

PENERAPAN FEATURE SELECTION PADA BAYESIAN NETWORK UNTUK PREDIKSI CACAT PERANGKAT LUNAK

Sukmawati Anggraeni Putri¹; Dewi Larasati²

¹ Sistem Informasi STMIK Nusa Mandiri Jakarta
www.nusamandiri.ac.id
sukmawati@nusamandiri.ac.id

Teknik Komputer AMIK BSI Jakarta
www.bsi.ac.id
dewi.dwi@bsi.ac.id

Abstract— *In its development of research on the field of software defect prediction, the more popular among researchers. To reduce maintenance costs and maintain software quality. One of the with the selection of defect and non defect modules in software using machine learning. One of them is machine learning Bayesian Network, which has better performance than Naive Bayesian. As has been done in this study, that Bayesian Network by integrating the algorithm of attribute selection such as Chi Square, Information Gain and Relief. The model can produce an accuracy level of up to 0.9 on one of the Nasa dataset used in this study. Therefore the performance and accuracy of the Bayesian Network on prediction of software defect is very good.*

Key Word: *Bayesian Network, Feature Selection, Software Defect Prediction, Naive Bayesian*

Intisari—*Pada perkembangannya penelitian pada bidang prediksi cacat perangkat lunak, semakin banyak diminati oleh para peneliti. Untuk mengurangi biaya perawatan dan menjaga kualitas perangkat lunak. Salah satunya dengan, pemilihan modul cacat dan tidak cacat pada perangkat lunak menggunakan machine learning. Salah satunya adalah machine learning Bayesian Network, yang memiliki kinerja lebih baik dari Naive Bayesian. Seperti yang telah dilakukan pada penelitian ini, bahwa Bayesian Network dengan mengintegrasikan algoritma pemilihan atribut seperti Chi Square, Information Gain dan Relief. Model tersebut dapat menghasilkan tingkat akurasi hingga 0,9 % pada salah satu dataset Nasa yang digunakan pada penelitian ini. Oleh karenanya kinerja dan tingkat akurasi Bayesian Network pada prediksi cacat perangkat lunak sangat baik.*

Kata Kunci: *Bayesian Network, Pemilihan Fitur, Prediksi Cacat Perangkat Lunak, Naive Bayesian*

PENDAHULUAN

Pada perkembangannya Kualitas dari software sangat diperhatikan, terutama dalam pengujian komponen yang cacat atau rusak dapat biaya dan penggunaan waktu (de Carvalho, Pozo, & Vergilio, 2010).

Oleh karenanya hingga saat ini, banyak peneliti yang melakukan penelitian dalam memprediksi modul software yang cacat (Catal, 2011). Dimana sebagian besar penelitian yang dilakukan berfokus pada perkiraan jumlah cacat pada perangkat lunak, mencari hubungan antara modul yang cacat, serta mengklasifikasikan perangkat lunak yang cacat menjadi dua kelas yaitu cacat dan tidak cacat (Song, Jia, Shepperd, Ying, & Liu, 2011). Klasifikasi pada machine learning merupakan pendekatan populer untuk memprediksi cacat perangkat lunak. Pengklasifikasi seperti Naive Bayesian (Sukmawati Anggraeni Putri, 2017), Logistic Regression (Lessmann, Member, Baesens, Mues, & Pietsch, 2008), Bayesian Network (Bouguila, Wang, & Hamza, 2008).

Dari beberapa klasifikasi machine learning tersebut, Bayesian Network (Fenton, Krause, & Mishra, 2007) menghasilkan kinerja yang baik. Bayesian Network merupakan klasifikasi turunan Bayesian yang telah disempurnakan yang menghasilkan akurasi prediksi yang semakin baik dibandingkan dengan Naive Bayesian (Wang, Gao, & Wang, 2016). Namun metode ini tidak optimal dalam kasus yang memiliki dataset cacat perangkat lunak yang memiliki atribut yang tidak relevan atau noise atribut.

Sedangkan dataset NASA MDP (Shepperd, Song, Sun, & Mair, 2013) yang digunakan oleh peneliti sebelumnya merupakan dataset bersifat publik serta high dimensionality. Dimana atribut yang dimiliki tidak semuanya dapat digunakan (Gao & Khoshgoftaar, 2011).

Seperti penelitian yang dilakukan oleh (Kabir & Murase, 2012) menjelaskan bahwa untuk masalah ini dilakukan penyeleksian atribut dengan tujuan untuk menyederhanakan dan meningkatkan kualitas dataset. Metode seleksi atribut yang digunakan seperti Information Gain (IG), Chi Square (CS) (Gao & Khoshgoftaar, 2011) dan Relief (R) (Sukmawati Anggraeni Putri, 2017).

Ketiga metode seleksi atribut ini merupakan metode seleksi atribut filter, dimana lebih baik dibandingkan metode seleksi atribut wrapper (Gao & Khoshgoftaar, 2011).

Pada penelitian ini kami mengusulkan untuk menggunakan metode seleksi atribut filter yaitu Information Gain (IG), Chi Square (CS) dan Relief pada klasifikasi Bayesian Network untuk menghasilkan akurasi yang lebih baik pada prediksi cacat perangkat lunak.

BAHAN DAN METODE

A. Penelitian Terkait

Berjalan dengan perkembangan teknologi terutama perangkat lunak, penelitian prediksi cacat perangkat lunak telah banyak dilakukan oleh para peneliti sebelumnya. Dari penelitian tersebut dapat diketahui, saat ini permasalahan yang telah banyak dibahas adalah mengenai atribut pada dataset perangkat lunak yang tidak relevan. Sehingga kualitas dari perangkat lunak akan menurun. Oleh karenanya banyak peneliti yang melakukan penelitian untuk menemukan model yang tepat dalam menangani masalah ini.

Seperti yang dilakukan oleh (Gao & Khoshgoftaar, 2011; Gao, Khoshgoftaar, Wang, & Seliya, 2011) bahwa dataset NASA MDP yang sering digunakan para peneliti untuk meneliti prediksi cacat perangkat lunak ini memiliki kekurangan yakni banyaknya atribut yang tidak relevan. Sehingga untuk memilih atribut yang relevan pada penelitian ini menggunakan metode seleksi atribut yang terbagi menjadi dua jenis, yakni filter dan wrapper. Dari hasil penelitian ini pun diperoleh bahwa teknik filter lebih baik dibandingkan dengan teknik wrapper. Teknik filter tersebut, seperti Chi Square, Information Gain.

Sedangkan pada penelitian yang dilakukan oleh (Sukmawati Anggraeni Putri, 2017) menemukan bahwa salah satu metode seleksi atribut yaitu Relief menunjukkan kinerja yang baik pada penelitian prediksi cacat perangkat lunak. Dengan dataset yang digunakan adalah dataset NASA MDP. Dimana dataset ini memiliki atribut yang tidak relevan sehingga dapat menurunkan kinerja dari akurasi prediksi klasifikasi machine learning.

Selanjutnya, klasifikasi machine learning pada perkembangannya sering digunakan oleh para peneliti diberbagai bidang. Tidak terkecuali penelitian pada bidang prediksi cacat perangkat lunak. Seperti penelitian yang dilakukan oleh (Lessmann et al., 2008) menggunakan 22 classifier machine learning, salah satunya Bayesian Network (BN). Menurutnya BN merupakan *statistical classifier* yang memiliki performa yang baik pada proses klasifikasi. Diantara statistical classifier lainnya, seperti *Linear Discriminant Analysis* (LDA), *Quadratic Discriminant Analysis* (LQA), Logistic Regression (LogReg) serta Naive Bayes (NB). Proses ini dilakukan dengan pendekatan yang terdiri dari model multivariate *linear regression* untuk memilih atribut utama yang akan digunakan.

Selain itu, menurut penelitian (Menzies, Greenwald, & Frank, 2007) machine learning Naive Bayes menghasilkan performa yang cukup baik, yaitu sekitar 71%. Namun, pada penelitian (Sukmawati Anggraini Putri & Wahono, 2015) dilakukan penelitian untuk memperbaiki performa akurasi Naive Bayes dalam memprediksi cacat perangkat lunak. Menurut penelitian (Sukmawati Anggraini Putri & Wahono, 2015), sebelum dilakukan proses klasifikasi menggunakan machine learning Naive Bayes, maka dilakukan proses seleksi atribut menggunakan algoritma Information Gain. Proses ini untuk memilih atribut yang relevan dan memiliki performa yang baik untuk menentukan akurasi prediksi cacat perangkat lunak. Dari penelitian ini, menghasilkan akurasi prediksi cacat perangkat lunak sebesar 86%.

B. Metode

1. Seleksi Atribut

Atribut yang tidak relevan merupakan salah satu permasalahan pada penelitian prediksi cacat perangkat lunak. Terutama pada dataset perangkat lunak NASA MDP. Sehingga beberapa atribut yang tidak relevan ini harus dilakukan penghapusan untuk meningkatkan akurasi prediksi cacat perangkat lunak.

Sementara hasil dari penelitian sebelumnya yang dilakukan oleh (Gao et al., 2011) terdapat dua teknik pada metode seleksi atribut, yaitu teknik filter dan teknik wrapper. Pada penelitian tersebut dihasilkan bahwa teknik filter lebih baik dibandingkan dengan teknik wrapper. Oleh karenanya pada penelitian ini menggunakan teknik filter, seperti algoritma Chi Square (CS), Information Gain (IG) dan Relief.

2. Metode CS (Chi Square)

Metode CS dapat mengevaluasi nilai atribut dengan menghitung nilai statistik berkaitan dengan kelas. CS statistik (juga dilambangkan sebagai χ^2) merupakan teknik statistik nonparametrik dengan menggunakan data nominal (kategori) dengan tes frekuensi.

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \dots \dots \dots (1)$$

di mana χ^2 adalah statistik uji yang asimtotik mendekati χ^2 distribusi, O_i adalah frekuensi yang diamati, dan E_i adalah frekuensi yang diharapkan. n adalah jumlah hasil yang mungkin dari setiap peristiwa.

3. Metode IG (Information Gain)

Pada metode IG mampu menilai pentingnya atribut dengan mengukur information gain berkaitan dengan kelas. Secara umum IG memperkirakan perubahan entropi informasi sebelum keadaan yang mengambil beberapa informasi.

$$IG(Class, Attribute) = H(Class) - H(Class|Attribute) \dots (2)$$

di mana H menentukan entropi. Lebih khusus lagi, misalkan A adalah himpunan semua atribut dan Kelas menjadi atribut tergantung dari semua contoh pelatihan, nilai (a, y) dengan $y \in Class$ mendefinisikan nilai dari contoh spesifik untuk atribut $a \in A$, V merupakan himpunan nilai-nilai atribut, yaitu $V = \{value(a, y) \mid a \in A \cap y \in Class\}$ dan $|s|$ adalah jumlah elemen dalam set s. IG untuk atribut $a \in A$ didefinisikan sebagai berikut:

$$IG(Class, a) = H(class) - \sum_{v \in V} \frac{| \{y \in class \mid value(a, y) = v\} |}{|Class|} \times H(\{y \in class \mid value(a, y) = v\}) \dots \dots \dots (3)$$

4. Metode RLF (Relief)

Pada algoritma RLF diberikan contoh R, Relief menemukan nearest neighbor dari kelas yang sama maupun yang berbeda, yang disebut 'nearest hit H' dan 'nearest miss M'. Hal tersebut akan diperbarui estimasi kualitas W [A] untuk semua atribut A tergantung pada nilai-nilai mereka untuk R, M, dan H. Proses ini diulang sesuai dengan nilai m, dimana m ditentukan oleh pengguna. Fungsi diff (Atribut, Instance1, Instance2) didefinisikan jelas sesuai dengan jenis atribut. Untuk atribut diskrit tersebut didefinisikan sebagai berikut:

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{\max(A) - \min(A)} \dots \dots \dots (4)$$

Hipotesis yang mendasari adalah bahwa atribut yang relevan adalah mampu membedakan antara hal dari kelas yang berbeda dan menunjukkan tidak ada perbedaan antara contoh dari kelas yang sama.

5. Bayesian Network Classifier (BNC)

Bayesian Network Classifier merupakan klasifikasi probabilitas yang terkenal karena kesederhanaan, efisiensi yang tinggi dan akurasi klasifikasi yang baik. Hal ini mampu memproses atribut secara langsung dan berkesinambungan dan telah banyak digunakan dalam penelitian *medical diagnosis, text categorization, mail filtering, information retrieval* dan sebagainya. (Wang et al., 2016).

Pada dasarnya, (Sun & Erath, 2015) Bayesian Network untuk satu set variabel $\chi = \{\chi_1, \dots, \chi_n\}$ terdiri dari dua bagian yakni:

- 1) Bagian kualitatif adalah struktur jaringan G dalam bentuk directed acyclic graph (DAG), dimana simpul berada dalam pemetaan one-to-one dengan variabel acak dan link mencirikan ketergantungan antar variable yang terhubung.
- 2) Bagian kuantitatif adalah seperangkat distribusi atau tabel probabilitas $\sigma = \{P(\chi_1|\pi_1), \dots, P(\chi_n|\pi_n)\}$ untuk setiap node atau variabel χ_1 dengan kondisi pada parent π_1 yaitu :
 $P(X) = \prod_{i=1}^n P(\chi_i|\pi_i) \dots \dots \dots (5)$
Distribusi probabilitas P(X) secara eksklusif dapat dikodekan oleh pasangan (G, σ).

C. Teknik Validasi

Pada penelitian ini menggunakan teknik validasi 10 fold cross validation, dengan menghasilkan confusion matrix (Ling, 2003) yang akan dijelaskan pada Tabel 1. Pada confusion matrix, TN merupakan hasil negatif yang benar diklasifikasikan (true negative). FN merupakan hasil positif yang tidak benar diklasifikasikan sebagai negatif. TP merupakan hasil positif diklasifikasikan dengan benar (true positive). FP merupakan hasil negatif yang tidak benar diklasifikasikan sebagai positif (false positive).

Tabel 1 Confusion Matrix

Kelas	Nilai Awal		
	Benar	Salah	
Nilai Prediksi	Benar	TP	FP
	Salah	FN	TN

Sumber: Ling (2003)

Dari nilai confusion matrix akan menghasilkan kurva ROC (Receive Operating Characteristics) yang bertugas untuk mengevaluasi kinerja dari algoritma pengklasifikasi. Kemudian Area Under the ROC

sebagai acuan untuk mengevaluasi yang memberikan sejumlah ringkasan untuk kinerja algoritma pengklasifikasi (Ling, 2003).

Area Under the ROC (Receive Operating Characteristic) (AUC) adalah pengukuran nilai tunggal yang berasal dari deteksi sinyal. Nilai AUC berkisar dari 0 sampai 1. Kurva ROC digunakan untuk mengkarakterisasi *trade-off* antara *true positive rate* (TPR) dan *false positive rate* (FPR). Sebuah *classifier* yang menyediakan area yang luas di bawah kurva adalah lebih dari *classifier* dengan area yang lebih kecil di bawah kurva (Ling & Zhang, 2003).

a. Teknik Evaluasi

Pada evaluasi statistik terdiri dari uji parametrik dan uji non parametrik. Sedangkan untuk menguji perbedaan signifikan dari kinerja algoritma pengklasifikasi menggunakan uji non parametrik, seperti uji friedman (Demsar, 2006).

Uji Friedman merupakan uji non parametrik yang setara dengan ANOVA pada uji parametrik. Pada uji Friedman peringkat algoritma untuk setiap data diatur secara terpisah, kinerja algoritma yang baik mendapat peringkat 1 sedangkan untuk yang terbaik diberi peringkat dua. Uji friedman dilakukan dengan uji post hoc yang sesuai untuk perbandingan lebih dari satu pengklasifikasi dengan beberapa dataset.

Berikut akan ditampilkan formula dari uji friedman:

$$X_r^2 = \left[\frac{12}{(n \times k)(k+1)} \times \sum_{j=1}^k (R_j)^2 \right] - [(3n)(k+1)] \dots (6)$$

Dimana:

X_r^2 = nilai khai - kuadrat jenjang dua arah friedman

N = jumlah sampel

K = banyaknya kelompok sampel

1, 3, 12 = konstanta

HASIL DAN PEMBAHASAN

A. Dataset

Pada penelitian ini menggunakan dataset *software metric* dari (*National Aeronautics and Space Administration*) MDP *repository*. Dataset tersebut bersifat publik yang biasa digunakan oleh para peneliti sebelumnya pada bidang prediksi cacat perangkat lunak. Dataset NASA MDP dapat diperoleh melalui laman resmi wikispaces (<http://nasa-softwaredefectdatasets.wikispaces.com/>). Dataset yang digunakan pada penelitian ini terdiri dari CM1, MW1, PC1, dan PC4 akan dijelaskan pada Tabel 2.

Tabel 2 Dataset NASA MDP

		Data set NASA MDP				
		CM	MW	PC1	PC	
		1	1		4	
LOC Count	<i>LOC_total</i>	X	X	X	X	
	<i>LOC_blank</i>	X	X	X	X	
	<i>LOC_code_and_comment</i>	X	X	X	X	
	<i>LOC_comment</i>	X	X	X	X	
	<i>LOC_executable</i>	X	X	X	X	
	<i>Number_of_lines</i>	X	X	X	X	
Atribut Halstead	<i>Content</i>	X	X	X	X	
	<i>Difficulty</i>	X	X	X	X	
	<i>Effort</i>	X	X	X	X	
	<i>Error_est</i>	X	X	X	X	
	<i>Length</i>	X	X	X	X	
	<i>Level</i>	X	X	X	X	
	<i>Prog_time</i>	X	X	X	X	
	<i>Volume</i>	X	X	X	X	
	<i>Num_operands</i>	X	X	X	X	
	<i>Num_operators</i>	X	X	X	X	
	<i>Num_unique_operands</i>	X	X	X	X	
	<i>Num_unique_operators</i>	X	X	X	X	
	Atribut McCabe	<i>Cyclomatic_complexity</i>	X	X	X	X
		<i>Cyclomatic_density</i>	X	X	X	X
<i>Design_complexity</i>		X	X	X	X	
Atribut miscellaneous (lainnya)	<i>Essential_complexity</i>	X	X	X	X	
	<i>Branch_count</i>	X	X	X	X	
	<i>Call_pairs</i>	X	X	X	X	
	<i>Condition_count</i>	X	X	X	X	
	<i>Decision_count</i>	X	X	X	X	
	<i>Decision_density</i>	X	X	X	X	
	<i>Design_density</i>	X	X	X	X	
	<i>Edge_count</i>	X	X	X	X	
	<i>Essential_density</i>	X	X	X	X	
	<i>Parameter_count</i>	X	X	X	X	
	<i>Maintenance_severity</i>	X	X	X	X	
	<i>Modified_condition_count</i>	X	X	X	X	
	<i>Multiple_condition_count</i>	X	X	X	X	
	<i>Global_data_complexity</i>					
	<i>Global_data_density</i>					
	<i>Normalized_cyclomatic_complexity</i>	X	X	X	X	
	<i>Precent_comments</i>	X	X	X	X	
<i>Node_count</i>	X	X	X	X		
Jumlah atribut kode	37	37	37	37		
Jumlah modul	342	26	759	139		
Jumlah modul cacat	41	28	61	178		

Sumber: Lessmann (2008)

Seperti yang ditampilkan pada Tabel 2, bahwa setiap dataset terdiri dari beberapa modul perangkat lunak, bersama dengan jumlah kesalahan dan atribut kode karakteristik. Pemrosesan awal dataset NASA MDP memiliki 38 atribut ditambah satu atribut cacat atau tidak cacat (*defective?*). Atribut tersebut terdiri dari jenis atribut Halstead, McCabe, *Line of Code* (LOC) dan atribut *miscellaneous* (Lessmann et al., 2008).

Dataset diperoleh dari matrik perangkat lunak NASA MDP yang akan dijelaskan pada Tabel 3, sebagai berikut:

Tabel 3 Spesifikasi dan Atribut NASA MDP

Sistem	Bahasa	Dataset	Total LOC
Instrumen pesawat ruang angkasa	C	CM1	17K
Database	C	MW1	8K
Software penerbangan untuk satelit yang mengorbit bumi	C	PC1 PC4	26K 30K

Sumber: Lessmann (2008)

B. Pembahasan dan Hasil Eksperimen

Penelitian yang dilakukan menggunakan machine learning Bayesian Network Classifier (BNC) ini menggunakan 4 dataset NASA MDP pada proses pengujiannya. Selain itu untuk memilih atribut yang relevan maka digunakan metode seleksi atribut filter yang terdiri dari Chi Square (CS), Information Gain (IG) dan Relief. Ketiga metode seleksi atribut ini akan diterapkan pada machine learning Bayesian Network Classifier (BNC). Seperti, BNC + CS, BNC + IG, serta BNC + Relief.

Hasil percobaan dapat dilihat pada Tabel 4 yakni nilai AUC setiap algoritma yang digunakan

Tabel.4 Nilai AUC

Classifier	CM1	MW1	PC1	PC4
Naive Bayes	0,69	0,727	0,77	0,8
Naive Bayes + Chi	0,71	0,725	0,8	0,8
Naive Bayes + Relief	0,75	0,753	0,79	0,8
Naive Bayes + IG	0,74	0,725	0,79	0,8
Net Bayes	0,73	0,688	0,8	0,8
Bayes Net + Chi	0,78	0,696	0,83	0,9
Bayes Net + Relief	0,73	0,682	0,81	0,9
Bayes Net + IG	0,76	0,696	0,83	0,9

Sumber: Sukmawati (2017)

Dari hasil percobaan yang dijelaskan pada Tabel 4 dapat diketahui nilai AUC terbaik pada dataset yang digunakan yakni CM1, MW1, PC1, PC4 yakni pada percobaan menggunakan machine learning Bayesian Network dengan algoritma seleksi atribut chi square dan Information Gain. Dengan hasil terbaik pada dataset CM1, PC1, PC4. Sedangkan pada MW1 memperlihatkan hasil terbaik ketika menggunakan *machine learning* Naive Bayes dengan algoritma seleksi atribut Relief.

Sehingga dapat disimpulkan kinerja machine learning Bayesian Network lebih baik dibandingkan Naive Bayesian ketika digunakan pada penelitian prediksi cacat perangkat lunak. Dimana machine learning Bayesian Network dapat meningkatkan akurasi dari prediksi cacat perangkat lunak dibandingkan menggunakan machine learning Naive Bayesian.

KESIMPULAN

Dari hasil perhitungan pada penelitian integrasi machine learning Bayesian Network (BNet) dengan seleksi atribut yakni Chi Square, Information Gain dan Relief pada klasifikasi Naive Bayes menghasilkan nilai AUC yang lebih baik dibandingkan dengan model lainnya. BNet dan Chi Square lebih unggul pada tiga set dataset dari empat dataset yang digunakan, yakni dengan nilai 78% pada dataset CM1, 83% pada dataset PC1 dan 90% pada dataset PC4. Sedangkan hasil kurang baik di dataset MW1 hanya 69%, hasil ini lebih baik menggunakan Naive Bayesian dengan nilai 75%. Dapat dilihat dari Tabel 4, bahwa machine learning BNet lebih baik dibandingkan dengan tingkat akurasi pada machine learning Naive Bayesian.

Tetapi dari hasil tersebut, penggunaan machine learning dan algoritma seleksi atribut pada penelitian prediksi cacat perangkat lunak dapat dilakukan pengembangan pada penelitian selanjutnya, diantaranya:

- Untuk pemilihan atribut pada penelitian lanjutan dapat menggunakan teknik wrapper pada metode seleksi atribut.
- Pada penelitian lanjutan dapat menggunakan kombinasi antara metode sampel dengan metode ensemble untuk meningkatkan kinerja pengklasifikasi.
- Pada penelitian lanjutan dapat menggunakan pengklasifikasi lainnya, seperti Logistic Regression, Neural Network dan SVM.

REFERENSI

Bouguila, N., Wang, J. H., & Hamza, A. Ben. (2008). A Bayesian Approach for Software Quality Prediction. In *Intelligent Systems, 2008. IS '08. 4th International IEEE Conference* (pp. 49-54). <http://doi.org/10.1109/IS.2008.4670508>

Catal, C. (2011). Software fault prediction: A literature review and current trends. *Expert Systems with Applications, 38*(4), 4626-4636. <http://doi.org/10.1016/j.eswa.2010.10.024>

de Carvalho, A. B., Pozo, A., & Vergilio, S. R. (2010). A symbolic fault-prediction model based on multiobjective particle swarm optimization. *Journal of Systems and Software, 83*(5), 868-882. <http://doi.org/10.1016/j.jss.2009.12.023>

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *The*

- Journal of Machine Learning Research*, 7, 1–30.
- Fenton, N., Krause, P., & Mishra, R. (2007). Predicting software defects in varying development lifecycles using Bayesian nets. *Information and Software Technology*, 49, 32–43.
<http://doi.org/10.1016/j.infsof.2006.09.001>
- Gao, K., & Khoshgoftaar, T. M. (2011). Software Defect Prediction for High-Dimensional and Class-Imbalanced Data. *Conference: Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering*, (2).
- Gao, K., Khoshgoftaar, T. M., Wang, H., & Seliya, N. (2011). Choosing software metrics for defect prediction: an investigation on feature selection techniques. *Software: Practice and Experience*, 41(5), 579–606.
<http://doi.org/10.1002/spe>
- Kabir, M., & Murase, K. (2012). Expert Systems with Applications A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems With Applications*, 39(3), 3747–3763.
<http://doi.org/10.1016/j.eswa.2011.09.073>
- Lessmann, S., Member, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. *IEEE Transactions on Software Engineering*, 34(4), 485–496.
- Ling, C. X. (2003). Using AUC and Accuracy in Evaluating Learning Algorithms, 1–31.
- Ling, C. X., & Zhang, H. (2003). AUC: a statistically consistent and more discriminating measure than accuracy. *Proceedings of the 18th International Joint Conference on Artificial Intelligence*.
- Menzies, T., Greenwald, J., & Frank, A. (2007). Data Mining Static Code Attributes to Learn Defect Predictors. *IEEE Transactions on Software Engineering*, 33(1), 2–13.
<http://doi.org/10.1109/TSE.2007.256941>
- Putri, S. A. (2017). Combining Integrated Sampling Technique with Feature Selection for Software Defect Prediction. In *2017 5th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1–6). Denpasar, Indonesia.
<http://doi.org/10.1109/CITSM.2017.8089264>
- Putri, S., A. (2017). Laporan Akhir Penelitian Mandiri. Jakarta: STMIK Nusa Mandiri Jakarta
- Putri, S. A., & Wahono, R. S. (2015). Integrasi SMOTE dan Information Gain pada Naive Bayes untuk Prediksi Cacat Software. *Journal of Software Engineering*, 1(2), 86–91.
- Shepperd, M., Song, Q., Sun, Z., & Mair, C. (2013). Data Quality: Some Comments on the NASA Software Defect Data Sets. *Software Engineering, IEEE Transactions*, 39(9), 1–13.
- Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011). A General Software Defect-Proneness Prediction Framework. *IEEE Transactions on Software Engineering*, 37(3), 356–370.
<http://doi.org/10.1109/TSE.2010.90>
- Sun, L., & Erath, A. (2015). A Bayesian network approach for population synthesis. *TRANSPORTATION RESEARCH*, 61, 49–62.
<http://doi.org/10.1016/j.trc.2015.10.010>
- Wang, S., Gao, R., & Wang, L. (2016). Bayesian network classifiers based on Gaussian kernel density. *Expert Systems with Applications: An International Journal*, 51(C), 207–217.
<http://doi.org/10.1016/j.eswa.2015.12.031>