An Empirical Study on the Equivalence and Stability of Feature Selection for Noisy Software Defect Data

Zhou Xu[1], Jin Liu[1]*, Zhen Xia[1], Peipei Yuan[2]
[1]State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan, China
[2]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China
*Corresponding author email: jinliu@whu.edu.cn

***Abstract*—Software Defect Data (SDD) are used to build defect prediction models for software quality assurance. Existing work employs feature selection to eliminate irrelevant features in the data to improve prediction performance. Previous studies have shown that different feature selection methods do not always yield similar prediction performance on SDD, which indicates that these methods are not equivalent. Also, previous studies have shown that SDD usually contains noise that may interfere the process of feature selection. In this work, we empirically investigate and measure the equivalence of different feature selection methods for SDD. Further, we intend to analyze the stability of the methods for noisy SDD. We perform statistical analyses on eight projects from NASA dataset with eight feature selection methods. For the equivalence analysis, we introduce Principal Component Analysis (PCA) and overlap index to qualitatively and quantitatively analyze the equivalence of these methods respectively. For the stability analysis, we apply consistency index to measure the stability of these methods. Experimental results indicate that different feature selection methods are indeed not equivalent to each other, and Correlation and Fisher Score methods achieve better stability.**

Keywords—defect data; feature selection; equivalence analysis; stability analysis;

I. INTRODUCTION

Software debugging is a key and expensive phase during the software development lifecycle. Defect identification is one of the main activities for software debugging [1], [2]. Software Defect Prediction (SDP) automatically detect the more defect-prone software modules (methods, classes or files) with software metrics (i.e., features) in Software Defect Data (SDD).

In the past decade, many researchers mainly focused on applying various data mining and machine learning algorithms to build defect prediction models on SDD for SDP, to identify the quality of a given software module by classifying it as defect-prone or not [3], [4], [5]. Defect prediction can help practitioners to reasonably allocate limited project resources to the potential defect-prone modules, and thus improve the efficiency and save the cost of software development.

One challenge in defect prediction modeling is the high dimensionality phenomenon, i.e. there may exist irrelevant or redundant features in SDD. Building prediction models with all features is unrealistic since these useless features may deteriorate the performance of prediction models. Thus, the issue of how to select the most appropriate features is of great importance.

Nowadays, a plenty of feature selection methods have been proposed in data mining area. Various methods have been successfully introduced to assist the selection of a feature subset that could benefit the defect prediction process on SDD. Previous studies have shown that diverse feature selection methods yield quite different performance on prediction models for SDD [6], [7], which implies that different methods might be not equivalent, that is, different methods would identify different set of features as relevant. However, to the best of our knowledge, no previous studies proposed a method to investigate the equivalence of different feature selection methods. In this paper, we conduct an empirical study to qualitatively analyze whether different feature selection methods are equivalent with Principal Component Analysis (PCA) technique. Further, we use overlap index to quantitatively analyze to what extent different feature selection methods are not equivalent each other.

Meanwhile, due to some unexpected reasons, such as improper software data collection and recording process [8], there may exist noise in SDD. One potential challenge for feature selection methods is their sensitivity to the noise in SDD (i.e. the impact of noise on the selection of relevant features from SDD), which is defined as the stability of the feature selection methods to the noise on SDD in this paper. It is valuable to study the stability of different feature selection methods, because if the features selected by a specific method vary when there exists noise in the dataset, it is difficult to say whether these features are the most representative ones. If a feature selection method is high in stability, it will enable practitioners to select the representative features of SDD before cleaning the dataset. However, there are few relevant studies on this issue yet. In this work, we employ consistency index to measure the stability of feature selection methods on noisy SDD.

Statistical analyses on the data from eight projects of NASA dataset confirm that different feature selection methods are not equivalent to each other. The analytic results suggest that Correlation (Cor) and Fisher Score (FS) methods are more stable.

Our main contributions are highlighted as follows:

(1) We introduce Principal Component Analysis (PCA) technique to investigate the equivalence of different feature selection methods. To the best of our knowledge, this is the first empirical study to qualitatively analyze the equivalence of feature selection methods.

(2) We employ overlap index to measure to what extent these feature selection methods are not equivalent to each other.

(3) We analyze the stability of feature selection methods in the context of noisy SDD and identify the stable ones.

DOI reference number: 10.18293/SEKE2017-097

II. Related Work

A. Feature Selection on SDD

Many previous studies have investigated the effect of feature selection on the performance of defect prediction models. Song et al. [3] pointed out that feature selection is an indispensable part of a general defect prediction framework. Shivaji et al. [9] investigate the impact of six methods on the classification-based bug prediction. They found that selecting about 10% features could achieve a satisfactory performance. He et al. [21] suggested that prediction models built with a simplified feature set could achieve acceptable performance for defect prediction.

All these studies only focus on evaluating different feature selection methods using prediction performance indicators, none of them pay attention to the equivalence between these methods on their suggested features. In this work, we conduct an empirical study to investigate this issue and also evaluate feature selection methods from the perspective of stability.

B. Stability of Feature Selection for SDD

Recently, a few studies evaluated feature selection methods by their stability for SDD. The stability of a method is defined as the degree of consensus of a feature subset pair selected by the method on two variants of a given dataset obtained by sampling. It is worth studying the stability of feature selection methods since the method that tends to select the highly similar features despite changes in the data could be more trustworthy.

To the best of our knowledge, the earliest study on the stability of the feature selection methods for SDD was performed by Gao et al. [26]. They empirically studied the impact of three data sampling techniques on the stability of six filter-based feature selection methods for SDD. Wang et al. [22] investigated the impact of the dataset perturbations (i.e. randomly removing a certain proportion software modules) and the number of selected features on 6 filter-based feature selection methods for SDD.

Different from our empirical study, these literatures aim to explore the stability of feature selection methods to the data perturbation on SDD, while our work focuses on investigating the stability of the methods to the noise on SDD. To the best of our knowledge, this is the first work to investigate this issue.

III. Preliminary and Analysis Method

In this work, we conduct an empirical study to investigate two issues: the equivalence analysis to explore whether different feature selection methods select similar features for SDD and stability analysis to investigate the stability of different feature selection methods for noisy SDD.

A. Feature Ranking Methods

Feature selection is a critical data preprocessing technique in many fields. In general, the feature selection methods fall into two major categories as feature ranking and feature subset selection [10], [25]. The feature ranking methods have gained favor due to its simplicity and efficiency. In this paper, we empirically study the equivalence and stability of feature ranking methods for noisy SDD. The eight methods used in this work are Chi-Square (**CS**), Correlation (**Cor**), Information Gain (**IG**), Symmetrical Uncertainty (**SU**), Fisher Score (**FS**), Welch T-Statistic (**WTS**), ReliefF (**RF**), One Rule (**OneR**). The reason why we choose these methods is that they are widely used in defect prediction and belong to different feature selection families [28], [30]. CS is a statistic-based method, Cor is a correlation-based method, IG and SU are entropy-based methods, FS and WTS are first order statistics-based methods, RF is a instance-based, OneR is a classifier-based method. The detailed description of these methods are available in [11], [29].

B. Research Questions

To reveal the answers of the two issues, we empirically study the following three research questions (RQs).

RQ1: Are different feature selection methods equivalent to each other when being applied to SDD?

RQ2: To what extent different feature selection methods are equivalent to each other?

RQ3: Which feature selection method is more stable among the six methods for noisy SDD given in this study?

C. Equivalence Analysis

The first two research questions aim to conduct the equivalence analysis of feature selection methods. In this work, we introduce PCA technique to qualitatively analyze the equivalence of the eight methods (***RQ1***). Further, we apply the overlap index to quantitatively analyze to what extent these methods are equivalent to each other (***RQ2***).

1) PCA

We assume that given n evaluated objects (i.e. n features in this work), let $x_1, x_2, \dots, x_m$ denote the m indicator variables (i.e. m different feature selection methods in this work) and $x_{ij}, (i = 1,2, \dots, n; j = 1,2, \dots, m)$ denotes the score of the jth evaluated object with the ith indicator variable, namely the relevant score of the ith feature assigned by the jth feature selection method. PCA technique involves four steps:

a) Normalization

Considering the difference in score values assigned by different indicator variables, we apply the z-score to normalize the values in this step. The x_{ij} can be transformed to $\tilde{x}_{ij}$ as:

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \tag{1}$$

where $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$, $s_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}$.

b) Correlation coefficient matrix

Let $R = (r_{ij})_{m\times m}$ denotes the correlation coefficient matrix with respect to the indicator variables, where r_{ij} denotes the correlation coefficient between the ith and the jth indicator variables. r_{ij} $(i, j = 1,2, \dots, m)$ is defined as:

$$r_{ij} = \frac{\sum_{k=1}^{n} \tilde{x}_{ki}\tilde{x}_{kj}}{n-1} = \frac{\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)^2 \cdot \sum_{k=1}^{n}(x_{kj} - \bar{x}_j)^2}} \tag{2}$$

c) Eigenvalues and eigenvectors

In this step, we calculate the eigenvalues and eigenvectors of the correlation coefficient matrix R. By solving the determinant $|R - \lambda I| = 0$ (I is identity matrix), we can obtain m eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$,$(\lambda_1 \geq \lambda_2 \geq, \dots, \geq \lambda_m \geq 0)$ and the corresponding eigenvectors $\alpha_1, \alpha_2, \dots, \alpha_m$, where $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{im}]^T (i = 1,2, \dots, m)$. Then the original indicator variables are projected to m new orthogonal variables as:

$$\begin{cases} y_1 = \alpha_{11}x_1 + \alpha_{12}x_2 + \cdots + \alpha_{1m}x_m \\ y_2 = \alpha_{21}x_1 + \alpha_{22}x_2 + \cdots + \alpha_{2m}x_m \\ \vdots \\ y_m = \alpha_{m1}x_1 + \alpha_{m2}x_2 + \cdots + \alpha_{mm}x_m \end{cases} \tag{3}$$

where $\alpha_{k1}^2 + \alpha_{k2}^2 + \cdots + \alpha_{km}^2 = 1$ ($k = 1,2, \ldots, m$), y_i denotes the i th principal component, and α_{ij} denotes the correlation coefficient between the original variable x_j and the principal component y_i. The coefficient α_{ij} represents the contribution of the original variables x_j to the principal component y_i, i.e., the importance of x_j to y_i.

In this work, we use this coefficient to measure the correlation between different feature selection methods and the principal components. A greater absolute value of the correlation coefficient α_{ij} indicates that the j th original variable captures the ith principal component well. If different feature selection methods capture same principal components, it indicates that these methods are equivalent to each other.

d) Contribution percentage

Let c_i denotes the percentage of the variance of principal component y_i in total variances, which is also called the contribution percentage of the principal component y_i to original variables [12]. It is defined as:

$$c_i = \frac{\lambda_i}{\sum_{k=1}^{m} \lambda_k} \tag{4}$$

where λ_i denotes the i th eigenvalue of the correlation coefficient matrix R. The contribution percentages reflect the synthesis or explanatory ability of the principal components towards the original variables. The first principal component is the largest and the rest can be deduced by analogy.

Then the cumulative contribution percentage cc_p of the top p principal components is defined as:

$$cc_p = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{j=1}^{m} \lambda_j} \tag{5}$$

where $p \leq m$. This index reflects the synthesis ability of the top p principal components towards the original variables.

In this work, we use the contribution percentage to measure the percentage (or the amount of information) of the original variables covered by different principal component.

2) Overlap Index

The analysis based on PCA enables us to test whether different feature selection methods are equivalent or not. It is unknown yet to what extent these methods are equivalent to each other. We further use an indicator, called overlap index [24], to measure the extent of equivalence between any method pair.

In this work, we apply overlap index to quantify the extent of the equivalence under a cut point μ. Note that μ is actually a pre-specified percentage of selected features. Given a specific cut point μ, we compute the following overlap index as:

$$|(A \cap B)_\mu| = |D(A) \cap D(B)|_\mu \tag{6}$$

where $D(A)$ and $D(B)$ denote the top μ relevant features selected by feature selection method A and B for a specific dataset D, respectively. Under the cut point μ, $|(A \cap B)_\mu|$ denotes the cardinal number of features selected by both methods. In general, a higher overlap value indicates a larger extent of equivalence of the two methods under a given cut point μ. In this work ,we empirically set μ as 15%, 20%, 25%, 30%, 35% and 40%.

D. Stability Analysis

A potential threat for feature selection methods is the presence of noise in the data. It would be valuable to study of the stability of different feature selection methods for noisy dataset. If different feature selection methods are not equivalent to each other, the identification of the most stable method will benefit the selection of the high representative features of the noisy data without cleaning it.

In this work, we further conduct an empirical study to investigate the stability of the eight methods and attempt to find out the more stable ones for noisy SDD (***RQ3***). Specifically, we use an indicator to compute the similarity of the relevant feature subset pair selected from the noisy and clean versions of a dataset with a feature selection method.

Previous studies have proposed different similarity metrics to measure the stability of feature selection methods [13]. In this study, we employ consistency index [14]. Given a feature selection method A, dataset D and cut point μ, let U_1 and U_2 denotes the feature subsets selected by method A from noisy and clean versions of D, respectively. Then consistency index $IC(A)_{D_\mu}$ is defined as:

$$IC(A)_{D_\mu} = \frac{dn - t2}{t(t-n)} \tag{7}$$

where d denotes the cardinality of the intersection of U_1 and U_2, n denotes the total number of the features in D, t denotes the cardinality of U_1 or U_2, i.e., the number of selected features, and $-1 < IC(A)_{D_\mu} \leq 1$. Consistency index have been widely used to measure the stability of feature selection methods to the random perturbation on the dataset [15], [16].

IV. BENCHMARK DATASET

To investigate the equivalence and the stability of the feature selection methods for noisy SDD, we used eight original version projects of NASA dataset and the corresponding clean version preprocessed by Shepperd et al. [17] as our experimental dataset. NASA dataset is a method-level software defect dataset that is characterized by static code metrics [5]. The original NASA dataset is known to be noisy, to alleviate the data quality of this dataset, Shepperd et al. applied some preprocessing criteria to clean the dataset. We remove the features with one value. Table I presents the details of the two versions of the eight projects, including the number of features (#F), modules (#M) defective modules (#D) and the percentage of defective modules (%D).

TABLE I. DESTRIPTION OF THE TWO VERSIONS OF NASA DATASET

Projects	#F	Noise			Clean		
		#M	#D	%D	#M	#D	%D
CM1	37	505	48	9.5%	327	42	12.8%
KC1	21	2107	325	15.4%	1162	294	25.3%
KC3	39	458	43	9.4%	194	36	18.6%
MC1	38	9466	68	0.7%	1847	36	2.0%
MC2	39	161	52	32.3%	125	44	35.2%
MW1	37	403	31	7.7%	251	25	10.0%
PC1	36	5589	23	0.4%	734	16	2.2%
PC5	38	17186	516	3.0%	1679	459	27.3%

V. ANALYSIS RESULTS

A. *RQ1*

To answer this question, we conduct the analyses with the eight feature selection methods on the data from eight projects of original NASA dataset with PCA technique. Owing to space constraint, Table II and III only report the analytic results on KC3 and PC5 projects randomly selecting from the eight studied projects. The Ci ($i = 1, \ldots, 8$) denotes the i th principal

TABLE II. RESULTS OF PCA ON KC3 PROJECT

Methods	C1	C2	C3	C4	C5	C6	C7	C8
CS	**0.40**	-0.27	0.12	-0.23	0.51	-0.62	-0.22	0.03
Cor	**0.42**	-0.13	-0.09	0.28	-0.32	-0.05	0.10	0.78
IG	**0.42**	-0.20	-0.14	-0.13	0.33	0.75	-0.29	-0.01
SU	**0.41**	-0.19	-0.21	-0.24	-0.16	-0.01	0.73	-0.36
FS	**0.41**	0.00	0.08	0.48	-0.40	-0.11	-0.39	-0.51
WTS	0.25	**0.55**	0.42	0.38	0.43	0.09	0.34	0.01
RF	-0.27	-0.50	-0.29	**0.65**	0.36	-0.02	0.19	-0.08
OneR	-0.10	-0.53	**0.80**	-0.03	-0.15	0.17	0.11	0.01
CP(%)	61.19	18.04	10.37	5.60	2.33	0.98	0.82	0.67

TABLE III. RESULTS OF PCA ON PC5 PROJECT

Methods	C1	C2	C3	C4	C5	C6	C7	C8
CS	0.34	0.55	0.07	0.07	0.27	0.01	-0.06	0.70
Cor	**0.49**	-0.06	-0.25	-0.22	-0.30	-0.15	0.73	0.03
IG	0.28	**0.63**	0.05	-0.05	0.14	0.15	-0.01	-0.69
SU	0.33	-0.07	-0.05	**0.86**	-0.35	0.04	-0.12	-0.06
FS	**0.46**	-0.13	-0.28	-0.40	-0.26	-0.16	-0.67	0.00
WTS	0.30	-0.26	**0.59**	-0.19	-0.12	0.66	0.01	0.06
RF	-0.32	0.27	-0.52	-0.07	-0.39	0.61	-0.01	0.14
OneR	0.24	-0.36	-0.48	0.11	0.67	0.34	0.03	-0.06
CP(%)	42.82	22.26	14.72	9.02	7.24	3.55	0.27	0.13

component, the CP denotes the contribution percentage while the other values represent the correlation coefficients. For the CP values, from the complete PCA results on all projects (see in [32]), we observe that PCA technique can mainly identify four principal components with a cumulative contribution percentage between 88.82% (for PC5) and 97.45% (for KC1) to capture the most information of the original variables (i.e., feature set in this study). The first principal component contributes the major proportion varying from 42.73% to 74.67%, and the second principal component contributes the proportion varying from 12.12% to 22.26%, while the third and the forth components contribute proportion varying from 6.06% to 18.27% and from 3.28% to 7.03%, respectively.

For the correlation coefficients between the methods and the principal components, the values in bold highlight the feature selection methods that best represent the corresponding principal components. Note that the negative coefficients only denote negative correlation without the meaning of numerical size. From Table II, we observe that, for KC3 project, the first component, which reflects 61.19% amount of information towards the original feature set, is mainly captured by CS, Cor, IG, SU, and FS (since their coefficient values are very close) while poorly captured by the other methods; the second principal component is only captured by WTS; the third principal component is mainly captured by OneR as the coefficient value 0.8 and the fourth principal component is mainly captured by RF as the coefficient value 0.65. Table III shows that, for PC5 project, the first component is mainly captured by Cor and FS while poorly captured by the others; the second to the fourth principal component are captured by IG, SU, WTS, respectively. For other projects (see in [32]), similar observations are obtained except that the methods that capture each component vary among different projects.

As mentioned above, the analytic results of the PCA indicate that different feature selection methods indeed capture different components. It confirms that different feature selection methods do not have the similar effect on selecting the relevant features, i.e., they assign different relevance proneness to the module features. Besides, for the methods that belong to the same feature selection family (i.e., IG and SU, FS and WTS), they are not always capture the same principal component. For example, on KC3 project, FS captures the first principal component while WTS captures the second component; on PC5 project, IG and SU capture the second and the fourth principal component respectively while FS and WTS capture the first and the third component respectively. It denotes that the feature selection methods that belong to the same family are not equivalent to select the similar relevant features. In addition, we find that Cor and FS capture the same principal component on most projects.

> We conclude that, in general, different feature selection methods be not equivalent to each other for the given SDD in this study.

B. *RQ2*

Although different feature selection methods are testified to be not equivalent to each other by PCA in Section V-A, but it could not tell us to what extent they are equivalent to each other. This question is actually to supplement this issue using the overlap index.

We also only report the analysis results on PC5 project in Table IV due to the space limit (see in [32] for complete results). The last line of the table reports the average percentage (AP) of the overlap index. For example, for CS and Cor pair, the overlap value is equal to 1 while the number of the selected features is equal to 6 under the cut point 15%, so the percentage of the overlap is equal to 16.7% (1/6). From the table, we observe that the overlap values of most method pairs are relative low. For example, most overlap values are equal to 0 under the cut point 15%. In addition, most overlap percentages are less than 50% under four cut points. But for the method pair Cor and FS, the overlap values are nearly equal to the total number of selected

TABLE IV. OVERLAP RESULTS ON PC5 PROJECT

A	B	15%	20%	25%	30%	35%	40%
CS	Cor	1	1	2	2	3	6
CS	IG	3	5	8	11	13	14
CS	SU	0	0	0	1	1	3
CS	FS	1	1	2	2	3	6
CS	WTS	1	1	2	2	5	7
CS	RF	0	0	0	2	4	7
CS	OneR	0	0	0	0	2	6
Cor	IG	0	0	1	2	4	5
Cor	SU	0	0	2	5	6	8
Cor	FS	6	8	10	12	14	16
Cor	WTS	3	3	6	6	8	12
Cor	RF	1	2	3	4	5	7
Cor	OneR	3	4	5	7	8	11
IG	SU	0	0	0	1	1	2
IG	FS	0	0	1	2	4	5
IG	WTS	0	1	1	2	5	6
IG	RF	0	0	0	2	4	6
IG	OneR	0	0	0	0	1	4
SU	FS	0	0	2	5	6	8
SU	WTS	0	0	2	3	4	6
SU	RF	0	0	0	0	3	3
SU	OneR	0	3	6	8	9	10
FS	WTS	3	3	6	6	8	12
FS	RF	1	2	3	4	5	7
FS	OneR	3	4	5	7	8	11
WTS	RF	3	4	4	5	8	9
WTS	OneR	1	1	2	5	5	9
RF	OneR	1	1	1	1	2	4
AP(%)		18.5	19.6	26.4	31.8	38.0	46.9

features under all cut points. This observation accords to that in ***RQ1*** which shows that the Cor and FS usually capture the same principal component. For other projects (see in [32]), we can also get the similar observations.

Moreover, from the complete results in [32], we find that the overlap percentage increases as the cut point value increases on most cases. The reason may be that a feature selection method pair tends to select more common features as the relevant ones when the cut point increases.

> To sum up, the analytic results show that the overlap values and their average percentages of method pairs are low on most case. This observation also confirms that different feature selection methods are quite not equivalent to each other.

C. RQ3

Since different feature selection methods are indeed not equivalent (as shown in ***RQ1*** and ***RQ2***) and the noise is inevitable in SDD, in this question, we are particularly interested in investigating the stability of feature selection methods for noisy SDD. As the first work to empirically study this issue, we conduct the analyses with the eight feature selection methods on the two versions of eight NASA projects. We apply consistency index to measure the stability of these methods.

TABLE V. AVERAGE STABILITY ON EACH PROJECT UNDER SIX CUT POINTS

Cutoff	CS	Cor	IG	SU	FS	WTS	RF	OneR
15%	0.54	0.83	0.44	0.38	0.83	0.67	0.66	0.50
20%	0.55	0.80	0.47	0.33	0.80	0.71	0.74	0.42
25%	0.52	0.85	0.54	0.35	0.83	0.67	0.78	0.50
30%	0.50	0.84	0.58	0.37	0.84	0.72	0.70	0.51
35%	0.59	0.85	0.56	0.41	0.83	0.73	0.72	0.55
40%	0.60	0.80	0.53	0.40	0.77	0.79	0.76	0.58

Table V summaries the average stability values of each method across all projects under each cut point (detailed results is available in [32]). We observe that Cor and FS methods can achieve better results under all cut points, and their values are very close. It is also consistent with the observations in ***RQ1*** and ***RQ2*** that Cor and FS select very similar features. In addition, WTS and RF also achieve competitive results under the cut point 40%.

TABLE VI. P-VALUES AND AVERAGE RANKINGS UNDER EACH CUT POINTS

Cutoff	p-values	CS	Cor	IG	SU	FS	WTS	RF	OneR
15%	0.0057	4.94	2.75	5.50	6.25	2.31	4.38	4.31	5.56
20%	0.0002	4.69	2.69	5.81	6.56	2.44	3.63	3.63	6.56
25%	0.0012	5.56	2.38	5.69	6.69	2.88	4.31	3.13	5.38
30%	0.0004	6.06	2.63	4.81	6.56	2.63	3.19	4.25	5.88
35%	0.0016	5.00	2.88	5.06	6.81	2.38	3.38	4.63	5.88
40%	0.0080	4.94	3.25	5.38	6.94	3.56	3.06	3.50	5.38

To statistically analyze the stability results under each cut point, we perform Friedman test, a non-parametric test, to compare the eight methods over the eight projects by ranking each method on each project separately. The p-value less than 0.05 in Friedman test indicates that the average rankings of these methods are statistically significant. We further apply Nemenyi test as a post-hoc test to all pairs of methods to determine which method performs statistically different.

Table VI reports the p-values of Friedman test and the average ranking of each method across all projects under each cut point. The p-values of Friedman test in the table are all less than 0.05, which indicates that the differences among the average rankings of these methods are significant under each cut point. In addition, the lower average ranking value represents the higher stability. We observe that Cor and FS usually obtain better ranking values. Fig.1 visualizes the results of Nemenyi test in terms of stability index. In the figure, the average rankings of the methods are plotted on the top line in each subfigure under each cut point. Methods that are not statistically significant are connected with blue lines. The lower average ranking locates on the left side of the axis.

This figure depicts that the stability results between FS and SU under cut point 15%, between FS, Cor and OneR, SU under cut point 20%, between Cor, FS and SU under cut point 25%, 30% and 35%, between WTS and SU under cut point 40% are statistically significant, respectively. In other cases, the differences are not significant. In addition, Cor and FS can always achieve best rankings except under cut points 40% while WTS achieves the best ranking under this cut point. This conclusion is quite different from that in [22] and [23], which indicate that the RF method is the most stable method. The reason is that we focus on the stability of feature selection methods for noisy SDD, while the studies [22], [23] aim at the stability of the methods to data perturbation for SDD.

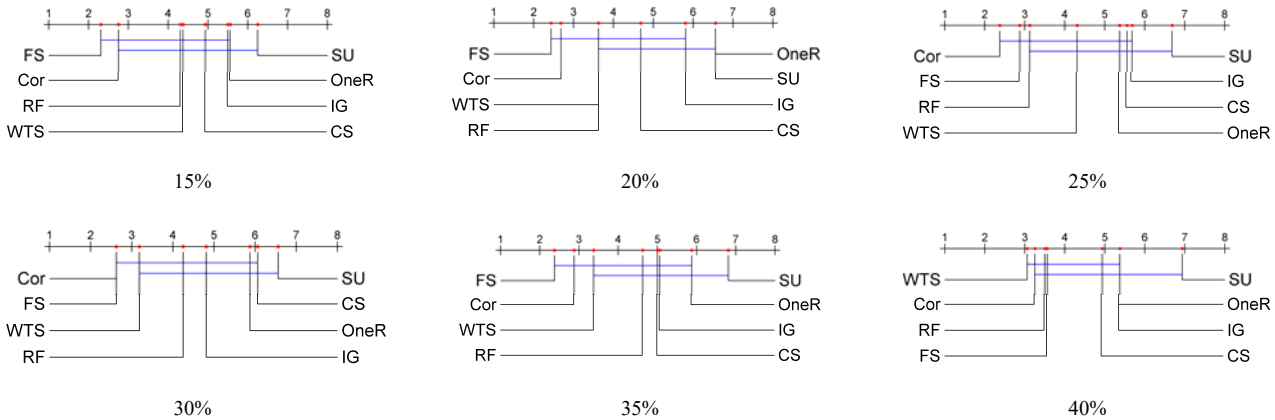

Fig.1. Comparison of all methods against each other in stability. Methods that are not significant different (the Nemenyi test, at p=0.05) are connected.

With the evidence provided by the above activities, we conclude that Cor and FS can achieve the most stable results for the noisy SDD on most cases.

VI. THREATS TO VALIDITY

For the generalization of our results, we carefully chose the NASA dataset which is commonly used in previous studies in software engineering domain [4], [17], [18], [19], [20]. Besides, previous work also conducted case studies on NASA dataset to investigate the effect of noise on SDD [8], [27]. So using NASA dataset can make our results more comparable and persuasive. For the bias in the choice of feature selection methods, as the first work to empirical study the equivalence and stability of feature selection methods for noisy SDD, we just select some typical methods for SDD. For the evaluation metrics, overlap index is reasonable to measure the extent of equivalence of feature selection methods since it has been used to evaluate the extent of equivalence of different machine learning techniques for defect prediction [24]. In addition, consistency index has been widely used to analyze the stability of feature selection methods for SDD in terms of data perturbations [15], [16], [31].

VII. CONCLUSION AND FUTURE WORK

This paper reports an empirical study of feature selection methods on SDD. This study involves two aspects: an equivalence analysis and a stability analysis. For the equivalence analysis, we raised the issue of the equivalence of different feature selection methods, that is, whether the methods have the similar effect to select relevant features. Analytic results by PCA and overlap index on eight projects of NASA dataset show that the studied methods are usually not equivalent to each other. The stability analysis with consistency index indicate that Cor and FS achieve better stability for noisy SDD.

In the future, we plan to take more feature selection methods, including feature subset selection methods, as our research objects. Meanwhile, we would explore the effect of different noise levels on the stability of feature selection methods for SDD.

ACKNOWLEDGEMENT

This work is partly supported by the grants of National Natural Science Foundation of China (No.61572374, No.U163620068, No.U1135005) and the Academic Team Building Plan from Wuhan University and National Science Foundation (NSF) (No. DGE-1522883).

REFERENCES

[1] T. Britton, L. Jeng, G. Carver, P. Cheak, T. Katzenellenbogen, Reversible Debugging Software. *Tech. Rep*., University of Cambridge, Judge Business School, 2013.

[2] C. Parnin, A. Orso, Are automated debugging techniques actually helping programmers? in: Proceedings of the 2011 *International Symposium on Software Testing and Analysis,(ISSTA)*, New York, NY, USA, pp.199–209, 2011.

[3] Q. Song, Z. Jia, M. Shepperd, S. Ying, and J. Liu. A general software defect-proneness prediction framework. *IEEE Transactions on Software Engineering*, 37(3): 356-370, 2011.

[4] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch. Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Transactions on Software Engineering*, 34(4): 485-496, 2008.

[5] T. Menzies, J. Greenwald, and A. Frank. Data mining static code attributes to learn defect predictors. *IEEE Transactions on Software Engineering*, 33(1): 2-13, 2007.

[6] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 1437-1447, 2003.

[7] K. Gao, T. M. Khoshgoftaar, H. Wang, and N. Seliya. Choosing software metrics for defect prediction: an investigation on feature selection techniques. *Software: Practice and Experience*, 41(5): 579-606, 2011.

[8] C. Catal, O. Alan, and K. Balkan. Class noise detection based on software metrics and ROC curves. *Information Sciences*, 181(21): 4867-4877, 2011.

[9] S. Shivaji, E. J. Whitehead, R. Akella, and S. Kim. Reducing features to improve code change-based bug prediction. *IEEE Transactions on Software Engineering*, 39(4): 552-569, 2013.

[10] 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*. ACM, 7, 2014.

[11] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1): 63-90, 1993.

[12] A. K. Dubey and V. Yadava. Multi-objective optimization of Nd:YAG laser cutting of nickel-based superalloy sheet using orthogonal array with principal component analysis. *Optics and Lasers in Engineering*, 46(2):124-132, 2008.

[13] T. M. Khoshgoftaar, A. Fazelpour, H. Wang, and R. Wald. A survey of stability analysis of feature subset selection techniques. In *Proceedings of the 14th International Conference on Information Reuse and Integration (IRI)*. IEEE, 424-431, 2013.

[14] R. Real and J. M. Vargas. The probabilistic basis of the Consistency's index of similarity. *Systematic biology*, 45(3): 380-385, 1996.

[15] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms. *International Conference on Data Mining (ICDM)*, 218-225, 2005.

[16] S. Alelyani, Z. Zhao, and H. Liu. A dilemma in assessing stability of feature selection algorithms. In *Proceedings of the 13th International Conference on High Performance Computing and Communications (HPCC)*. 701-707, 2011.

[17] M. Shepperd, Q. Song, Z. Sun, and C. Mair. Data quality: Some comments on the NASA software defect datasets. *IEEE Transactions on Software Engineering*, 39(9): 1208-1215, 2013.

[18] Y. C. Liu, T. M. Khoshgoftaar, and N. Seliya. Evolutionary optimization of software quality modeling with multiple repositories. *IEEE Transactions on Software Engineering*, 36(6): 852-864, 2010.

[19] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto. Automated parameter optimization of classification techniques for defect prediction models. *ICSE*, 321-332, 2016.

[20] J. Nam and S. Kim. Heterogeneous defect prediction. In *Proceedings of the 10th Joint Meeting on Foundations of Software Engineering*. 508-519, 2015.

[21] P. He, B. Li, X. Liu, J. Chen, and Y. Ma. An empirical study on software defect prediction with a simplified metric set. *Information and Software Technology*, 59: 170-190, 2015.

[22] H. Wang, T. M. Khoshgoftaar, and R. Wald. Measuring robustness of feature selection techniques on software engineering datasets. In *Proceedings of the 12th International Conference on Information Reuse and Integration (IRI)*. IEEE, 309-314, 2011.

[23] H. Wang, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. A novel dataset-similarity-aware approach for evaluating stability of software metric selection techniques. In *Proceedings of the 13th International Conference on Information Reuse and Integration (IRI)*. IEEE, 1-8, 2012.

[24] A. Panichella, R. Oliveto, and A. De Lucia. Cross-project defect prediction models: L'union fait la force. In *Proceedings of the Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE)*, 164-173, 2014.

[25] T. M. Khoshgoftaar, K. Gao, and A. Napolitano. An empirical study of feature ranking techniques for software quality prediction. *International Journal of Software Engineering and Knowledge Engineering*, 22(02): 161-183, 2012.

[26] K. Gao, T. M. Khoshgoftaar, and A. Napolitano. Impact of data sampling on stability of feature selection for Software Defect Data. *In Proceedings of the 23rd International Conference on Tools with Artificial Intelligence*. IEEE, 1004-1011, 2011.

[27] J. Van Hulse and T. Khoshgoftaar. Knowledge discovery from imbalanced and noisy data. *Data and Knowledge Engineering*, 68(12): 1513-1542, 2009.

[28] Shivaji S. Efficient bug prediction and fix suggestions[J]. Dissertations & Theses - Gradworks, 2013.

[29] Z. Xu, J. Liu, Z. Yang, and X.Jia. The Impact of Feature Selection on Defect Prediction Performance: An Empirical Comparison. *International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 309-320, 2016.

[30] K. Gao, T. M. Khoshgoftaar, H. Wang, and N Seliya. Choosing software metrics for defect prediction: an investigation on feature selection techniques. *Software-practice & Experience*, 41(5):579-606, 2011.

[31] H. Wang, T. M. Khoshgoftaar, and R. Wald. A Study on First Order Statistics-Based Feature Selection Techniques on Software Metric Data. *SEKE*, 7-472, 2013.

[32] Z. Xu, J. Liu, and Z. Xia. Complete experimental results. [Online]. http://pan.baidu.com/s/1mic7ep6