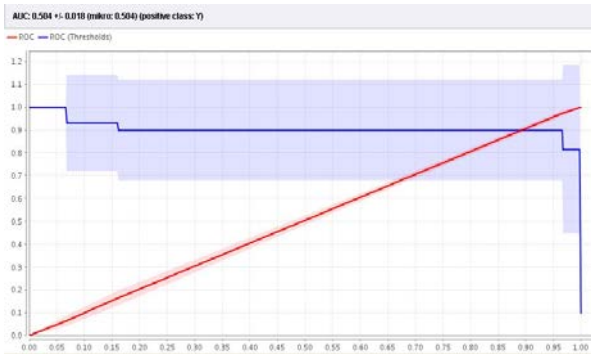

Figure 6. AUC for CM1 Dataset with C4.5 Algorithm

The dataset is tested using Multi Layer Perceptron method with Genetic Algorithm optimization.

Results obtained from Multi Layer Perceptron + GA method, i.e. with the value of AUC: 0.712; TP: 8; TN: 295; FP: 34; And FN: 7. Of the 4 values for the confusion matrix generate the following table:

Table 3 Confusion Matrix for CM1 Dataset with Multi Layer Percepton Algorithm

CM1			
	True N	True Y	Precision
Pred.N	295	34	89.67%
Pred.Y	7	8	53.33%
Recall	97.68%	19.05%	

The accuracy value of the confusion matrix is as follows:

$$\text{accuracy} = \frac{(TN+TP)}{(TN+FN+TP+FP)} = \frac{(295+8)}{(295+7+8+34)} = 0.8808 = \mathbf{88,08}\%$$

ROC Curve for CM1 dataset :

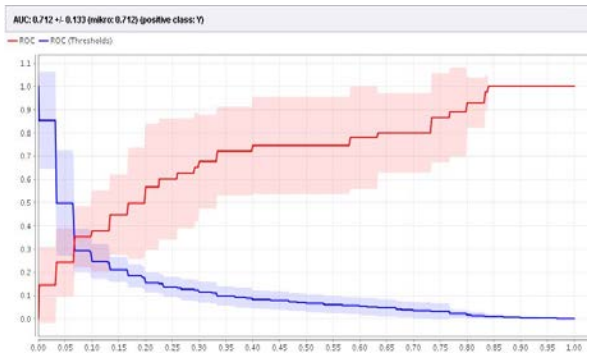

Figure 7. AUC for CM1 Dataset with Multi Layer Perceptron Algorithm

The dataset is tested using Naïve Bayes method with Genetic Algorithm optimization.

Results obtained from Naive Bayes + GA method, ie with AUC value: 0.723; TP: 13; TN: 284; FP: 29; And FN: 18. Of

the 4 values for the confusion matrix generate the following table:

Table 4. Confusion Matrix for CM1 Dataset with Multi Layer Percepton Algorithm

CM1			
	True N	True Y	Precision
Pred.N	284	29	90.73%
Pred.Y	18	13	41.94%
Recall	94.04%	30.95%	

The accuracy value of the confusion matrix is as follows:

$$\text{accuracy} = \frac{(TN + TP)}{(TN + FN + TP + FP)}$$
$$= \frac{(284 + 13)}{(284 + 18 + 13 + 29)}$$
$$= 0{,}86337 = \mathbf{86,33}\%$$

ROC Curve for CM1 dataset :

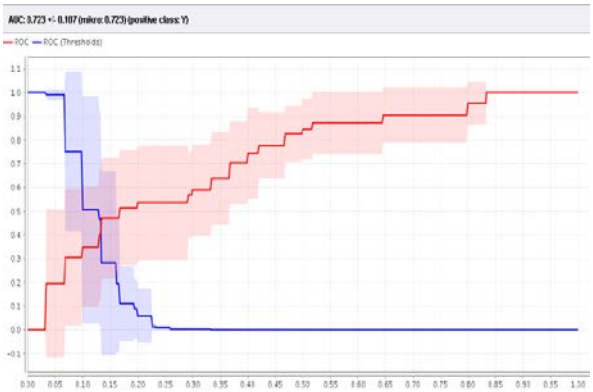

Figure 8. AUC for CM1 Dataset with Naïve Bayes Algorithm

Table 5. Comparison of Accuracy Algorithm C4.5, MLP and Naive Bayes with GA Optimization

Data Set	Accuracy GA		
	C4.5	MLP	NB
CM1	86.34%	88.09%	86.34%
JM1	81.46%	82.25%	81.80%
KC1	84.49%	79.53%	85.73%
KC3	83.00%	85.00%	83.00%
MC1	97.21%	96.20%	96.58%
MC2	72.63%	77.17%	75.06%
MW1	86.42%	89.39%	87.61%
PC1	92.09%	93.41%	91.44%
PC2	97.41%	97.66%	97.48%
PC3	87.38%	84.00%	86.58%
PC4	88.13%	89.03%	90.13%
PC5	97.21%	94.15%	97.23%
Average	87.81%	87.99%	88.25%

Table 6. Comparison of AUC Value Algorithm C4.5, MLP and Naive Bayes with GA Optimization

Data Set	AUC dengan GA		
	C4.5	MLP	NB
CM1	0.540	0.712	0.723
JM1	0.723	0.709	0.627
KC1	0.726	0.726	0.790
KC3	0.563	0.624	0.726
MC1	0.723	0.769	0.872
MC2	0.583	0.691	0.664
MW1	0.476	0.772	0.769
PC1	0.642	0.822	0.795
PC2	0.585	0.906	0.736
PC3	0.721	0.658	0.779
PC4	0.727	0.893	0.832
PC5	0.890	0.801	0.947
Rata-Rata	0.658	0.757	0.772

6. *Deployment Phase*

At this stage a model that has the best accuracy in the software development department or the relevant agency to predict Software Defect using new data.

3 CONCLUSION AND FUTURE WORK

In this research, a new model is proposed using C4.5, MLP and Naïve Bayes method of optimization with Genetic Algorithm performed on NASA Dataset. The proposed model is comparative to produce the most appropriate and most accurate model for predicting software defects. To produce the most accurate value, we used cross validation at the test stage and used the Genetic Algorithm method for the selection of variables. Experiments on the model are evaluated and validated with confusion Matrix and AUC (Area Under Curve) with ROC (Receiver Operating Characteristic).

Based on the evaluation and validation it can be concluded that the Naïve Bayes algorithm with Genetic Algorithm has the best accuracy and performance on average for all datasets which is 88.25% and the AUC (Area Under Curve) value is 0.772.

In this study, the results of the comparison methods C4.5, MLP, and Naïve Bayes conclude that Naïve Bayes after optimized with Genetic Algorithm is more accurate on average than other methods. However, other methods have an advantage and dominate on different values on several datasets. Therefore, further research is required to determine the most dominating and superior models of all datasets, example by:

1. Using other classification algorithms contained in data mining, such as K-Nearest Neighbor, ID3, CART, Random Forest, Linear Discriminant Analysis and Neural Network (SVM, RBF) algorithms.

2. Using other optimization techniques or methods, such as Particle Swarm Optimization, Backward Elimination, Forward Selection, or others. With other optimizations, it may be able to produce more accurate and better value.
3. Using datasets whose data count is large and not duplicate. The dataset used in the complete dataset as well as the number of variables is also the same.

REFERENCES

[1] Anbarasi, M.,Anupriya, E., dan Iyengar, N. *Prediction of Heart Disease with Feature Subnet Selection using Genetic Algorithm*. International Journal Of engineering Science and Technology, 2010

[2] Alpaydin, Ethem. *Introduction to Machine Learning 2nd.* London: The MIT Press, 2010.

[3] Bramer, Max. *Principles of Data Mining.* London: Springer, 2007.

[4] Challagulla, Venkata U. B., Farock B. Bastani, and I Ling Yen. "Empirical Assessment of Machine Learning based Software Defect Prediction Techniques." *Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'05).* Computer Society, 2005.

[5] Chang, Chingpao Pao, dan Chu Chih Ping. "An Action-Based Approach for Software Process Measurement." *Journal of Systems and Software*, 2007.

[6] Chemuturi, Murali. *Mastering Software Quality Assurance.* Florida: J.Ross Publishing, 2011.

[7] Dick, S., dan A. Kandel. *Computational Intelligence in Software Quality Assurance.* World Scientific Publishing, 2005.

[8] Fenton, N.E. "A Critique of Software Defect Prediction Models." *IEEE*, 1999: 1.

[9] Fenton, N.E., dan S.L. Pfleeger. *Software Metrics, A Rigorous & Practical Approach.* London: International Thomson Computer Press, 1997.

[10] Gayatrri, N., S. Nickolas, dan A.V. Reddy. "Feature Selection Using Decision Tree Induction in Class Level Metrics *Dataset* for Software Defect Predictions." *Proceedings of the World Congress on Engineering and Computer Science*, 2010.

[11] Gorunescu, Florin. *Data Mining Concepts, Models and Techniques.* Springer-Verlag, 2011.

[12] Gustafson, David A. *Theory and Problems of Software Engineering.* McGraw-Hill Companies,Inc, 2002.

[13] Hall, T., S. Beecham, D. Bowes, Gray D., and S. Counsell. "A Systematic Literature Review on Fault Prediction Performance in Software Engineering." 2011.

[14] Han, Jiawei, and Micheline Kamber. *Data Mining: Concepts and Techniques.* Second Edition. San Francisco: Elsevier Inc., 2006.

[15] Hand, David, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining.* Cambridge: MIT Press, 2001.

[16] Kothari, C. R. *Research Methology Methods and Techniques.* India: New Age International Limited, 2004.

[17] Kusrini. *Algoritma Data Mining.* Jakarta: Andi, 2009.

[18] Larose, Daniel T. *Data Mining: Methods and Models.* Wiley Interscience, 2007.

[19] Larose, Daniel T. *Discovering Knowledge In Data : An Introduction to Data Mining.* Canada: John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

[20] Lessmann, Stefan, Bart Baesens, Christophe Mues, and Swantje Pietsch. "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings." *IEEE Transactions on Software Engineering*, 2008.

[21] Liao, T Warren, dan Evangelos Triantaphyllou. *Recent Advances in Data Mining of Enterprise Data: Algorithms and Applications.* 5 Toh Tuck Link: World Scientific Publishing Co. Pte. Ltd., 2007.

[22] Maimon, Oded, dan Lior Rokach. *Data Mining and Knowledge Discovery Handbook.* Israel: Springer Science and Business Media, 2010.

[23] Menzies, Tim, Jeremy Greenwald, dan Art Frank. "Data Mining Static Code Attributes to Learn Defect Predictors." *IEEE Transactions on Software Engineering*, 2007.

[24] Myatt, Glenn J. *Practical Guide to Exploratory Data Analysis and Data Mining.* United State: John Wiley, 2007.

[25] Oral, Atac Deniz, and Ayse Basar Bener. "Defect Prediction for Embedded Software." *IEEE*, 2007.

[26] Pai, Ganesh J., dan Joanne Bechta Dugan. "Empirical Analysis of Software Fault Content and Fault Proneness Using Bayesian Network." *IEEE Transactions on Software Engineering*, 2007.

[27] Pressman, Roger.S. "Software Engineering : A Practioner's Approach." 5th . McGrawHill. 2001.

[28] Sacha, K. *Software Engineering Techniques: Design for Quality.* Springer, 2006.

[29] Shepperd, Martin, Qinbao Song, Zhongbin Sun, and Carolyn Mair. "Data Quality: Some Comments on the NASA Software Defect Data Sets." 2011.

[30] Sommerville, Ian. *Software Engineering 8th*. Pearson Education, 2007.

[31] Song, Jia, Shepperd, Ying, dan Liu. "A General Software Defect-Proneness Prediction Framework." *IEEE Transactions On Software Engineering*, 2011.

[32] Song, Qinbao, Martin Shepperd, Michelle Cartwright, dan Carolyn Mair. "Software Defect Association Mining and Defect Correction Effort Prediction." *IEEE Transactions on Software Engineering*, 2006: 69.

[33] Sugiyanto. *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta, 2008.

[34] Trisedya, Bayu D., dan Hardinal Jais. "Klasifikasi Dokumen Menggunakan Algoritma Naive Bayes dengan Penambahan Parameter Probabilitas Parent Category." 2009.

[35] Vercellis, Carlo. *Business Intelligence: Data Mining and Optimization for Decision Making*. United Kingdom: John Wiley & Sons., 2009.

[36] Witten, Ian h., Eibe Frank, and Mark A. Hall. *Data Mining Pratical Machine Learning Tools and Techniques*. Third Edition. Burlington: Elsevier Inc., 2011.

[37] Wu, Xindong, and Vipin Kumar. *The Top Ten Algorithms in Data Mining*. Taylor & Francis Group, LLC, 2009.