A Comparative Analysis of Filter-based Feature Selection Methods for Software Fault Prediction

Ha Thi Minh Phuong[1], Le Thi My Hanh[2], Nguyen Thanh Binh[1]
[1] The University of Danang - Vietnam-Korea University of Infomation and Communication Technology
[2] The University of Danang - University of Science and Technology

Tác giả liên hệ: Ha Thi Minh Phuong, Email: htmphuong@vku.udn.vn
Ngày nhận bài: 28/12/2020, ngày sửa chữa: 09/04/2021, ngày duyệt đăng: 19/05/2021
Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.969

Abstract: The rapid growth of data has become a huge challenge for software systems. The quality of fault prediction model depends on the quality of software dataset. High-dimensional data is the major problem that affects the performance of the fault prediction models. In order to deal with dimensionality problem, feature selection is proposed by various researchers. Feature selection method provides an effective solution by eliminating irrelevant and redundant features, reducing computation time and improving the accuracy of the machine learning model. In this study, we focus on research and synthesis of the Filter-based feature selection with several search methods and algorithms. In addition, five filter-based feature selection methods are analyzed using five different classifiers over datasets obtained from National Aeronautics and Space Administration (NASA) repository. The experimental results show that Chi-Square and Information Gain methods had the best influence on the results of predictive models over other filter ranking methods.

Keywords: *Feature selection, filter, wrapper, hybrid, embedded.*

Tên bài: **Phân tích so sánh các kỹ thuật lựa chọn đặc trưng dựa trên phương pháp lọc trong dự đoán lỗi phần mềm**

Tóm tắt: Sự phát triển mạnh mẽ của dữ liệu trong các hệ thống đã trở thành một thách thức lớn cho ngành công nghệ phần mềm. Chất lượng của các mô hình dự đoán lỗi phụ thuộc nhiều vào chất lượng của các tập dữ liệu. Trong đó, dữ liệu đa chiều là vấn đề chính ảnh hưởng đến hiệu quả của các mô hình dự đoán lỗi. Để giải quyết vấn đề dữ liệu đa chiều này, phương pháp lựa chọn đặc trưng đã được các nhà nghiên cứu tập trung khai thác trong những năm gần đây. Phương pháp lựa chọn đặc trưng cung cấp một giải pháp hiệu quả bằng cách loại bỏ các thuộc tính bị nhiễu, không liên quan và dư thừa từ đó góp phần giảm thời gian tính toán và cải thiện độ chính xác của mô hình học máy. Trong nghiên cứu này, tác giả tập trung phân tích và so sánh hiệu quả của các kỹ thuật lựa chọn đặc trưng dựa trên phương pháp lọc. Ngoài ra, chúng tôi tiến hành thực nghiệm để so sánh hiệu quả của năm kỹ thuật lựa chọn đặc trưng dựa trên phương pháp lọc trên các thuật toán phân loại khác nhau. Thực nghiệm được tiến hành trên các bộ dữ liệu thu được từ kho dữ liệu NASA. Các kết quả thực nghiệm cho thấy các phương pháp Chi-Square và Information Gain cho kết quả của các mô hình dự đoán tốt hơn so với các phương pháp lọc khác.

Từ khóa: *Lựa chọn đặc trưng, phương pháp lọc, phương pháp bao bọc, phương pháp lai, phương pháp nhúng.*

I. INTRODUCTION

Recently, applications have generated massive amounts of data such as video, photos, voice and data obtained from social networking and from the cloud computing. Such complex data contains the characteristics of multi-dimensional dimensions, noisy data, redundant or missing attributes that pose challenges for data analysis and decision making. In order to handle this issue, feature selection has been investigated in the data preprocessing phase. Feature selection (FS) is used to select optimal subset that improves the performace of the predictive model. Feature selection minimizes redundant, irrelevant, and interference features. As a results, feature selection may improve efficiency of classifiers and give the most informative features [1]. FS methods are classified into three categories including Filter, Wrapper and Embedded. Filter method is divided into Filter Feature Ranking and Filter Feature Subset Selection [2]. Filter Feature Ranking methods assess and rank attributes in datasets which are independent with learning algorithms that requires less computation time. Some of the statistical measurement methods used in the

Filter include Information gain, Chi-square, Fisher score, Gain Ratio, and Relief. Wrapper uses machine learning techniques to evaluate a subset of features against their respective criteria. Wrapper's performance is dependent on the sorting algorithm. The best subset of the features are selected based on the results of the classification algorithm. Computationally, Wrapper methods require more complex computation than the Filters, due to the iterative learning steps and cross validation. However, these methods are more accurate than Filter. Some algorithms used in Wrapper are Recursive feature elimination [3], Sequential feature selection algorithm [4] and Genetic algorithm. The third approach is the Embedded method which uses combined learning methods and hybrid methods to select optimal features. Feature selection is a crucial data preprocessing step as it solves the high dimensional data [5], improves the quality of data, redudes the computational time and increases the predictive performance of models. In this study, we examine a comparative performance analysis of five filter-based feature methods namely Chi-Square, Fisher Score, Information Gain, Gain Ratio and Relief based on five different classifiers. Five classifiers including K-nearest Neighbors, Decision Tree, Random Forest, Naive Bayes, and Multilayer Perceptron. From our experimental results, Chi-Square and Information Gain recorded the best performance of filter methods across datasets and models. The rest of this paper is structured as follows. Section 2 presents a literature review of feature selection methods. Experimental design and results of some filter-based feature selection methods are presented in section 3 and section 4. Section 5 concludes the comparative analysis and discusses future work.

II. BACKGROUND

Studies have shown that feature selection techniques can improve predictive efficiency and accuracy for machine learning techniques. The feature selection technique plays an important role in minimizing computational complexity, capacity and cost [6].

1. Feature Selection Process

The process of selecting features in a data set consisting of 4 stages: selecting search techniques, determining search strategy, determining evaluation criterion and reaching stopping criteria.

a) Selecting search techniques

Ang et al. [7] state that the first stage in the feature selection process is to find subsets search techniques. Search techniques are classified into forward, backward and random search. The search process starts with an empty set so that new properties are added in each loop called forward search. In contrast to forward search, backward search starts with a data set with full properties and properties are discarded until an optimal subset is reached. Another approach is random search which constructs a subset of properties by adding and removing properties at each loop. After selecting the trait search techniques, the search strategy will be applied at stage 2.

Table I
LIST OF FEATURE SELECTION METHODS

	Feature Selection	Search Method
Filter-based Feature Ranking Methods	Information Gain	Ranker Search
	Attribute Evaluator(IG)	Ranker Search
	Gain Ratio Attribure	Ranker Search
	Evaluator(GR)	Ranker Search
	Chi-Square	Ranker Search
	Fisher Score	Ranker Search
	Relief	Ranker Search
	Correlation	Correlation
Filter-based Subset Selection	Correlation-based Feature Subset Selection (CFS)	Best First Search (BFS)
		Greedy
		Greedy Stepwise Search (GSS)
		Ant Search (AS)
		Bat Search (BAT)
		Firefly Search (FS)
		Genetic Search (GS)
		PSO
		Search (PSOS) Best
	Consistency Feature Subset Selection (CNS)	Best First Search (BFS) Greedy
		Greedy Stepwise Search (GSS)
		Ant Search (AS)
		Bat Search (BAT)
		Firefly Search(FS)
		Genetic Search (GS)
		PSO Search (PSOS)

b) Determine search strategy

The possible search strategies are randomized, exponential and sequential in the literature. Table I lists the different search strategies and their algorithms [2]. A good search strategy requires optimal solutions, local search capabilities and computational efficiency [8]. Based on these search requests, the algorithms are further classified as the optimal and suboptimal selection of the algorithm.

c) Determine evaluation criterion

The most optimal features are selected based on evaluation criteria. Based on the technical evaluation methods, the features [9] are classified into Filter, Wrapper, Embedded and Hybrid.

d) Reach stopping criteria

The selection criteria specify a process for feature selection that stops when the optimal feature subset is reached. Stopping criteria for feature selection are effective with low complexity computation finding subsets of optimal features

and solving overfitting problems. The choice of the stopping standards is influenced by the preexecution stages. Some of the stopping standards include: predefined quantities of properties, predetermining the number of repetitions, percentage progress between 2 consecutive loops, relying on evaluation functions.

e) Validate the final results

To evaluate the results of the feature selection techniques, some evaluation measures are used such as Crossvalidation, Confusion matrix, Jaccard similarity-based measure and Rand Index. Some of the evaluation metrics for classification techniques classification and clustering include Accuracy, Precision, Recall...

2. Filter-based Feature Selection Techniques

The Filter method relies on the unique feature of the data to evaluate and select a subset of properties, using the evaluation criteria extracted from the data set such as distance, information, dependency, consistency. Specifically, the filter method uses typical criteria of the ranking technique and the ranking order method for the selection of variables. Due to simplicity, high performance and easy to generate optimal features, filter methods are used in this study [10].

a) Chi-square

In statistics, Chi-square [11] is applied to check the independence of two events, where if two events A and B are defined as independent if $P(AB) = P(A).P(B)$, which is equivalent to $P(A|B) = P(A)$ and $P(B|A) = P(B)$. In feature selection, the two main events are features and targets. We use the Chi-square value to find out which feature contains a lot of information for the model. We calculate the Chi-square value between each feature and the target. Characteristics that give high value is a good one. Chi-square is calculated as follows:

$$X^2_{(D,t,c)} = \Sigma_{e_t \in 0,1} \Sigma_{e_c \in 0,1} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (1)$$

where e_t, e_c have 2 values: 0 and 1, N is the observed value in D and E is the expected value.

b) Fisher score

Fisher score is one of the most widely used feature selection algorithm for determining a subset of features. The idea in Fisher score is to select each feature independently according to its score under the Fisher criterion. The Fisher Score of the j^{th} feature is computed below [12]:

$$F(x^j) = \frac{\sum_{k=1}^{c} n_k(\mu_k^j - \mu^j)^2}{(\sigma^i)^2} \quad (2)$$

where $(\sigma^i)^2 = \sum_{k=1}^{c} n_k(\sigma_k^i)^2$
μ_k, n_k are the mean vector and size of the k^{th} class respectively in the reduced data space. After computing the Fisher score for each feature, it selects the top-m ranked features with large scores.

c) Information Gain

Information Gain is an entropy-based feature evaluation method. For example, in text classification problem, it is defined as the amount of information provided by the feature item for text category. Information gain is calculated by how much of a term can be used for classification of information, in order to measure the importance of lexical items for the classification. The formula of the information gain is shown below [13]:

$$G(D,t) = -\sum_{i=1}^{m} P(C_i) log P(C_i) + P(t) \sum_{i=1}^{m} P(C_i|t) log P(C_i|t) + P(\bar{t}) \sum_{i=1}^{m} P(C_i|\bar{t}) log P(C_i|\bar{t}) \quad (3)$$

C is a set of document collection, in whitch there is not the feature t. The value of $G(D,t)$ is greater; t is more useful for the classification for C. This t should be selected. If the greater value of $G(D,t)$ is wanted, it should make the value of $P(t)$ and $P(\bar{t})$ smaller.

d) Gain Ratio

Gain ratio is a modification of the information gain that reduces its bias. Gain ratio takes number and size of branches into account when choosing a feature. It corrects the information gain by taking the intrinsic information of a split into account. Intrinsic information is entropy of distribution of instances into branches. Value of feature decreases as intrinsic information gets larger [14].

$$Gainratio(Feature) = \frac{Gain(Feature)}{Intrinsicinfo(Feature)} \quad (4)$$

e) Relief

The main idea of the Relief algorithm by Kira and Render [15] is similar to the basic rules of k-nearest neighbor algorithm. Being the same class is much more likely for the closer distances to a given distance. If a feature is useful, it is expected that the closest distances of the same class is closer to range given throughout this attribute than the closest distances of all the other classes. The weight of a given feature is calculated below:

$$W = \frac{(W - diff(x_{xj}), nearhit_{ij}^2 + diff(x_{ij}, nearmiss_{ij})^2)}{m} \quad (5)$$

where m is the sample size (randomly selected from a subnet of the training set), $diff(x_{xj}, nearhit_{ij}^2)$ is difference between values of attribute within randomly selected j

Table II
THE DESCRIPTION OF DATASETS

Dataset	Lang	No. of features	No. of Modules	No. of Defective Modules	Defect rate
CM1	C	41	505	48	9.5
MW1	C	38	253	27	10.67

distance and $nearhit_{ij}^2$ value of attribute within the closest training sample in the same class, $diff(x_{ij}, nearmiss_{ij})$ is described as value of the closest training sample from different class. For an useful attribute, x_{ij} and $nearmiss_{ij}$ values are expected to be very close to each other. If an attribute is not useful, both differences are expected to take almost the same distribution.

III. EXPERIMENTAL DESIGN

1. Experimental Setup

From previous studies, many researchers proved that feature selection methods can improve the performance of predictive models. The selection of optimal metrics considerably reduces the computational time, complexity and storage [6]. However, only small number of studies [16, 17] have been carried out to evaluate the effectiveness of different feature selection methods and identify which ones are the most useful. Therefore, we examine the comparative performance analysis of 5 filter-based feature ranking methods namely Chi-Square, Fisher Score, Information Gain, Gain Ratio and Relief based on five different classifiers. The classifiers are K-nearest Neighbors, Decision Tree, Random Forest, Naive Bayes and Multilayer Perceptron. The experiments were conducted on two datasets namely CM1 and MW1 from NASA repository. A brief description of the datasets is shown in Table II.

2. Software Defect Datasets

The experimental results are affected by the quality of datasets, hence choosing the right datasets plays an important role. According to Catal and Diri [18], by using datasets from National Aeronautics Space Administration (NASA) Facility Metrics Data Program (MDP) repository[14], the classification results are more reliable. A brief description of the datasets with the number of features and modules is shown in Table II.

3. Performance Evaluation Metrics

In order to assess and rank features in datasets, filter feature ranking method uses the computational metrics of dataset. After grading each feature based on different characteristics such as statistics, probability or instance, features are generated according to their score [18]. In this study, $log_2 N$ [19] is used to filter top-ranked features, where N is the number of metrics in full defect dataset. Table III illustrates the filter feature ranking methods used in this study.

In order to evaluate the effectiveness of filter feature selection methods, performance metrics include Accuracy, Recall, Precision, AUC (area under ROC curve) and F-measure. Table IV presents a confusion matrix for the performance of a model.

- True Positive (TP): indicates that the positive samples are correctly labeled by the classifier.
- True Negative (TN): indicates that to the negative samples are correctly labeled by the classifier.
- False Positive (FP): indicates that the negative samples are incorrectly labeled as positive.
- False Negative (FN): indicates that the positive samples are mislabeled as negative.
- Accuracy is a ratio of correctly predicted observation to the total observations. In the experiment, the value of accuracy is ranged from 0 to 1.
- Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The high precision shows that the predictive model is more accurate.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

F_1 is the weighted average of Precision and Recall. F_1 assesses whether the increase in precision (recall) is greater than the reduction in recall (precision).

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

ROC curve (the Receiver Operating Curve) and AUC (Area Under the ROC Curve) evaluate the performance of software fault prediction models. The nearer the curve reaches the left border of the ROC space and then reaches the top border of the ROC space, the more correct the test. Otherwise, the curve approaches the diagonal line and AUC value is small; the fault prediction model brings low efficiency.

IV. EXPERIMENTAL RESULTS

This section illustrates the experimental results of comparative performance analysis of five above filter-based feature selection methods based on the accuracy metric. The results were compared on two cases (with and without feature selection). Table V and table VI present the comparisons of filter methods on each of the five classifiers respectively. From the results of table V, on the CM1 datasets, Chi-Square method had the highest accuracy value

Table III
FILTER FEATURE RANKING METHODS

Filter feature ranking methods	Search Method	Measured-based	Reference
Gain Ratio	Ranker Search	Probability-based	[20],[21]
Information Gain	Ranker Search	Statistical based	[20],[21]
Chi-square	Ranker Search	Statistical based	[20],[21]
Relief	Ranker Search	Instance based	[20],[21]
Fisher Score	Ranker Search	Statistical based	[21],[22]

Table IV
CONFUSION MATRIX

	Predicted	
Observed	**Faulty**	**Non-Faulty**
Fault Prone	C	41
Not Fault Prone	C	38

on the predictive performance of Multilayer Perceptron and Naive Bayes with 86.86 and 85.34 respectively. Moreover, Decision Tree and Random Forest based on Gain Ratio reached the best accuracy values of 82.89 abd 85.05. This result showed Chi-Square gave the best performance on CM1 datasets compared to other filter methods.

In table VI, Multilayer Perceptron with Information Gain achieved the highest accuracy of 97.83 on the MW1 dataset. Furthermore, the accuracy performance of all classfiers based on FS methods, in this case, Gain Ratio, Information Gain and Relief were better than when no FS methods are applied. Specifically, Naive Bayes using Gain Ratio and Information Gain had the highest accuracy value of 97.51 in comparison to 95.58 value achieved by the model had no FS methods. Same also was observed for K-nearest Neighbors, Decision Tree and Random Forest, the accuracy values are better than models without FS methods. Hence, the selection of these metrics can increase the accuracy of classifiers compared to models without feature selection. It was observed that Chi-Square and Information Gain had the best influence on the prediction models over other filter methods. This clearly shows the predictive model developed with feature selection methods outperformed other models.

It is clearly observed that based on each method, the best subset of features was selected with a certain threshold value. In table V and table VI, we have shown that the selected features were suggested for each methods. In order to generate the number of features by feature selection methods, the metric values are calculated based on $log_2 N$ where N is the number of metrics in the full defect dataset. In this case, the number of selected features is 6 on both CM1 and MW1 datasets. As presented in table V and table VI, the selected features are listed based on specific threshold values which are calculated by the formula given in section III.3. We observed that the filter methods with the lesser of features achieve better performance compared to models which were trained with all the features. This concludes that reducing number of features improve the performance of the prediction models.

V. CONCLUSION

In software fault prediction, the datasets used for machine learning may contain irrelevant, redundancy features or to be noisy. Hence, feature selection methods for training data plays an important role in selecting the optimal features for predictive models to improve the quality of system and achieve a strong prediction model. Our objective is to empirically evaluate the performance of different filter-based ranking methods using different classifiers over two datasets from NASA repository. The filter feature selection methods are Gain Ratio, Information Gain, Relief, Chi-square and Fisher Score. From the experimental results, Chi-Square and Information Gain are potential methods which had high improvement on the predictive performance of classifiers on the CM1 and MW1 respectively. Additionally, further analysis showed list of selected features for each method based on the threshold value. It was conclusively discovered that the performance of feature selection methods varies from the used datasets and the choice of classification algorithm. In the future work, we will study the ensemble learning in order to propose a hybrid approach which is based on feature selection and ensemble techniques.

REFERENCES

[1] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, vol. 207. Springer, 2006.

[2] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, "Performance analysis of feature selection methods in software defect prediction: a search method approach," *Applied Sciences*, vol. 9, no. 13, p. 2764, 2019.

[3] K. Yan and D. Zhang, ""feature selection and analysis on correlated gas sensor data with recursive feature elimination"," *Sensors and Actuators B: Chemical*, vol. 212, pp. 353–363, 2015.

[4] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[5] A. Akintola, A. Balogun, F. Lafenwa-Balogun, and H. Mojeed, "Comparative analysis of selected heterogeneous classifiers for software defects prediction using filter-based feature selection methods," *FUOYE Journal of Engineering and Technology*, vol. 3, 03 2018.

[6] M. Gutkin, R. Shamir, and G. Dror, "Slimpls: A method for feature selection in gene expression-based disease classification," *PloS one*, vol. 4, p. e6416, 02 2009.

Table V
PERFORMANCE ACCURACY VALUES OF FILTER METHODS ON CM1 DATASETS

Method	**No. of selected features**	**Threshold value**	**Selected Features**	**Classifier**				
				K-nearest Neighbors	**Decision Tree**	**Random Forest**	**Naive Bayes**	**Multilayer Perceptron**
Gain Ratio	6	0.65	halstead_effort, halstead_prog_time, halstead_volume, halstead_content, halstead_difficulty, percent_comments	78.95	**82.89**	**85.05**	84.13	85.50
Information Gain	6	0.029	loc_comments, num_unique_operands, halstead_content,call_pairs, num_unique_operators, loc_executable	76.19	74.77	77.22	85.97	86.42
Relief	6	28	multiple_condition_count, halstead_level, cyclomatic_complexity, number_of_lines, loc_blank, halstead_difficulty	61.60	61.80	69.16	83.23	86.11
Chi-square	6	0.029	loc_comments, num_unique_operands, loc_blank, halstead_content, num_operators, num_unique_operators	61.52	54.38	66.02	**85.34**	**86.86**
Fisher Score	6	31	multiple_condition_count, halstead_level, cyclomatic_complexity, number_of_lines, loc_blank, halstead_difficulty	73.45	81.30	82.60	84.73	86.25
Without FS	All features		All features	79.55	82.05	84.10	83.20	85.34

Table VI
PERFORMANCE ACCURACY VALUES OF FILTER METHODS ON MW1 DATASETS

Method	**No. of selected features**	**Threshold value**	**Selected Features**	**Classifier**				
				K-nearest Neighbors	**Decision Tree**	**Random Forest**	**Naive Bayes**	**Multilayer Perceptron**
Gain Ratio	6	0.52	halstead_effort, halstead_prog_time, halstead_content, halstead_volume, halstead_difficulty, halstead_length	95.02	96.19	97.20	**97.51**	97.79
Information Gain	6	0.054	loc_comments, halstead_volume, edge_count, node_count, halstead_error_est, loc_blank	**95.57**	**96.75**	97.40	**97.51**	**97.83**
Relief	6	31	halstead_difficulty, halstead_error_est, essential_complexity, percent_comments, parameter_count, halstead_content	94.46	95.39	**97.65**	96.12	97.79
Chi-square	6	0.055	loc_comments, halstead_volume, num_unique_operands, edge_count, node_count, design_complexity	93.63	96.40	96.88	96.40	97.80
Fisher Score	6	31	condition_count, essential_complexity, branch_count, num_unique_operands, design_density decision_density	95.45	96.43	97.75	96.54	97.79
Without FS	All features		All features	95.43	96.61	97.58	95.58	97.72

[7] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 5, pp. 971–989, 2015.

[8] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[9] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[10] T. M. P. Hà and T. Q. H. Phan, "Nghiên cứu các kỹ thuật lựa chọn đặc trưng trong tập dữ liệu," *Hội thảo Khoa học quốc gia CITA 2020 lần thứ 9*, pp. 204–210, 2020.

[11] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles," in *International Workshop on Data Mining for Biomedical Applications*, pp. 106–115, Springer, 2006.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. USA: Wiley-Interscience, 2000.

[13] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," *Proceedings of the 21st Australasian Computer Science Conference ACSC'98*, pp. 181–191, 1998.

[14] J. Han, M. Kamber, and J. Pei, "3 - data preprocessing," in *Data Mining (Third Edition)* (J. Han, M. Kamber, and J. Pei, eds.), The Morgan Kaufmann Series in Data Management Systems, pp. 83–124, Boston: Morgan Kaufmann, third edition ed., 2012.

[15] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992* (D. Sleeman and P. Edwards, eds.), pp. 249–256, San Francisco (CA): Morgan Kaufmann, 1992.

[16] K. Gao, T. Khoshgoftaar, H. Wang, and N. Seliya, "Choosing software metrics for defect prediction: An investigation on feature selection techniques," *Softw., Pract. Exper.*, vol. 41, pp. 579–606, 04 2011.

[17] K. Muthukumaran, A. Rallapalli, and N. L. B. Murthy, "Impact of feature selection techniques on bug prediction models," in *Proceedings of the 8th India Software Engineering Conference*, ISEC '15, (New York, NY, USA), p. 120–129, Association for Computing Machinery, 2015.

[18] C. Catal and B. Diri, "A systematic review of software fault prediction studies," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7346–7354, 2009.

[19] H. Wang, T. Khoshgoftaar, and A. Napolitano, "A comparative study of ensemble feature selection techniques for software defect prediction," in *2010 Ninth International Conference on Machine Learning and Applications*, pp. 135–140, 12 2010.

[20] S. S. Rathore and A. Gupta, "A comparative study of feature-ranking and feature-subset selection techniques for improved fault prediction," in *Proceedings of the 7th India Software Engineering Conference*, ISEC '14, (New York, NY, USA), Association for Computing Machinery, 2014.

[21] Z. Xu, J. Liu, Z. Yang, G. An, and X. Jia, "The impact of feature selection on defect prediction performance: An empirical comparison," *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 309–320, 2016.

[22] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, (Arlington, Virginia, USA), p. 266–273, AUAI Press, 2011.

Ha Thi Minh Phuong received Bachelor degree in Information Technology from The University of Danang-University of Science and Technology in 2010. She earned M.Sc. degree in Computer Science at Yuan Ze University, Taiwan in 2013. She is a lecturer at The University of Danang - Vietnam-Korea University of Infomation and Communication Technology. She is currently pursuing the Ph.D. degree in Computer Science at the University of Danang. Her current research focuses are on software testing, deep learning.
Email: htmphuong@vku.udn.vn

Le Thi My Hanh is currently a lecturer of the Information Technology Faculty, University of Science and Technology, Danang, Vietnam. She gained M.Sc. degree in 2004 and the Ph.D. degree in Computer Science at the University of Danang in 2016. Her research interests are about software testing and more generally application of heuristic techniques to problems in software engineering.
Email: ltmhanh@dut.udn.vn

Nguyen Thanh Binh graduated in Information Technology from the University of Danang - University of Science and Technology in 1997. He received PhD. degree in Information Technology at Grenoble Institute of Technology, France in 2004. He has been qualified as Associate Professor since 2013. He is currently working at the University of Danang - Vietnam-Korea University of Information and Communication Technology. His research interests include software engineering and software quality.
Email: ntbinh@vku.udn.vn