Stability of Three Forms of Feature Selection Methods on Software Engineering Data

Huanjing Wang
Western Kentucky University
Bowling Green, Kentucky 42101
huanjing.wang@wku.edu

Taghi M. Khoshgoftaar
Florida Atlantic University
Boca Raton, Florida 33431
khoshgof@fau.edu

Amri Napolitano
Florida Atlantic University
Boca Raton, Florida 33431
amrifau@gmail.com

***Abstract*—One of the major challenges when working with software metrics datasets is that some metrics may be redundant or irrelevant to software defect prediction. This may be addressed using feature (metric) selection, which chooses an appropriate subset of features for use in downstream computation. There are three major forms of feature selection: filter-based feature rankers, which uses statistical measures to assign a score to each feature and present the user with a ranked list; filter-based subset evaluation, which uses statistical measures on feature subsets to find the best choice; and wrapper-based subset selection, which builds classification models using different subsets to find the one which maximizes performance. Software practitioners are interested in which feature selection methods are best at providing the most stable feature subset in the face of changes to the data (here, the addition or removal of instances). In this study we select feature subsets using fifteen feature selection methods and then use our newly proposed Average Pairwise Tanimoto Index (APTI) to evaluate the stability of feature selection methods. We evaluate the stability of feature selection methods on a pair of subsamples generated by fixed-overlap partitions algorithm. Four different levels of overlap are considered in this study. Four software metric datasets from a real-world software project are used in this study. Results demonstrate that ReliefF (RF) is the most stable feature selection method and wrapper based feature subset selection shows least stability. In addition, as the overlap of partitions increased, the stability of the feature selection strategies increased.**

I. Introduction

For most software systems, superfluous software metrics (e.g., number of loops, number of global variables, and number of exit nodes) are often collected during the software development cycle. Some metrics may be redundant or irrelevant to software defect prediction. Therefore the identification and selection of of a small set of relevant features from a metric dataset could be used by software developers to guide their efforts to reduce software development cost and produce more reliable software systems [1]. The identification and selection process is called metric (feature) selection. Feature selection algorithms (which select a subset of features from the original dataset) are often used to reduce the original feature set down to a subset containing only the most important features. Numerous feature selection methods have been proposed in the data mining and software engineering domains.

While feature selection is a necessary step, very little work has focused on the robustness (stability) of the feature selection methods in regards to software metrics data. The purpose of studying the stability of a feature selection technique is to determine which technique provides the feature subset that is the most robust to changes in the data. In this study, we used a fixed-overlap partitions algorithm which was proposed by our research group to generate a pair of subsamples which have same number of instances and a specified degree of overlap (fraction of instances in common). Then feature subset chosen from the pair of subsamples using a feature selection method are compared. The proposed algorithm is different from the approaches used by most researchers, which either generate multiple random subsamples of the original dataset and compare the features chosen from these with one another, or compare the features from the subsamples directly with the features from the original data. It can be noted that the first approach can not control the overlap between subsamples, while the second approach compare features from different size of datasets.

(DOI reference number: 10.18293/SEKE2015!-198)

The primary focus of this paper is to evaluate the stability of fifteen feature selection methods through a case study of four consecutive releases of a very large telecommunications software system (denoted as LLTS). We consider fifteen different feature selection strategies: three feature rankers each coupled with three feature subset sizes, the Correlation-based Feature Selection (CFS) filter-based subset evaluator, and wrapper-based feature selection using one of five different learners inside the wrapper. We evaluate the stability of feature selection methods on a pair of subsamples generated by the fixed-overlap partitions algorithm. Four different levels of overlap are considered in this study. We find that in general, the most stable feature selection methods is a ranker-based approach. Among the rankers, ReliefF (RF) is the most stable one. The trends of rankers being the most stable feature selection overall, CFS often being a moderate stable feature selection, and wrappers being extremely poor choices of feature selection in terms of stability were all present in the results. In addition, as the overlap of partitions increased, the stability of the feature selection strategies increased.

The rest of the paper is organized as follows. We review relevant literature on feature selection and stability in Section II. Section III provides detailed information about the three classes of feature selection, fixed-overlap partitions , and metrics used for measuring stability (including our newly-proposed extension on the Tanimoto Index) used in our study. Section IV provides a description of the software measurement datasets used and presents empirical results of our study. Finally, in Section V, the conclusion is presented and the suggestions for future work are indicated.

II. Related Work

Feature selection is a necessary step in data mining. The main goal of feature selection is to select a subset of features that minimizes the prediction errors of classifiers. A number of papers have studied the use of feature selection techniques as a data preprocesing step. While many works have focused on the performance of models built using features selected by feature selection techniques, another way to evaluate a feature selection technique is through stability. Few studies exist on the stability of feature selection algorithms, but a

small number of studies have considered the stability of wrapper-based feature selection, which is often calculated in a similar fashion. Somol and Novovičová [2] conducted a comprehensive study of the stability of feature selection techniques and investigated the problem of evaluating the stability of feature selection techniques that produce subsets of varying size. They compared the stability of three wrapper techniques (Gaussian classifier, 3-Nearest Neighbor, and Support Vector Machines). Lustgarten et al. [3] proposed a stability measure called the Adjusted Stability Measure (ASM, based upon extending the consistency index to varying feature subset size), as opposed to Unadjusted Stability Measure (USM, based on the Jaccard index), that computes robustness of a feature selection technique with respect to random feature selection. They compared the stability of three wrapper approaches. Haury et al. [4] evaluated a number of feature ranking methods and one wrapper-based subset evaluation technique and considered stability in terms of how many features are in common between two subsets generated from independent subsamples of the original data. Dunne et al. [5] considered wrappers using a 3-nearest neighbor learner and three choices of search technique, evaluating stability by resampling the original dataset and finding the Hamming distance between the various feature subset masks. The overall stability is then defined by the average Normalized Hamming Distance. Kalousis et al. [6] used the Tanimoto coefficient that is a generalized version of the Jaccard index to measure similarity between two subsets of features. They concluded that stability provides an objective criterion to choose among feature selection algorithms. Selecting the most stable algorithm gives higher confidence in the quality of selected feature subset.

Few works consider the impact of dataset similarity when performing perturbation experiments, one paper, Alelyani et al. [7], does. In this paper, the authors note that without controlling for overlap, it is difficult to tell whether two feature subsets are different due to underlying stability issues with the ranker or due to differences in the datasets they were drawn from. To evaluate this, the researchers sampled 25% of the instances into one subset, and then created nine more subsets with exactly c of their instances in common with the first. The pairwise stability of the features from these subsets were evaluated as c varied from 0 to 1. They found that some algorithms were not able to outperform the inherent stability of the underlying datasets, and so should not be considered "stable" regardless of their stability performance. Although Alelyani et al. raises an important question about the role of dataset similarity, it does not necessarily address this question to the extent it deserves. Notably, during their experiments with varying the amount of overlap between subsets, only the overlap between the first subset and the remaining nine is considered; the overlap among the nine is not, and will depend on random chance. In addition, by consistently using only 25% of the instances from their datasets (which have as few as 85 instances to start with), they discard much of their data. Finally, although their proposal to compare a ranker's stability with the minimum stability provided by the dataset is useful, it doesn't address the problem of selecting stable rankers for different subset sizes, degree of class balance, size of underlying dataset, or difficulty of learning of the underlying dataset. These questions and more remain open.

Another work, Haury et al. [4], considers the role of overlap when considering the stability of gene subsets. In addition to other analysis of their datasets, the researchers consider the fraction of instances in common when comparing feature lists generated from subsamples of the original data which either have 80% or 0% overlap. They also compare feature lists among four distinct (but related) datasets. They found that the stability measures for the 0% overlap case more closely resembled the between-datasets case than did the results from the 80% overlap case. However, unlike the 0% case, where it is noted that the original data was divided into two mutually-exclusive groups (which therefore have 0% overlap), for the 80% case the two groups were generated by adding 80% of the data from the original dataset into each group, and then splitting the remaining 20% in half and putting each half into one of the groups. Thus, the 80% refers to proportion of the original data shared by the two groups, not the overlap between the two groups. This makes it difficult to generalize the approach to create datasets with arbitrarily-chosen overlaps.

The main contribution of the present work is that we consider stability of three forms of feature selection techniques by comparing the selected features generated from two subsamples which have same number of instances and a specified level of overlap, rather than directly comparing separate subsamples of the original dataset with original datasets. In addition the Average Pairwise Tanimoto Index (APTI), which does not require feature subsets have the same size, is used to evaluate the stability of a feature selection technique.

III. Methodology

We consider fifteen different feature selection strategies: three feature rankers each coupled with three feature subset sizes, the Correlation-based Feature Selection (CFS) filter-based subset evaluator, and wrapper-based feature selection using one of five different learners inside the wrapper. The feature selection techniques are presented in Section III-A, while the Fixed-Overlap Partitions are discussed in Section III-B, and the stability measure is presented in Section III-C.

A. Feature Selection

Many techniques exist for choosing the optimal feature subset, but these can generally be placed into two categories: ranking-based methods and subset-based methods. Within the subset group, either filters or wrappers can be used to perform the actual evaluation. Filters are algorithms in which a feature subset is selected without involving any learning algorithm. Wrappers are algorithms that use feedback from a learning algorithm to determine which feature(s) to include in building a classification model. Feature rankers tend to be more efficient than subset-based methods, because a ranker need only provide a single score for each feature, and then subsets can be built based on ranked feature lists. For subset-based methods, different subsets must be considered, with the number of calculations reaching to 2^n (n is the number of features) if exhaustive search is used. Subset-based methods will take more computational resources than feature rankers.

1) Feature Ranking: For feature ranking, we choose three representative techniques: Relief (RF), Area Under the Receiver Operating Characteristic (ROC) Curve, and Signal-To-Noise (S2N). These were chosen for two reasons. First of all, they represent three major groupings of feature ranking technique: RF is a commonly-used algorithm for ranking features, while ROC is an example of threshold-based feature selection (TBFS) [8], and S2N is an example of first-order statistics-based feature selection (FOS) [9].

Relief is an instance-based feature ranking technique [10]. ReliefF is an extension of the Relief algorithm that can handle noise and multi-class datasets. When the 'weightByDistance' (weight nearest neighbors by their distance) parameter is set as default (false), the algorithm is referred to as RF.

Threshold-based Feature Selection Techniques (TBFS) were proposed and implemented by our research group [8]. In TBFS, each

attribute is evaluated against the class, independent of all other features in the dataset. After normalizing each attribute to have a range between 0 and 1, simple classifiers are built for each threshold value $\in$ [0, 1] according to two different classification rules (e.g., whether instances with values above the threshold are considered positive or negative class examples). The normalized values are treated as posterior probabilities and the performance of these probabilities is evaluated using a chosen metric, in much the same way that the posterior probabilities from a standard classifier would be evaluated. However, as the feature values are used directly, no actual classifier is built. In the present work, we used the Area Under the ROC Curve metric (ROC), which plots the True Positive Rate vs. the False Positive Rate over all possible threshold values and then uses the area under this curve as the performance of the posterior probabilities. When used as a ranker, this area is the quality of the feature.

First-order statistics-based feature selection (FOS) [9] is a family of related techniques which all center around the use of first-order statistics such as mean and standard deviation. Signal-to-noise (S2N) ratio is a technique in this family which is a measure used in electrical engineering to quantify how much a signal has been corrupted by noise. It is defined as the ratio of signal's power to the noise's power corrupting the signal. The S2N ratio can also be used as feature ranking method [11]. For a binary class problem (such as fp, nfp), the equation for signal to noise is:

$$S2N = (\mu_P - \mu_N)/(\sigma_P + \sigma_N) \tag{1}$$

where μ_P and μ_N are the mean values of that particular attribute in all of the instances which belong to a specific class, either P or N (the positive and negative classes). σ_P and σ_N are the standard deviations of that particular attribute as it relates to the two classes, respectively. If one attribute's expression in one class is quite different from its expression in the other, and there is little variation within the two classes, then the attribute is predictive. The larger the S2N ratio, the more relevant a feature is to the dataset [12].

For all three rankers, we considered three different feature subset sizes: 3, 4, and 5. These were chosen based on previous research [13] and to give a wider spectrum of the most common choices used for feature ranking on software metrics datasets.

2) Filter-Based Subset Evaluation: In this study, we evaluate one filter-based feature subset selection algorithms: Correlation-based (CFS) [14] feature subset selection. CFS employs the Pearson correlation coefficient [14], which can be calculated using the following formula:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2}$$

In this formula, M_S is the merit of the current subset of features, k is the number of features, $\overline{r_{cf}}$ is the mean of the correlations between each feature and the class, and $\overline{r_{ff}}$ is the mean of the pairwise correlations between every two features. The numerator increases when the set of features is particularly good at classifying the data, while the denominator increases when the set has a significant amount of self-correlation, which implies redundancy.

3) Wrapper-based Feature Subset Evaluation: Wrapper-based feature subset selection is building a model using a potential feature subset and using the performance of this model as a score for the merit of that subset [15]. The wrapper-based feature selection methods employ some predetermined learning algorithms (classifiers or learners) to evaluate the goodness of the subset of features being selected. The performance of this approach relies on three factors: (1) the strategy to search the feature space for possible optimal feature subsets; (2) the criterion to evaluate the classification model built with the selected subset of features; (3) and the learner.

Suppose a large set of n features is given, we need to find a small subset of features for future model building. Inspecting all candidate subsets (2^n) is impractical. There are some strategies that can solve the problem. One way is to use a search algorithm to generate the possible feature subsets. Based on preliminary experimentation, we chose the Greedy Stepwise approach, which uses forward selection to build the full feature subset starting from the empty set. At each point in the process, the algorithm creates a new family of potential feature subsets by adding every feature (one at a time) to the current best-known set. The merit of all these sets are evaluated, and whichever performs best is the new known-best set. The wrapper and CFS procedures terminate when none of the new sets outperform the previous known-best set.

During the search process, classification models are built using a potential feature subset and using the performance of this model as a score for the merit of that subset [15]. For our experiments the wrapper process uses five-fold cross-validation: the training set is divided into five equal folds (partitions), a classifier is trained on four folds, then tested on the last (fifth) fold. This process is repeated five times, and the results are averaged to give the merit of the potential feature subset. In this study, the classification models are evaluated using the Area Under ROC (Receiver Operating Characteristic) Curve (AUC) performance metric.

In this work, five diverse learners are used within the wrapper-based feature subset selector, consisting of naïve Bayes, multilayer perceptron, k-nearest neighbors, support vector machine, and logistic regression. The five learners were selected because of their common use in the software engineering and other application domains, and also because they do not have a built-in feature selection capability. Unless stated otherwise, we use default parameter settings for the different learners as specified in WEKA [16]. Parameter settings are changed only when a significant improvement in performance is obtained.

1) Naïve Bayes (NB) utilizes Bayes's rule of conditional probability and is termed 'naive' because it assumes conditional independence of the features.
2) Multilayer Perceptron (MLP) is a neural network of simple neurons called perceptrons. Some related parameters of MLP were set as follows: the 'hiddenLayers' parameter was set to 3 to define a network with one hidden layer containing three nodes and the 'validationSetSize' was set to 10 (with 10% of the data being held aside for validating when to stop the backpropagation procedure).
3) K-Nearest Neighbors (KNN) [17], also called instance-based learning, uses distance-based comparisons. KNN was built with changes to two parameters. The 'distanceWeighting' parameter was set to 'Weight by 1/distance' and the 'kNN' parameter was set to 5.
4) Support Vector Machine (SVM), also called SMO in WEKA [16], had two changes to the default parameters: the 'complexity constant c' was set to 5.0 and 'build Logistic Models' was set to true. By default, a linear kernel was used.
5) Logistic Regression (LR) [18] is a statistical regression model for categorical prediction by fitting data to a logistic curve.

B. Fixed-overlap Partitions

Many approaches have been used to test the stability of feature selection techniques. Some take random subsamples from the original dataset and compare the features chosen on these subsamples with

each other; others compare the features chosen on the subsamples with those chosen from the original dataset. The first of these approaches has a known flaw: it does not control for the degree of overlap between the subsamples being compared (instead leaving this to random chance). This makes it difficult to determine whether the stability between feature subsets is due to similarity of the underlying datasets or is a property of the feature selection technique used. The second approach is somewhat limited in scope: although it is useful for observing stability in the case of adding or removing instances from a dataset, its use of two datasets of different sizes can impact how well the results generalize to other perturbation scenarios. Neither is able to evaluate how similar the feature subsets will be for two datasets which are equal in size and have a known degree of overlap. To address this, our research group proposed the Fixed-Overlap Partitions Algorithm [19] (Algorithm 1), which will create two new subsets that have the desired properties while also being as large as possible for the given degree of overlap. Note in this algorithm that c, the desired degree of overlap, can vary from 0 to 1, including the endpoints. A choice of $c = 0$ will find two entirely disjoint subsets, which will each contain half of the instances from the original dataset. On the other hand, $c = 1$ will create two copies of the original dataset which share all instances. This is generally not an interesting case to study, but is permitted by the algorithm.

Algorithm 1: Fixed-Overlap Partitions

input : Original dataset S with N instances
: c, the fraction of instances the two subsampled datasets should have in common ($0 \leq c \leq 1$)

output: Datasets S_1 and S_2 which have c of their instances in common while being identical in size and as large as possible for the given c

Let $d = 1/(2 - c)$ (e.g., $c = (2d - 1)/d$)
S_1 and S_2 start out empty
Randomly select dN instances from S and add them to S_1
Randomly select cdN instances from S_1 and add them to S_2
Take all instances in S which are not in S_1 and add them to S_2

There are three properties which must be guaranteed when selecting these subsets: 1) that they contain the same number of instances, 2) that they have the specified degree of overlap, and 3) that they are as large as possible while the first two properties hold true (since there is no reason to discard instances if they could be used to improve feature selection or classification). Based on Algorithm 1, we can see that S_1 contains dN instances. To find the number of instances in S_2, we note that two steps add instances to that dataset: one adds cdN instances and the other adds the instances not included in S_1 (e.g., $(1 - d)N$ instances). Working from here and using the definition of d in the algorithm, we have:

$$\begin{aligned} |S_2| &= cdN + (1-d)N \\ &= \left(\frac{2d-1}{d}\right) dN + (1-d)N \\ &= (2d-1)N + (1-d)N \\ &= 2dN - N + N - dN \\ &= dN \end{aligned}$$

Thus, we have $|S_1| = |S_2| = dN$, satisfying the first property. As for the second property, recall that S_1 and S_2 share precisely cdN instances; thus, they have $cdN/dN = c$ of their instances in common, as desired. For the third property, observe that adding any instances to either S_1 or S_2 would necessarily increase the fraction of overlap (since these would have to be instances already found in the other subsampled dataset). Thus, S_1 and S_2 are the largest datasets which are identical in size and have an overlap of precisely c. In this study, the degree of overlap is chosen from the set $\{0.25, 0.5, 0.7, 0.85\}$. A choice of $c = 0.85$ will generate two subsets with 0.87 ($d = 1/(2 - c)$)×N instances.

C. Stability Measurement

In order to measure stability, first we have to decide the measurement metric. In this study we choose the Average Pairwise Tanimoto Index (APTI), derived from work originating in Kalousis et al. [6], since it does not require feature subsets have the same size. Let S_i and S_j be two different subsets of features. The original Tanimoto Index defines the stability between the two feature subsets as follows:

$$T(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} = 1 - \frac{|S_i| + |S_j| - 2|S_i \cap S_j|}{|S_i| + |S_j| - |S_i \cap S_j|} \quad (3)$$

where $\frac{|S_i|+|S_j|-2|S_i \cap S_j|}{|S_i|+|S_j|-|S_i \cap S_j|}$ is the Tanimoto distance between the two feature subsets. In this work, we propose Average Pairwise Tanimoto Index (APTI) that can be used to determine the stability of a set of feature subset pairs which are generated by same feature selection method on a pair of subsamples:

$$APTI(S^1, S^2) = \frac{1}{W} \sum_{i=1}^{W} T(S_{i1}^1, S_{i2}^2) \quad (4)$$

Here, we assume that S^1 and S^2 are paired feature subsets generated by same feature selection method on a pair of subsamples. For each pair of subsamples S_{i1} and S_{i2}, there are two corresponding feature subsets, S_{i1}^1 and S_{i2}^2. The stability index ($APTI$) defined in Equation 4 varies in the interval of [0,1]. As APTI is an average of Tanimoto Index values, its maximum of 1 represents the case where all pairwise comparisons are identical subsets, and the minimum of 0 means that no pairs ever have any features in commons. In our experiments, W is set to 30. That means for each dataset, 30 pairs subsamples are generated with certain level of overlap.

IV. Experiments

A. Experimental Datasets

Experiments conducted in this study used software metrics and defect data collected from a real-world software project, and included data from four consecutive releases of a very large telecommunications software system (denoted as LLTS). The LLTS software system was comprised of several million lines of code. The data collection effort used the Enhanced Measurement for Early Risk Assessment of Latent Defect (EMERALD) system [20]. The software measurement dataset of LLTS contains data from four consecutive releases, which are labeled as SP1, SP2, SP3, and SP4. Each dataset includes 42 software metrics, including 24 product metrics, 14 process metrics, and four execution metrics. The dependent variable is the class of the program module: fault-prone (*fp*) or not fault-prone (*nfp*). A program module with one or more faults is considered *fp*, and *nfp* otherwise. Table I summarizes the numbers of the *fp* and *nfp* modules and their percentages in each dataset. A unique characteristic of these datasets is that they all are highly imbalanced datasets, where the proportion of *fp* modules is much lower than the *nfp* modules.

TABLE I
SOFTWARE DATASETS CHARACTERISTICS

	Data	#Metrics	#Modules	*%fp*	*%nfp*
LLTS	SP1	42	3649	6.28%	93.72%
	SP2	42	3981	4.75%	95.25%
	SP3	42	3541	1.33%	98.67%
	SP4	42	3978	2.31%	97.69%

TABLE II
STABILITY OF FEATURE SELECTION FOR SP1

Feature Selection	Overlap			
	0.25	0.5	0.7	0.85
RF, 3	**0.8333**	**0.9000**	0.8833	**0.9333**
RF, 4	0.8267	0.8400	0.8533	0.9067
RF, 5	0.6667	0.7111	0.7222	0.8111
ROC, 3	0.6400	0.7900	**0.9167**	0.9167
ROC, 4	0.5200	0.6000	0.6622	0.7778
ROC, 5	0.4266	0.4901	0.5909	0.7000
S2N, 3	0.4967	0.6000	0.5667	0.6500
S2N, 4	0.4159	0.5111	0.5644	0.6400
S2N, 5	0.4762	0.5540	0.6937	0.7667
CFS	0.4216	0.5223	0.5633	0.5972
Wrapper-NB	0.3973	0.5365	0.5696	0.6761
Wrapper-MLP	0.2645	0.3064	0.3419	0.3297
Wrapper-5NN	0.1679	0.1745	0.1988	0.2304
Wrapper-SVM	*0.1000*	*0.0931*	*0.1056*	*0.1413*
Wrapper-LR	0.3329	0.3504	0.4049	0.4707

B. Experimental Design

Experiments are conducted with fifteen different feature selection strategies on four software engineering metric datasets from a real-world software project. These feature selection strategies include three feature rankers each coupled with three feature subset sizes, the Correlation-based Feature Selection (CFS) filter-based subset evaluator, and wrapper-based feature subset selection using one of five different learners inside the wrapper. The goal of the experiments is to study how these feature selection methods can affect the stability of feature selection process. Thirty pairs of subsamples were generated from each original dataset with four different levels of overlap, and each feature selection method was applied to each pair of subsample. Once these feature subsets were created, the stability of the pairs of feature subset generated by same feature selection method were compared using our newly proposed Average Pairwise Tanimoto Index (APTI) described in Section III-C. In total, we calculate 240 APTI values (4 datasets $\times$ 15 feature selection $\times$ 4 overlap levels).

TABLE III
STABILITY OF FEATURE SELECTION FOR SP2

Feature Selection	Overlap			
	0.25	0.5	0.7	0.85
RF, 3	0.6167	0.6167	0.7000	0.6500
RF, 4	0.6756	0.7156	0.6978	0.6978
RF, 5	**0.7571**	**0.8254**	**0.9254**	0.9778
ROC, 3	0.6800	0.8233	0.9167	**0.9833**
ROC, 4	0.6400	0.6933	0.7867	0.8800
ROC, 5	0.6143	0.6476	0.7206	0.7698
S2N, 3	0.4067	0.4233	0.5867	0.7067
S2N, 4	0.4438	0.5117	0.6889	0.7333
S2N, 5	0.4898	0.6333	0.7444	0.8175
CFS	0.45	0.49	0.54	0.64
Wrapper-NB	0.4259	0.5069	0.5417	0.6462
Wrapper-MLP	0.2015	0.2120	0.2783	0.2246
Wrapper-5NN	0.1461	0.1949	0.2898	0.2900
Wrapper-SVM	*0.0728*	*0.0787*	*0.0717*	*0.0739*
Wrapper-LR	0.2871	0.2780	0.4063	0.4401

TABLE IV
STABILITY OF FEATURE SELECTION FOR SP3

Feature Selection	Overlap			
	0.25	0.5	0.7	0.85
RF, 3	**0.5667**	0.6000	0.7000	0.7833
RF, 4	0.4889	**0.6044**	**0.7511**	0.7867
RF, 5	0.4675	0.5619	0.7381	**0.8778**
ROC, 3	0.2633	0.3633	0.4167	0.4000
ROC, 4	0.2590	0.3263	0.4006	0.4356
ROC, 5	0.2381	0.3636	0.4706	0.4964
S2N, 3	0.4833	0.6000	0.7800	0.8567
S2N, 4	0.4654	0.5473	0.7244	0.8133
S2N, 5	0.4516	0.5340	0.6353	0.6635
CFS	0.1685	0.2352	0.3022	0.3890
Wrapper-NB	0.2094	0.3133	0.3417	0.4135
Wrapper-MLP	0.1412	0.2274	0.2511	0.2667
Wrapper-5NN	0.1305	*0.1368*	0.2289	0.2706
Wrapper-SVM	*0.0862*	0.1589	*0.1223*	*0.1262*
Wrapper-LR	0.1767	0.2126	0.2759	0.2663

TABLE V
STABILITY OF FEATURE SELECTION FOR SP4

Feature Selection	Overlap			
	0.25	0.5	0.7	0.85
RF, 3	0.7167	0.7667	0.8000	0.8000
RF, 4	**0.7911**	**0.8933**	**0.9333**	**1.0000**
RF, 5	0.7349	0.8667	0.8556	0.8667
ROC, 3	0.0767	0.1433	0.2533	0.3800
ROC, 4	0.1254	0.2108	0.3321	0.4470
ROC, 5	0.1696	0.2696	0.4106	0.5362
S2N, 3	0.5600	0.7033	0.8067	0.8833
S2N, 4	0.5689	0.6711	0.7422	0.7378
S2N, 5	0.5639	0.6492	0.7429	0.8063
CFS	0.2634	0.3485	0.4231	0.5061
Wrapper-NB	0.2839	0.3426	0.5246	0.4964
Wrapper-MLP	0.1323	0.1673	0.1879	0.3063
Wrapper-5NN	0.1596	0.1510	0.2082	0.1978
Wrapper-SVM	*0.0379*	*0.0636*	*0.0784*	*0.0317*
Wrapper-LR	0.2004	0.2539	0.3282	0.3628

C. Results and Analysis

Table II through Table V list the stability results for each dataset. These tables show the stability of subsets generated by each feature selection method (row) on subsamples with different level of overlap (column). For example, the first value in Table II, 0.8333, represents the stability of two feature subsets selected by Relief (RF) with feature subset size three and the overlap level of the pair of subsamples is 0.25. For each overlap level, the most and least value (stability) are printed in **bold** and *italics*, respectively. Figure 1 shows stability on average across all four datasets. From these tables and figure, we can observe the following facts:

- Overall, we can order the three classes of feature selection strategies from the most stability to least stability, ranker, filter-based subset evaluators, and wrapper-based subset evaluators. In terms of ranker, RF shows extremely high stability. The highest stability is found for RF with subset size four and overlap level 0.85 on dataset SP4. Followed by RF, ROC shows more stability than S2N for SP1 and SP2 datasets, while S2N shows more stability than ROC for SP3 and SP4 datasets. There are no patterns to show the relationship between feature subset size and stability of selected feature subsets.
- Comparing to other classes of feature strategies, the similarities of wrappers are low. Among the five wrappers, NB wrapper

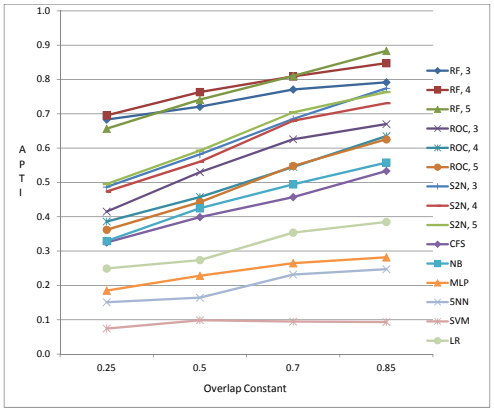

Fig. 1. Similarities of Feature Selection Methods

shows the most stability regardless of overlap level. The next similar wrapper is LR wrapper. SVM wrapper shows least stability, no feature subset pairs generated by SVM wrapper have a stability greater than 0.1 for SP2 and SP4 datasets and the stability value does not exceed 0.16 for SP1 and SP3 datasets. It is clear from these results that the choice of learner will have a very important effect on the chosen features.

- While intuitive, the results show that as the overlap of partitions increased, the stability of the feature selection strategies increased. This indicates that with enough change any selected subset become unstable.

V. Conclusion

Software metrics collected during project development play a critical role in software quality assurance. A typical project often collects large number of metrics. Metric (feature) selection plays an important role in data preprocessing step. By removing irrelevant and redundant features from a training dataset, software quality estimation based on some classification models may improve. One consequence of removing redundancy can be reducing stability: that is, the subset of chosen features may change significantly in the face of relatively small changes to the input dataset. In this paper, we propose a new metric for measure the stability on subset selected by feature selection techniques.

In this study, we present a stability analysis of of 15 feature selection methods (three feature ranking with three different subset sizes, one filter-based subset evaluator, and five wrappers) on a real-world software project. A newly-proposed variation of the Tanimoto Index (the Average Pairwise Tanimoto Index (APTI)) was used to evaluate the stability between subsets selected by feature selection methods. Experimental results demonstrate that the choice of feature selection methods has a major effect on the feature subsets. We find that there is the most stability (though not congruence) between the subsets chosen using rankers especially the RF ranker. The subsets selected by wrappers are even more dissimilar from one another. In addition, as the overlap of partitions increased, the stability of the feature selection strategies increased.

Future work may compare stability of a wide range of feature ranking techniques with more feature subset sizes, filter-based subset evaluators, and wrappers with different choices of learners and performance metrics. Experiments may be conducted on additional software metrics datasets from the software engineering domain.

References

[1] T. M. Khoshgoftaar, K. Gao, A. Napolitano, and R. Wald, "A comparative study of iterative and non-iterative feature selection techniques for software defect prediction," *Information Systems Frontiers*, vol. 16, no. 5, pp. 801–822, 2014.

[2] P. Somol and J. Novovičová, "Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 1921–1939, 2010.

[3] J. L. Lustgarten, V. Gopalakrishnan, and S. Visweswaran, "Measuring stability of feature selection in biomedical datasets," in *AMIA 2009 Annual Symposium Proceedings*, 2009, pp. 406–410.

[4] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," *PLoS ONE*, vol. 6, no. 12, p. e28210, 12 2011. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0028210

[5] K. Dunne, P. Cunningham, and F. Azuaje, "Solutions to Instability Problems with Sequential Wrapper-Based Approaches To Feature Selection," *Machine Learning*, no. TCD-CD-2002-28, pp. 1–22, 2002.

[6] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and Information Systems*, vol. 12, no. 1, pp. 95–116, May 2007.

[7] S. Alelyani, Z. Zhao, and H. Liu, "A dilemma in assessing stability of feature selection algorithms," in *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, Sept. 2011, pp. 701 –707.

[8] H. Wang, T. M. Khoshgoftaar, and J. Van Hulse, "A comparative study of threshold-based feature selection techniques," in *2010 IEEE International Conference on Granular Computing, GrC 2010, San Jose, California, USA, 14-16 August 2010*, 2010, pp. 499–504.

[9] H. Wang, T. M. Khoshgoftaar, R. Wald, and A. Napolitano, "A study on first order statistics-based feature selection techniques on software metric data," in *The 25th International Conference on Software Engineering and Knowledge Engineering, Boston, MA, USA, June 27-29, 2013.*, 2013, pp. 467–472.

[10] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of 9th International Workshop on Machine Learning*, 1992, pp. 249–256.

[11] C.-H. Yang, C.-C. Huang, K.-C. Wu, and H.-Y. Chang, "A novel gataguchi-based feature selection method," in *IDEAL '08: Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning*, Berlin, Heidelberg, 2008, pp. 112–119.

[12] M. Wasikowski and X. wen Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1388–1400, 2010.

[13] H. Wang, T. M. Khoshgoftaar, and N. Seliya, "How many software metrics should be selected for defect prediction?" in *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference, May 18-20, 2011, Palm Beach, Florida, USA*, 2011.

[14] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, April 1997.

[15] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, Dec. 1997. [Online]. Available: http://dx.doi.org/10.1016/S0004-3702(97)00043-X

[16] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann, 2005.

[17] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 1573–0565, January 1991.

[18] S. Le Cessie and J. C. Van Houwelingen, "Ridge estimators in logistic regression," *Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992.

[19] 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 *IEEE 13th International Conference on Information Reuse & Integration, IRI 2012, Las Vegas, NV, USA, August 8-10, 2012*, 2012, pp. 1–8.

[20] J. P. Hudepohl, S. J. Aud, T. M. Khoshgoftaar, E. B. Allen, and J. Mayrand, "EMERALD: Software metrics and models on the desktop," *IEEE Software*, vol. 13, no. 5, pp. 56–60, September 1996.