A Comparative Study of Threshold-based Feature Selection Techniques

Huanjing Wang  
*Western Kentucky University*, huanjing.wang@wku.edu

Taghi M. Khoshgoftaar  
*Florida Atlantic University*, taghi@cse.fau.edu

Jason Van Hulse  
*Florida Atlantic University*, jvanhulse@gmail.com

Follow this and additional works at: [http://digitalcommons.wku.edu/comp_sci](http://digitalcommons.wku.edu/comp_sci)

Part of the Artificial Intelligence and Robotics Commons, Databases and Information Systems Commons, and the Other Computer Sciences Commons

**Recommended Repository Citation**

Available at: [http://digitalcommons.wku.edu/comp_sci/6](http://digitalcommons.wku.edu/comp_sci/6)

*This Article is brought to you for free and open access by TopSCHOLAR®. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of TopSCHOLAR®. For more information, please contact topscholar@wku.edu.*
A Comparative Study of Threshold-based Feature Selection Techniques

Huanjing Wang  
Western Kentucky University  
huanjing.wang@wku.edu

Taghi M. Khoshgoftaar  
Florida Atlantic University  
taghi@cse.fau.edu

Jason Van Hulse  
Florida Atlantic University  
jvanhulse@gmail.com

Abstract

Given high-dimensional software measurement data, researchers and practitioners often use feature (metric) selection techniques to improve the performance of software quality classification models. This paper presents our newly proposed threshold-based feature selection techniques, comparing the performance of these techniques by building classification models using five commonly used classifiers. In order to evaluate the effectiveness of different feature selection techniques, the models are evaluated using eight different performance metrics separately since a given performance metric usually captures only one aspect of the classification performance. All experiments are conducted on three Eclipse data sets with different levels of class imbalance. The experiments demonstrate that the choice of a performance metric may significantly influence the results. In this study, we have found four distinct patterns when utilizing eight performance metrics to order 11 threshold-based feature selection techniques. Moreover, performances of the software quality models either improve or remain unchanged despite the removal of over 96% of the software metrics (attributes).

Keywords: performance metrics, threshold-based feature selection technique, software metrics, classification.

1 Introduction

Given a set of software metrics (independent features or attributes) the objective of feature selection is to remove irrelevant or redundant features, which can then be discarded from the analysis. Reducing the number of features in a data set can lead to faster model training and improved classifier performance. Feature selection has been widely used and thoroughly researched [6, 7, 10]. The two general categories for feature selection are filters and wrappers. Filters are algorithms in which a feature subset is selected without involving any learner. Wrappers are algorithms that use feedback from a learning algorithm to determine which feature(s) to include in building a classification model. Another categorization for feature selection techniques is feature ranking and feature subset selection techniques. Feature ranking ranks the attributes according to their individual predictive power, while feature subset selection selects subsets of attributes that collectively have good predictive power. We consider filter-based feature ranking techniques in this study.

In this paper, we present our newly proposed threshold-based feature selection techniques (TBFS) which represent a substantial extension of the FAST algorithm proposed by Chen and Wasikowsi [2]. Our technique is much more general than that of Chen and Wasikowski. Their procedure calculates a ROC curve by discretizing the distribution, while ours does not require discretization, making it more precise. Furthermore, there are 11 different versions of TBFS which are based on 11 different classifier performance metrics. TBFS can also be extended to incorporate additional metrics.

The 11 threshold-based feature selection techniques are evaluated using software measurement data in our case study, including three data sets of release 3.0 of a real-world software project, Eclipse [13]. In order to evaluate the classification performance of the TBFS techniques on the smaller subsets of attributes, several classification models are built using five commonly used classifiers. Since related literature lacks general agreement on which performance metrics should be used for evaluating classification performance [9, 8], learners are evaluated using eight performance metrics. The experiments demonstrate that the choice of a performance metric can significantly influence the conclusions. In addition, we have found four distinct patterns when we use eight performance metrics to order 11 threshold-based feature selection techniques.

The main contribution of this work is the presentation of a set of novel threshold-based feature selection techniques and an assessment and comparison of these feature selection methods using eight performance metrics and five commonly used classifiers.

The remainder of the paper is organized as follows. Section 2 explains our threshold-based feature selection methodology. Section 3 describes the learners, performance
Algorithm 1: Threshold-based Feature Selection Algorithm

input:
1. Data set D with features \( F_j, j = 1, \ldots, m \);
2. Each instance \( x \in D \) is assigned to one of two classes \( c(x) \in \{ f_p, n_f \} \);
3. The value of attribute \( F_j \) for instance \( x \) is denoted \( F_j(x) \);
4. Threshold-based feature ranking technique \( \omega \in \{ \text{BFM, OR, PO, PR, GI, MI, KS, DV, BGM, AUC, PRC} \} \);
5. A predefined threshold: number (or percentage) of the features to be selected.

output:
Selected feature subsets.

for \( F_j, j = 1, \ldots, m \) do
- Normalize \( F_j \) as \( \hat{F}_j = \frac{F_j - \min(F_j)}{\max(F_j) - \min(F_j)} \);
- Calculate metric \( \omega \) using attribute \( \hat{F}_j \).
- Create feature ranking \( \mathbb{R} \) using \( \omega_j(\hat{F}_j) \) \( \forall j \).
- Select features according to feature ranking \( \mathbb{R} \) and a predefined threshold.

metrics and case study data sets used in this work, and presents the experimental results. Finally, we conclude the paper in Section 4 and provide suggestions for future work.

2 Threshold-based Feature Selection Techniques

Filter-based feature ranking techniques (filters) rank features independently without involving any learning algorithm. Eleven threshold-based feature selection techniques (TBFS) were developed and implemented by our research group within Weka [12]. The procedure is shown in Algorithm 1. First each attribute’s values are normalized between 0 and 1 by mapping \( F_j \) to \( \hat{F}_j \). The normalized values are treated as posterior probabilities. Each independent attribute is then paired individually with the class attribute and the reduced two attribute data set is evaluated using 11 different performance metrics based on a set of posterior probabilities. In standard binary classification, the predicted class is assigned using the default decision threshold of 0.5. The default decision threshold is often not optimal, especially when the class is imbalanced. Therefore, we propose the use of performance metrics which allow for finding the optimal threshold.

The true positive (\( TPR \)), true negative (\( TNR \)), false positive (\( FPR \)), false negative (\( FNR \)), precision (\( PRE \)), negative predicted value (\( NPV \)) [11] rates can be calculated at each threshold \( t \in [0, 1] \) relative to the normalized attribute \( \hat{F}_j \). The threshold-based attribute ranking techniques we propose utilize these rates as described below.

- **Best F-measure (BFM):** is a single value metric derived from the F-measure that originated from the field of information retrieval.

\[
\text{BFM} = \max_{t \in [0,1]} \frac{(1 + \beta^2) \times TPR(t) \times \text{PRE}(t)}{\beta^2 \times TPR(t) + \text{PRE}(t)}
\]

\( \beta \) is set to 1 in this study. The maximum F-measure (BFM) is obtained when varying the decision threshold value between 0 and 1.

- **Odds Ratio (OR):** is the maximum value of the ratio of the product of correct (true positive rate times true negative rate) to incorrect (false positive rate times false negative rate) predictions. The odds ratio is defined as:

\[
\text{OR} = \max_{t \in [0,1]} \frac{TPR(t)}{FPR(t)} \cdot \frac{TNR(t)}{FNR(t)}
\]

- **Power (PO):** is a measure that avoids common false positive cases while giving stronger preference for positive cases [5]. Power is defined as:

\[
\text{PO} = \max_{t \in [0,1]} \left( (TNR(t))^k - (FNR(t))^k \right)
\]

where \( k = 5 \).

- **Probability Ratio (PR):** is the sample estimate probability of the feature given the positive class divided by the sample estimate probability of the feature given the negative class [5]. The probability ratio is defined as:

\[
\text{PR} = \max_{t \in [0,1]} \frac{TPR(t)}{FPR(t)}
\]

- **Gini Index (GI):** measures the impurity of a data set. GI for the attribute is then the minimum Gini index at all decision thresholds \( t \in [0, 1] \).

\[
\text{GI} = \min_{t \in [0,1]} [2\text{PRE}(t)(1 - \text{PRE}(t)) + 2\text{NPV}(t)(1 - \text{NPV}(t))]
\]

- **Mutual Information (MI):** measures the mutual dependence of the two random variables. High mutual information indicates a large reduction in uncertainty, and zero mutual information between two random variables means the variables are independent.

- **Kolmogorov-Smirnov (KS):** utilizes the Kolmogorov-Smirnov statistic to measure the maximum difference between the empirical distribution function of the attribute values of instances in each class. The larger the distance between the distribution functions, the better the attribute is able to distinguish between the two classes. It is effectively the maximum difference between the curves generated by the true positive and false positive rates as the decision threshold changes from 0 and 1.
was changed from 1.0 to 5.0 and hidden.

TNR as the decision threshold is varied be-

• Deviance (DV): is the minimum residual sum of
   squares based on a threshold t. That is, it measures the
   sum of the squared errors from the mean class given a
   partitioning of the space based on the threshold t.

• Best Geometric Mean (BGM): is a single-value perfor-
   mance measure that ranges from 0 to 1, which is cal-
   culated by finding the maximum geometric mean of
   TPR and TNR as the decision threshold is varied be-
   tween 0 and 1:

\[
\text{BGM} = \max_{t \in [0,1]} \sqrt{TPR(t) \times TNR(t)}
\]  

• Area Under ROC (Receiver Operating Characteristic)
   Curve (AUC): has been widely used to measure classi-
   fication model performance [4]. AUC is a single-value
   measure that ranges from 0 to 1. The ROC curve is
   used to characterize the trade-off between true posi-
   tive rate and false positive rate. In this study, ROC
curves are generated by varying the decision threshold
t used to transform the normalized attribute values into
a predicted class.

• Area Under the Precision-Recall Curve (PRC): is a
   single-value measure that originated from the area of
   information retrieval. The area under the PRC ranges
   from 0 to 1. The PRC diagram depicts the trade off
   between recall and precision.

In this study, AUC, PRC, BFM, and BGM serve as both
aids to the feature ranking process and the final inducive
algorithm evaluation process (see Section 3.2).

### 3 Experiments

#### 3.1 Classifiers

Classifiers are built with five well-known classification
algorithms [12] including naive Bayes (NB), multilayer per-
ceptron (MLP), k-nearest neighbors (KNN), support vec-
tor machine (SVM) [3] and logistic regression (LR). These
were selected because of their common use in data mining
applications. Unless stated otherwise, the default parameter
settings are used for the learners as specified in Weka [12].
Parameter settings were changed only when a significant
improvement in performance based on preliminary experi-
mentation was obtained. For the KNN classifier, 5 neigh-
bors were used and the distanceWeighting parameter was
set to “Weight by 1/Distance”. For the MLP learner, hidden-
Layers was changed to 3 to define a network with one
hidden layer containing three nodes, and validationSetSize
was changed to 10 to cause the classifier to leave 10% of
the training data aside to be used as a validation set to
determine when to stop the iterative training process. For
SVM, the complexity constant c was changed from 1.0 to 5.0
and buildLogisticModels was enabled.

#### 3.2 Classifier Performance Metrics

Eight performance metrics are used in the study includ-
ing AUC, PRC, DFM (Default F-measure corresponds to
the complexity constant \(c\) ∈ [0,1]), TPR and TNR, Deviance
\(D\), Best F-measure \(BFM\), and Best Geometric Mean
\(BGM\) which is the largest value of F-measure when varying
the decision threshold value between 0 and 1), DGM (Default
Geometric Mean), BGM (Best Geometric Mean), DAM

\[\text{TNR} (t) = \frac{TN}{TN + FP}\]

\[\text{TPR} (t) = \frac{TP}{TP + FN}\]

\[
DV = \log_{e} \left( \frac{TNR(t)}{1 - TPR(t)} \right)
\]

\[
\text{PRC} \left( \frac{BP(t)}{TPR(t)} \right)
\]

\[
\text{AUC} = \int_0^1 \text{TPR}(t) \, dt
\]

\[
\text{BFM} = \max_{t \in [0,1]} \left( \frac{TPR(t) + TNR(t)}{2} \right)
\]

\[
\text{BGM} = \max_{t \in [0,1]} \sqrt{TPR(t) \times TNR(t)}
\]

\[
\text{PR} = \frac{TP}{TP + FP}
\]

\[
\text{PO} = \frac{TN}{TN + FN}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]

\[
\text{KS} = \max \left( \frac{TPR(t)}{TNR(t)} \right)
\]

\[
\text{PR} = \frac{TP}{TP + FN}
\]

\[
\text{PO} = \frac{TN}{TN + FP}
\]
Table 3. Performance Metrics using KNN

<table>
<thead>
<tr>
<th>Data Set</th>
<th>AUC</th>
<th>PRC</th>
<th>DFM</th>
<th>BFM</th>
<th>DGM</th>
<th>BGM</th>
<th>DAM</th>
<th>BAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse 3.0-5</td>
<td>0.9406</td>
<td>0.7911</td>
<td>0.6773</td>
<td>0.7314</td>
<td>0.7118</td>
<td>0.8479</td>
<td>0.7461</td>
<td>0.8480</td>
</tr>
<tr>
<td>Eclipse 3.0-10</td>
<td>0.9372</td>
<td>0.7942</td>
<td>0.6705</td>
<td>0.7281</td>
<td>0.7251</td>
<td>0.8479</td>
<td>0.7461</td>
<td>0.8480</td>
</tr>
</tbody>
</table>

(3.4) Experimental Results and Analysis

In the experiments, ten runs of five-fold cross-validation are performed. For each of the five folds, one fold is used as test data while the other four are used as the training data. First, we rank the attributes using the 11 threshold-based feature ranking techniques separately. Once the attributes are ranked, the top [\log_2 208] = 8 (there are 208 independent features) attributes are selected (as well as the class attribute) to yield final training data. Selecting eight attributes was deemed reasonable for these experiments, and space considerations keep us from presenting results with other parameter choices. This final training data is then used to build the classification model, the resulting model is applied to the test-fold, and the eight performance metrics are calculated.

Table 4. Performance Metrics using LR

<table>
<thead>
<tr>
<th>Data Set</th>
<th>AUC</th>
<th>PRC</th>
<th>DFM</th>
<th>BFM</th>
<th>DGM</th>
<th>BGM</th>
<th>DAM</th>
<th>BAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse 3.0-5</td>
<td>0.9406</td>
<td>0.7911</td>
<td>0.6773</td>
<td>0.7314</td>
<td>0.7118</td>
<td>0.8479</td>
<td>0.7461</td>
<td>0.8480</td>
</tr>
<tr>
<td>Eclipse 3.0-10</td>
<td>0.9372</td>
<td>0.7942</td>
<td>0.6705</td>
<td>0.7281</td>
<td>0.7251</td>
<td>0.8479</td>
<td>0.7461</td>
<td>0.8480</td>
</tr>
</tbody>
</table>

3.4 Experimental Results and Analysis

In the experiments, ten runs of five-fold cross-validation are performed. For each of the five folds, one fold is used as test data while the other four are used as the training data. First, we rank the attributes using the 11 threshold-based feature ranking techniques separately. Once the attributes are ranked, the top [\log_2 208] = 8 (there are 208 independent features) attributes are selected (as well as the class attribute) to yield final training data. Selecting eight attributes was deemed reasonable for these experiments, and space considerations keep us from presenting results with other parameter choices. This final training data is then used to build the classification model, the resulting model is applied to the test-fold, and the eight performance metrics are calculated.
Table 5. Summary of Optimal TBFS Filter

<table>
<thead>
<tr>
<th>Filters</th>
<th>AUC</th>
<th>PRC</th>
<th>BFM</th>
<th>DFM</th>
<th>DAM</th>
<th>BAM</th>
<th>Total % of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFM</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>OR</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6.7%</td>
</tr>
<tr>
<td>PO</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4.2%</td>
</tr>
<tr>
<td>PR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>GI</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>6.3%</td>
</tr>
<tr>
<td>MI</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.2%</td>
</tr>
<tr>
<td>KS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2.1%</td>
</tr>
<tr>
<td>DV</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>14.1%</td>
</tr>
<tr>
<td>BGM</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>AUC</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>22</td>
</tr>
</tbody>
</table>

Focusing on Factor A, the test results indicate that for each performance metric, there was a significant difference between the average values of the 11 TBFS methods. For all eight performance metrics, the $p$-value is less than 5%, although the significance varies substantially among metrics (e.g., PRC in Table 6(b) compared to AUC in Table 6(a)). Multiple comparison tests were conducted on Factor A, since this study mainly focuses on the attribute selection techniques and their classifier performance evaluation. Both ANOVA and multiple comparison tests are implemented in MATLAB. An exhaustive discussion of Factor B is avoided due to space consideration. 

The performance of the threshold-based filters was ranked from best to worst for each performance metric as shown in Table 7. Each filter is labeled with a superscript. The filters labeled with the same superscripts implies that they were from same performance group, in which no statistically significant difference was found between filters. Some findings can be summarized from Table 7: (1) Four distinct groups of results were found when we order 11 filters based on eight performance metrics (over all the classifiers built): (a) AUC, BAM, and BGM; (b) PRC; (c) BFM; and (d) DFM, DAM, and DGM. (2) Among the 11 threshold-based feature selection techniques, the performance of AUC-based filter performed best overall. PR- and GI-based filters performed worst, followed by OR regardless of performance metrics. While PO was most often the optimal technique (Table 5), this is somewhat offset by relatively worse performance in other situations.

Table 8 presents performances of the defect classification models built with the complete set of features. Comparing these results to Tables 1 through 4, classification models
Table 8. Performance of Full Data Sets

<table>
<thead>
<tr>
<th>Data</th>
<th>Learner</th>
<th>AUC</th>
<th>MI</th>
<th>GI</th>
<th>PO</th>
<th>KS</th>
<th>BGM</th>
<th>PR</th>
<th>OR</th>
<th>BFM</th>
<th>DFM</th>
<th>BAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse 3.0-10</td>
<td>MLP</td>
<td>0.7697</td>
<td>0.3838</td>
<td>0.4000</td>
<td>0.4539</td>
<td>0.5722</td>
<td>0.7435</td>
<td>0.6598</td>
<td>0.7526</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.7325</td>
<td>0.3566</td>
<td>0.3999</td>
<td>0.4205</td>
<td>0.4034</td>
<td>0.7322</td>
<td>0.5566</td>
<td>0.7588</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.8091</td>
<td>0.4697</td>
<td>0.4168</td>
<td>0.4506</td>
<td>0.6172</td>
<td>0.7460</td>
<td>0.6008</td>
<td>0.7597</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LR</td>
<td>0.6396</td>
<td>0.1991</td>
<td>0.3299</td>
<td>0.3385</td>
<td>0.3935</td>
<td>0.6600</td>
<td>0.6527</td>
<td>0.6584</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eclipse 3.0-5</td>
<td>MLP</td>
<td>0.8422</td>
<td>0.6655</td>
<td>0.5770</td>
<td>0.6131</td>
<td>0.7072</td>
<td>0.8042</td>
<td>0.7383</td>
<td>0.8089</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.8114</td>
<td>0.5880</td>
<td>0.5536</td>
<td>0.5928</td>
<td>0.6295</td>
<td>0.7604</td>
<td>0.6880</td>
<td>0.7729</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.8987</td>
<td>0.6741</td>
<td>0.6180</td>
<td>0.6489</td>
<td>0.7408</td>
<td>0.6360</td>
<td>0.6701</td>
<td>0.8373</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LR</td>
<td>0.7652</td>
<td>0.4294</td>
<td>0.4974</td>
<td>0.5208</td>
<td>0.5706</td>
<td>0.6746</td>
<td>0.7095</td>
<td>0.7380</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eclipse 3.0-3</td>
<td>MLP</td>
<td>0.8700</td>
<td>0.6408</td>
<td>0.5602</td>
<td>0.6022</td>
<td>0.6762</td>
<td>0.7483</td>
<td>0.7091</td>
<td>0.7594</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>0.7770</td>
<td>0.5282</td>
<td>0.5046</td>
<td>0.5846</td>
<td>0.6299</td>
<td>0.7201</td>
<td>0.6731</td>
<td>0.7218</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.6875</td>
<td>0.7216</td>
<td>0.5685</td>
<td>0.6667</td>
<td>0.7346</td>
<td>0.7953</td>
<td>0.7548</td>
<td>0.7959</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LR</td>
<td>0.7452</td>
<td>0.4966</td>
<td>0.5245</td>
<td>0.5431</td>
<td>0.6004</td>
<td>0.6980</td>
<td>0.6899</td>
<td>0.7055</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Future work will involve conducting additional empirical studies with data from other software projects and application domains. Additional experiments should also be conducted to analyze the impact of the number of selected features.

References


