SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Integration of Feature Selection with Data Level Approach for Software Defect Prediction

Ade Suryadi
Universitas Bina Sarana Informatika
Jakarta, Indonesia
ade.axd@bsi.ac.id

Abstract—The dataset of software metrics in general are not balanced (unbalanced). An imbalance distribution of classes and attributes that are not relevant may decrease the performance of the model prediction software defect, because the majority of the class predictions tend to produce than minority class. This research uses a public dataset from NASA (National Aeronautics and Space Administration) MDP (Metrics Data Program) repository. This research aims to reduce the influence of class imbalance in the dataset, so that performance can be improved in the classification of defect prediction software. The model proposed in this research is applying the technique feature selection with Particle Swarm Optimization (PSO), approaches the level of data, by using Random Under Sampling (RUS) and SMOTE (Synthetic Minority Over-sampling Technique) and (Ensemble) Bagging with Naive Bayes Classifier. Research results show that the proposed model can improve the performance of Naive Bayes of the overall value of the AUC (Area Under Curve) reached > 0.8. Statistical tests indicate that there is a significant difference between a Naive Bayes model with the model proposed by the p-value (0.043) smaller than the alpha values (0.05) which means there is a significant difference between the two models.

Keywords—Class Imbalance, Approach the Level Data, Feature Selection, Software Defect Prediction

I. INTRODUCTION

Software defect prediction is one of the testing phase in the Software Development Life Cycle (Arora, et.al, 2015). Software testing process can identify a software contains defects or not. Highest potential defects occur at the stage of encoding (Jones, 2013) compared to other stages. NASA MDP is a software metric data is frequently used in research of prediction software defect. Datasets are easily retrieved and available to the public because as much as 64.79% research using public datasets and 35.21% private dataset using research (Wahono, 2015). Problems in the prediction of software defects include redundant data, the correlation irrelevant features and missing samples. This problem can create unbalanced dataset because it is difficult to ensure the data is flawed or not (Laradji, et.al, 2015).

Feature selection (selection features or attributes) can handle to reduce the problem of redundant data nd features that are not relevant. Feature selection is an important step in Machine Learning (Laradji, et.al, 2015). The purpose of the selection of the features of which is simplifying and improving quality dataset by selecting relevant attributes. Handling class imbalance in general there are three approaches to handle the unbalanced datasets (imbalanced), that approach on the level of data, algorithmic level, and combine or pair (ensemble) method (Yap, et.al, 2014). Approach on the data tier includes several techniques, manipulating data resampling trainer to correct the distribution class leanings, such as Random Over-Sampling (ROS) and Random Under Sampling (RUS), and SMOTE (Synthetic Minority Over-sampling Technique). The approach level algorithms is combining or pair (ensemble) method, there are two algorithms are the most popular ensemble-learning, *i.e.* boosting and bagging (Yap, et.al, 2014). Bagging (bootstrap aggregating) is a technique that can improve the classification with the combination of classification at random on the dataset and bagging training can also reduce variance and avoid overfitting (Wahono and Suryana, 2013).

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

II. LITERATURE REVIEW

Research on defect prediction software has long been done and there have been many research results are published. Study of previous research can identify the methods, data, and models that have ever been used.

As research done by (Wahono, et.al, 2014), the proposed combination of metaheuristic optimization method and bagging technique to improve the performance of software defect prediction. Metaheuristic optimization methods (genetic algorithm and particle swarm optimization) applied to handle the selection of features, and Bagging technique used to deal with the problem of imbalance class. The result shows that the proposed method can provide an impressive improvement on the performance of the prediction model for most classifications.

Then research of (Putri and Frieyadie, 2017) proposes a method of sampling techniques are integrated with the method of selection of features. The method of sample selection used SMOTE. After doing research, process integration techniques SMOTE with a relief method used on Naive Bayes classification, prediction value is better than the other method i.e. 82%.

On the research (Putri and Wahono, 2015) proposed combination of techniques SMOTE with the Information Gain algorithm to enhance performance software defect prediction. SMOTE applied to handle imbalance class. While the Information Gain algorithm used to process the selection of relevant attributes to handle noise attributes. The results showed that the proposed model achieve a higher classification accuracy. Where is the average value of the AUC on the model NB SMOTE + IG is 0.798.

The proposed model is applied in this research include a selection of features with the PSO with the approach level data using RUS and techniques (ensemble) bagging with Naive Bayes. Then selection features with PSO approach level data using SMOTE and techniques (ensemble) bagging with Naive Bayes..

III. PROPOSED METHOD

4.1 Feature Selection

PSO is a Population-based optimization technique developed by Eberhart and Kennedy in 1995 [53], each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors (Cong and Shu-wei, 2015). It is initialized with a population of random potential solutions and the algorithm searches for optima satisfying some performance. The potential solutions, called particles, are flown through a multidimensional search space. To find the optimal solution, each of the particles change direction search according to two factors, the best experience before (pbest) and the best experience of all other members (gbest) (Wahono, et.al, 2014). PSO perform a search by using the population (swarm) individual (particles) that are updated from iteration to iteration.

Each particle i has a position represented by a position vector X. A swarm of particles moves through a d-dimensional problem space, with the velocity of each particle represented by a vector Vi. The particle velocity and position equations form are given by:

$$V_i(t+1) = w \cdot V_i(t) + C_i r_i \left(P_{i,best}(t) - X_i(t)\right) + C_2 r_2 \left(P_{global}(t)\right) - X_i(t) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

where t is current iteration number, w is inertia weight, c are positive constants, r_1 and r_2 are uniformly distributed random numbers in the range [0,1]. $P_{i,best}$ and P_{global} are the best previously visited position of the particle i and the best value of all particle position values, respectively. Where $X_i(t) = (X_{i1}(t), X_{i2}(t), ..., X_{id}(t))$, and $Vi(t) = (Vi_1(t), Vi_2(t), ..., V_{id}(t))$. The initial velocities in particles are probabilities limited to a range of [0,1]. The first part of Eq. (1) represents the inertia of the previous velocity, the second part is the cognition part and it tells us about the personal experience of the particle, the third part represents the cooperation among particles and is therefore named as the social component. And w, c_1 and c_2 are predefined.
The cost value of particle i, at iteration t, is as follows:

$$C\left(X_i(t)\right) = \frac{1}{q}\sum_{k=1}^{q}\sum_{j=1}^{d}\left(X_y^{(k)}(t) - X_{ij}^{(k)}(t)\right)^2 \quad (3)$$

where for particle i, X ij is the jth output component of the kth sample, and X(k)ij is the jth actual component of the kth observation sample. For Eq. (3), C(·) is as small as possible. For minimization problem, the smaller the objective function value, the better the cost value is. The best position $P_{i,best}$ of particle i is updated by the following formula:

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$P_{i,best}(t+1) = \begin{cases} X_i(t), if C(X_i(t)) < CP_{i,best}(t) \\ P_{i,best}(t), if C(X_i(t)) \geq C\left(P_{i,best}(t)\right) \end{cases} \quad \text{.....(4)}$$

At each update step of PSO, the velocity of each particle is calculated according to (1) and the position is updated according to (2). When a particle finds a better position than the previously best position, it will be stored in the memory. The algorithm goes on until a satisfactory solution is found or the max number G
of iterations is met.

4.2 Level Data Approach

Data-level approach is one approach to solving the problem of imbalance in the dataset class. The approach commonly used sample is over-sampling and random under sampling (Yap, *et.al*, 2014).

a) RUS Algorithm

Sampling was done randomly, first calculated the difference between the majority of the minority. Done looping a number of difference in the results of the calculation, as long as the majority of the class data deleted looping randomly, so the majority of the number of classes is equal to the number of classes in the minority.

b) SMOTE Algorithm

SMOTE is an approach for oversampling on minority class. The approach was conducted with sample for oversampling to create "synthetic ". This method of synthesizing new minority class samples between some examples of minorities who are located close together. This algorithm is simulated by finding the nearest k for each sample of a minority, and then for each neighbor, randomly pick a point on the line connecting neighbors and the sample itself by adding new minority samples to in the training data.

4.3 Level Algorithm Approach

The approach used is the algorithm level techniques (ensemble) bagging with Naive Bayes. Bootstrap aggregating (Bagging) is a learning method that is simple and effective. Bagging is a widely used method of ensemble for classification, with the aim to improve the accuracy of classification by combining a single classification, and the results were a bit better than random sampling (Alfaro, et.al, 2013).

Bagging is a method that combines the bootstrapping and aggregating. Bootstrap samples is obtained by performing a resampling with replacement from the original dataset to produce the same number of elements from the original dataset.

4.4 Naive Bayesian (NB) Classifier

Naïve Bayes assumes that the impact of a certain class attribute value is independent of the values of other attributes. This assumption is called the independent class conditional. This is done to simplify the calculation involved, and in this sense, it is considered naive. Naïve Bayes allows representation of dependencies among a subset of attributes (Jain and Richariya, 2012) by mathematical calculation as follows:

$$P(X|C_i) \approx \sum_{k=1}^{n} P(X_k|C_i) \quad \text{.....(5)}$$

The probability $P(X_1|C_1)$, $P(X_2|C_j)$, …, $P(X_n|C_i)$ can be easily estimated from the training set. Given that X_k refers to the attribute values for the sample X.

a) If A_k is a category, then $P(X_k|C_j)$ is number of tuples in D class C_j has a value X_k to attribute A_k, divided from $|C_{1,D}|$, number of class C_j tuples in D.
b) If A_k is a continuous value, it is usually assumed that the values have a Gaussian distribution with mean (μ) and standard deviation (σ), can be defined as follows:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\,\pi \cdot \sigma}}\ e - \frac{(x-\mu)^2}{2\sigma^2} \quad \text{.....(6)}$$

Where

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \quad \text{.....(7)}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n-1}} \quad \text{.....(8)}$$

while

$$P\left(X_\mu \mid C_i\right) = g(X_k, \mu_{ci}, \sigma_{ci}) \quad \text{.....(9)}$$

We need to calculate $\mu\, ci$ and σ_{ci}, where the mean and standard deviation of the value attribute A_k for training samples of class C_j.

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

4.5 Validation Technique

In this study using validation techniques 10-fold cross validation, with resulting confusion matrix which are described in Table 1. In the confusion matrix, TN is true negative results are classified (true negative). FN is a positive result, that is not properly classified as negative. TP is a positive result correctly classified (true positive). FP is the negative results are not correctly classified as positive (false positive).

TABLE 1. CONFUSION MATRIX

Class		**Initial Value**	
		TRUE	**FALSE**
Prediction	**TRUE**	TP	FP
Value	**FALSE**	FN	TN

Calculated values of accuracy, sensitivity or called the recall or the True Positive Rate (TPrate), specificity or called True Negative Rate (TNrate), False Positive Rate (FPrate), the False Negative Rate (FNrate), precision or so-called Positive Predictive Value (PPV), Negative Predictive Value (NPV), F-Measure, the Geometric Mean (G-Mean), and the Area Under the ROC Curve (AUC).

IV. RESULT AND DISCUSSION

4.1 Dataset

This research uses a dataset from NASA MDP repository by using the dataset i.e. 4 CM1, MW1, PC1, PC4 and using machine learning tools WEKA 3.8. Specification of the dataset used in this research are as follows

Table 2. Specification 4 Dataset from Nasa MDP repository

Dataset	Attributes	Modul	Defect	Defect (%)
CM1	38	327	42	0,1284
MW1	38	250	25	0,1
PC1	38	679	55	0,081
PC4	38	1270	176	0,1386

It can be seen that the number of classes of disability is minority classes at each dataset and more dominated by not flawed or the majority of the class.

4.2 Naive Bayes Model Testing (NB)

The first test was conducted by using a single Naive Bayes classification model against the four datasets.

TABLE 3. THE RESULTS OF THE PERFORMANCE OF NAIVE BAYES

Dataset	TPrate	FPrate	Precision	Sensitivitas
CM1	0,31	0,112	0,289	0,31
MW1	0,6	0,16	0,294	0,6
PC1	0,4	0,066	0,349	0,4
PC4	0,307	0,059	0,458	0,307

Dataset	Specificity	F-measure	G-mean	Accuracy	AUC
CM1	0,887	0,299	0,524	0,81345	0,645
MW1	0,84	0,395	0,711	0,816	0,78
PC1	0,934	0,373	0,611	0,891	0,793
PC4	0,941	0,367	0,537	0,8535	0,814

The table above indicates that the value of the AUC's highest earns on PC4 dataset of 0.814.

4.3 Testing the Model Using PSO with SMOTE and (ensemble) Bagging with Naive Bayes

The first model proposed is to use techniques selection feature using the PSO. The result of the next feature selection techniques do SMOTE for balance class and techniques (ensemble) Bagging with Naive Bayes. The results obtained against the proposed model are presented in the following table.

TABLE 4. THE PERFORMANCE RESULTS OF THE PSO, SMOTE, BAGGING AND NAIVE BAYES

Dataset	TPrate	FPrate	Precision	Sensitivitas
CM1	0,369	0,144	0,431	0,369
MW1	0,6	0,129	0,508	0,6
PC1	0,427	0,08	0,901	0,91
PC4	0,568	0,134	0,576	0,568

Dataset	Specificity	F-measure	G-mean	Accuracy	AUC
CM1	0,781	0,397	0,536	0,7452	0,733
MW1	0,8	0,55	0,692	0,8218	0,806
PC1	0,891	0,84	0,9	0,846	0,84
PC4	0,829	0,572	0,686	0,7932	0,844

The results of the analyses showed the average value of the model proposed PSO + SMOTE + BG + NB on fourth dataset covers 80% of accuracy,

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Sensitivity 0.825, Specificity 0.611, Precision 0.604, F-measure 0.589, G-Mean 0.703 and AUC 0.805.

4.4 Testing the Model Using PSO with RUS and (ensemble) Bagging with Naive Bayes

The second model proposed is to use the feature selection techniques using PSO, approaches the level of the data using the technique of RUS and techniques (ensemble) Bagging with Naive Bayes.

Table 5. The performance results of the PSO, RUS, Bagging and Naive Bayes

Dataset	Tprate	Fprate	Precision	Sensitivitas
CM1	0,429	0,167	0,72	0,429
MW1	0,64	0,2	0,762	0,64
PC1	0,509	0,109	0,824	0,509
PC4	0,614	0,17	0,786	0,614

Dataset	Specificity	F-measure	G-mean	Accuracy	AUC
CM1	0,856	0,537	0,605	0,6309	0,741
MW1	0,871	0,696	0,746	0,72	0,776
PC1	0,919	0,626	0,683	0,7	0,869
PC4	0,865	0,688	0,728	0,7215	0,82

The results of the analyses showed the average value of the model proposed PSO + RUS + BG + NB on four datasets include accuracy of 69%, Sensitivity 0.548, Specificity 0.877, Precision 0.773, F-measure 0.636, G-Mean 0.690 and AUC 0.801.

4.5 Performance Evaluation Model

Comparison of the performance of Naive Bayes classification model and optimized models are presented with the AUC comparison chart

TABLE 5. COMPARISON OF PERFORMANCE MODELS BASED ON AUC

Model	NB	PSO+SMOTE+BG+NB	PSO+RUS+BG+NB
CM1	0,645	0,733	0,741
MW1	0,78	0,806	0,776
PC1	0,793	0,84	0,869
PC4	0,814	0,844	0,82

The results show that there is a difference of the three models. The third difference model has done an experiment can be seen more clearly in the following diagram

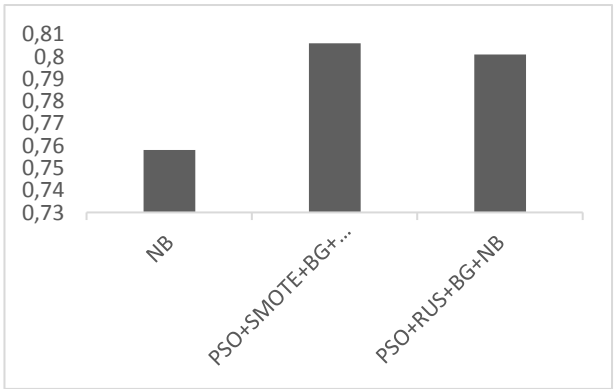

Figure 1. Diagram Comparison Of AUC

The diagram shows that the AUC comparison model PSO + RUS + BG + NB with PSO + SMOTE + BG + NB has a value higher than the AUC model NB. The value of the AUC's highest obtained from model PSO + SMOTE + BG + NB. Analysis of the AUC against the proposed model showed that the average AUC values exceeding 0.8 which means including a category either. The performance of the model proposed PSO + SMOTE + BG + NB more dominate than the other proposed model, but the model of the PSO + RUS + BG + NB also has a good standing with the difference that is not too much. The conclusion that the model proposed PSO + SMOTE + BG + NB and PSO + RUS + BG + NB can improve the performance of Naive Bayes classification model. This result is obtained from the values of the AUC that can reach 80%.

4.6 Statistical Tests

Testing to prove whether there is a significant difference between the model proposed by Naive Bayes models. Then performed statistical tests to compare the results of the value of the AUC. T-test was conducted to compare the two models by measuring the p-value, if the p-value < alpha values (0.05), then there is a significant difference between the two models. Conversely, if the p-value > alpha value, then no a significant difference. The t-test performed on the AUC by using statistical methods to test the hypothesis model NB with PSO + SMOTE + BG + NB and NB with PSO + RUS + BG + NB.

The first hypothesis:
H0: there is no significant difference between the model NB with PSO + SMOTE + BG + NB

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

H1: there is a significant difference between the model NB with PSO + SMOTE + BG + NB

TABLE 6. PAIRED SAMPLE T TEST AUC MODEL NB AND PSO + SMOTE + BG + NB

Paired Differences					T	df	Sig. (2-tailed)
Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
			Lower	Upper			
-0,04775	0,0283358	0,014168	-0,09284	-0,00266	-3,37	3	0,043

It can be seen that the value of P is 0.043 and this shows that the value of p is smaller than the alpha values (0.05) so the hypothesis H0 is rejected and the H1 is accepted. Hypothesis H1 is accepted means there is a significant difference between the model NB with PSO + SMOTE + BG + NB.

The second hypothesis:
H0: there is no significant difference between the model NB with PSO + RUS + BG + NB
H1: there is a significant difference between the model NB with PSO + RUS + BG + NB

TABLE 6. PAIRED SAMPLE T TEST AUC MODEL NB AND PSO + RUS + BG + NB

Paired Differences					T	df	Sig. (2-tailed)
Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
			Lower	Upper			
-,0435000	,0499166	,0249583	-,1229284	,0359284	-1,743	3	,180

It can be seen that the value of P is 0.180 and this shows that the value of p is greater than the value of the alpha (0.180 > 0.05) so the hypothesis H0 and H1 were rejected. The hypothesis H0 is accepted means there is no significant difference between the model NB with PSO + RUS + BG + NB.

From the second test can be concluded that there is a significant difference on the model of the proposed PSO + SMOTE + BG + NB with NB.

4.7 Comparison Between Previous Models

Research on defect prediction software has a lot do with different methods and results. Comparison of the results from previous research will show how influential contributions to the research area of knowledge although elements of a research success is not always measured from how well the numbers are in the produce.

TABLE 7. AUC OF COMPARISON BETWEEN PREVIOUS MODELS

Model	CM1	MW1	PC1	PC4	Average
Putri, Wahono (2015), NB with SMOTE and IG	0,751	0,767	0,817	0,856	0,79775
Putri, Frieyadi (2017), NB with SMOTE and RLF	0,761	0,779	0,821	0,86	0,80525
(Proposed Model), PSO+SMOTE + BG+NB	0,733	0,806	0,84	0,844	0,80575
(Proposed Model), PSO+RUS + BG+NB	0,741	0,776	0,869	0,82	0,8015

The results of comparisons with previous research showing that the performance of the proposed model are statistically higher average 0.805..

V. CONCLUSION AND SUGGESTION

Feature Selection PSO integration with data-level approach SMOTE and RUS is proposed to improve the performance of classification (ensemble) Bagging with Naïve Bayes. The proposed model is applied to 4 dataset from NASA MDP repository i.e. CM1, MW1, PC1 and PC4. The results showed that the model selection features with PSO and approaches the level of data using SMOTE and techniques (ensemble) Bagging with Naive Bayes, was able to increase the overall classification with higher AUC values of the Naive model Bayes.

The performance of the model proposed PSO + SMOTE + BG + NB can be seen from the average value of the AUC are higher than the other proposed model, but the model of the PSO + RUS + BG + NB also had good results with a difference that is not too much. The conclusion that the model proposed PSO + SMOTE + BG + NB and PSO + RUS + BG + NB can improve the performance of Naive Bayes classification model. This result is obtained from the values of the AUC that can reach 80%. However, if the comparison of the numerical values of AUC then PSO model + SMOTE + BG + NB is said to be better than both of these models. The second model based on classification criteria table AUC then it can be inferred that the model proposed is a good-value criteria in the AUC reached more than 0.8.

Based on the results of the test statistic t is well known that the average value of the PSO model + SMOTE + BG + NB is bigger than the value of the mean of a model NB. The t-test statistics showed that the value of p is 0043 and shows that the value of p is

Journal Publications & Informatics Engineering Research
Volume 4, Number 1, October 2019
DOI : https://doi.org/10.33395/sinkron.v3i1.10137
e-ISSN : 2541-2019
p-ISSN : 2541-044X

smaller than the alpha values (0.05) so that there is a significant difference between the model NB with model PSO + SMOTE + BG + NB.

VI. ACKNOWLEDGMENT

A million thanks to all lecturers, mentors, who have helped this research. Also, for all friends which has supported, shared, and gave the idea to me in completing this research.

VII. REFERENCES

Arora, I., Tetarwal, V., & Saha, A. (2015). *Open Issues in Software Defect Prediction. Procedia Computer Science*, Volume 46, p. 906-912.

Jones, C. (2013). *Software Defect Origins and Removal Methods*. Namcook Analytics.

Wahono, R. S. (2015). A *Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks. Journal of Software Engineering*, 1-16.

Laradji, I. H., Alshayeb, M., & Ghouti, L. (2015). *Software Defect Prediction Using Ensemble Learning on Selected Features. Information and Software Technology*, 388-402.

Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). *An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. Proceedings of the First International Conference on Advanced Data and Information Engineering* (DaEng-2013). 285, pp. 13-22. Singapore: Springer. doi:10.1007/978-9814585-18-7_2

Wahono, R. S., & Suryana, N. (2013). *Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect*. IJSEIA, 153-166.

Wahono, R. S., Suryana, N., & Ahmad, S. (2014). *Metaheuristic Optimization based Feature Selection for Software Defect Prediction. Journal of Software*, 1324-1333.

Putri, S. A. & Frieyadie (2017). Combining Integreted Sampling Technique with Feature Selection For Software Defect Prediction, *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, Denpasar, 2017, pp. 1-6. doi: 10.1109/CITSM.2017.8089264

Putri S. A. and Wahono R. S. (2015). *Integrasi SMOTE dan Information Gain pada Naive Bayes untuk Prediksi Cacat Software*. Journal Software Engineering, vol. 1, no. 2, pp. 86–91.

Cong jin & Shu-Wei Jin. (2015). *Prediction approach of software fault-proneness based on hybrid artificial neural network and quantum particle swarm optimization. Applied Soft*

Alfaro, E., Gamez, M., & García, N. (2013). adabag: An R Package for Classification with Boosting and Bagging. *Journal of Statistical Software*, *54*(2), 1 - 35.

Jain, M., & Richariya, V. (2012). *An Improved Techniques Based on Naive Bayesian for Attack Detection. International Journal of Emerging Technology and Advanced Engineering*, 2(1), 324-33