Incremental Feature Selection Method for Software Defect Prediction

N. Gayatri, S. Nickolas, A. Subbarao

Abstract— Software defect prediction models are essential for understanding quality attributes relevant for software organization to deliver better software reliability. This paper focuses mainly based on the selection of attributes in the perspective of software quality estimation for incremental database. A new dimensionality reduction method Wilk's Lambda Average Threshold (WLAT) is presented for selection of optimal features which are used for classifying modules as fault prone or not. This paper uses software metrics and defect data collected from benchmark data sets. The comparative results confirm that the statistical search algorithm (WLAT) outperforms the other relevant feature selection methods for most classifiers. The main advantage of the proposed WLAT method is: The selected features can be reused when there is increase or decrease in database size, without the need of extracting features afresh. In addition, performances of the defect prediction models either remains unchanged or improved even after eliminating 85% of the software metrics.

Key Words: Software defect prediction, ANOVA, Wilk's Lambda, incremental feature selection.

1. INTRODUCTION

Defect prediction has lot of importance in software engineering process. An established approach for this task is building software quality prediction model that estimates program module's quality in terms of defect prone and not defect prone [1-5]. Practitioner can apply such models towards implementing a software quality improvement activity for more cost effective usage of the limited project resources. Software qualities of products or process are characterized by software attributes for software development [6]. From the literature, it is understood that focus is given to software metrics like code level metrics and defect data for building these models as it is assumed that these software metrics will confirm the quality of end product. One can build effective defect prediction model by exploring the knowledge from the historical data. Generally defect prediction models formed from previous data available and after validating the model; it is ready to predict the quality in terms of the fault proneness of modules which are under current development. The goal is to obtain high software reliability and quality with effective use of resources. Since quality prediction models are built using the available software metrics and knowledge stored in them, the selection of relevant metrics data becomes an integral and important part of model building in order to achieve high classification accuracies [6]. In other words, the selection of appropriate quality measurement data is necessary for the model to achieve high predictive accuracy. The number of features will also be reduced to lower the classifier's complexity and computational time. Furthermore, as the days go on there, may be a chance of incrementing or updating the database as some changes may occur in the source code in the maintenance phase. In this case when the database gets incremented, accordingly there is a chance that some metrics may lose their strength and some may gain strength for contributing to the final accuracy of the classifier to predict the class. So there is a need of reselecting the important attributes for better accuracy. Hence the process needs to be run from the scratch which is a time consuming procedure. To overcome this disadvantage there is a need for finding the feature selection model which works for incremental databases also.

The paper mainly focuses on

i. Feature selection process is based on the statistical measure (WLAT) and thus improving the quality of software defect prediction models and

ii. Re usage of the selected features for incremental database without any time overhead with enhanced performance.

There are two types of feature selection methods, namely: Wrappers and Filters. Wrapper method chooses the relevant features based on the predictive accuracy of the model [8]. Though the wrapper methods give better performances, they are comparatively computationally expensive. Filter methods select the features without constructing the predictive accuracy of the model, but by heuristically determined relevant knowledge [7] and wrapper method will be too expensive also for incremental database computing.

This paper focuses on the statistical method based on Analysis of Variance Discriminant Analysis (ANOVA DA) Wilk's lambda to select features which can be used when the database is updated. The comparative analysis is done using three feature ranking techniques and validated using six classifiers. For most of the classifiers, features selected through the proposed method gives better performance and reusability of the selected metrics is also shown for

Revised Version Manuscript Received on July 10, 2019.

N. Gayatri, Assistant Professor, Kakatiya Institute of Technology and Science, Warangal, Telangana, India E-mail: gayatriakkala@gmail.com

S. Nickolas, Professor, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India E- Mail: nickolas@nitt.edu

A. Subbarao, Associate Professor, S R Engineering College, Warangal, Telangana, India E-mail: subbarao_ka@yahoo.com

incremental database. Software defect prediction datasets are used for evaluation of feature selection and classification techniques. Further experimental analysis shows that the performance of the model remains almost same even after removing over 85% of the attributes from the original dataset or even improving in some cases. It is uncovered that a reduced search space neither declines the outcomes of the feature subset selection methods, nor does it poorly affect the predictive performances of the classification models. In fact, the subsets of software attributes are obtained from the reduced search spaces and they are more relevant to the class (fault-proneness label) attribute.

The paper is organized as follows. The next section discusses the literature survey of feature selection in software defect prediction followed by the proposed methodology and Experimental Analysis, and finally the paper concludes with a discussion of results.

2. RELATED WORK

Software defect prediction is important for effective resource utilization and for aiding of project managers. Recently Data mining techniques are applied for software defect predictions. The high dimensionality of data is one of the demanding problems in data mining process. It needs lot of computation throughout the learning process and also high dimensionality degrades the performance or results of learning algorithm. One of the effective methods for handling high dimensionality of the data is a feature or attribute selection method which reduces irrelevant and redundant attributes. Feature selection is one of the data preprocessing step which is performed before actual learning task in the data mining process. Feature selection techniques are generally divided into two groups: wrapper-based and filter-based [7-8]. The key factor of the filter-based approach lies in its faster computation compared to wrapper based approach. Many applications of feature selection are available in various fields have been reported [9-12].

Hall and Holmes [13] investigated six feature selection methods which produce ranked attributes and applied these methods on several UCI machine learning repository. The main focus of feature selection is in identifying the condition of patient for cardiac pacemaker implantation and is studied by Ilczuk et al. [14]. The importance of feature selection in text mining is studied by Forman [9]. Recently the feature selection techniques have been used widely in software quality domain for prediction of better software quality and reliability. Rodr'iguez et al. [15] stated that the reduction in feature set or dataset maintained the prediction capability of the original datasets while using fewer attributes. He applied three filter based approaches and two wrapper based models for five software engineering datasets. Importance of feature selection for software cost/effort estimation is studied by Chen et al., [16]. They used COCOCMO-1, COCOCMO-2 datasets and recommended the feature selection in cost modeling, particularly when dealing with very small datasets and concluded that reduced datasets could improve the performance. Pizzi et al. [17] explained a stochastic metric selection method identifies subset which is most effective in prediction of software module complexity. Three benchmark datasets have been used for evaluation of classification method. Jong et al. [18] used SVM's for feature selection in the context of text mining in which attributes have binary value. In a recent study [19], the author has studied feature selection techniques with filter based approaches for defect prediction and concluded that performance of classification models either improved or not effected when only 15% of the original features were included. However the disadvantage of ranking approaches used in filter methods is to decide the numbers of features are to be used for classification. There is no known theory that particular numbers of features are to be used for better classification accuracy. Also, they cannot be used for incremental database as the features themselves may lose their power as the number of instances increased or decreased in the database. This disadvantage is overcome by the proposed method of this paper, and also the time taken is reduced for selection of best features even for the incremental database. In recent years incremental feature learning is being given attention, the learning algorithm is designed with an effective feature selection algorithm which can handle updated features [20]. However, incremental learning cannot handle feature selection for incremental database. Also the incremental feature selection has not been addressed till now as far as literature is considered for software defect prediction and this paper proposes the reusability of the extracted or reduced features by using the statistical feature selection method, which works when additional records or instances are added.

3. RESEARCH METHODOLOGY

This paper proposes a statistical measure based approach using Wilk's Lambda Average threshold Technique (WLAT) which reduces the features in the search space and selects the optimal feature set which can be used for the incremental database also.

3.1 ANOVA Discriminant Analysis

ANOVA Discriminant Analysis (ADA) method uses single dependent continuous variable and make use of more than one independent categorical variable. The ADA is an exceptional statistical method for classification as it provides the relation of features among groups and within-groups. ANOVA is a special form of the General Linear Model. It can be written as

$$y = Xb+e \qquad (1)$$

where, X is a matrix with predictors or Independent Variables (IVs), y is a Dependent Variable (DV), b is a vector of regression coefficients (weightings) and e is a vector of error terms. ADA is a procedure which determines whether differences exist among two or more population means by analyzing the within-group and between group variances. For classification, SPSS tool is used [21], where a dependent variable signifies 0 for Not Fault prone (NFP) and 1 for Fault Prone (FP). The features are treated as independent categorical variables. The ADA classifier predicts the

discriminating power for each feature for classifying the modules either as Fault Prone (FP) or Not Fault prone (NFP).

In this paper, Wilk's lambda measure is used for checking the classification capability of various features. Wilks lambda is very much useful for testing the null hypothesis in which populations have identical means on D (discriminating function). Wilks' lambda is defined as the ratio of within-group sum of squares to the total sum of squares.

Wilk's lambda,

$$\lambda = \frac{ss\_withingroup}{ss\_total} \qquad (2)$$

For each feature, Wilk's lambda measure can be calculated using Eq. (2). The low Wilk's lambda indicates high classification capability of the feature. Hence, the Wilk's lambda statistic measure obtained by ADA is used to calculate the classification capability of each feature and its significance in classification accuracy.

3.2 WLAT Feature Reduction Technique.

The goal of this proposed method is to reduce the number of features used for classification of defects. In this method, the Wilk's Lambda statistics for all features are obtained and average is calculated. The average Wilk's lambda statistic measure is obtained by equation (3)

$$\lambda avg = \frac{\lambda f1 + \lambda f2 + ...\lambda fn}{n} \qquad (3)$$

where λf is Wilk's lambda statistic for each feature. Using the calculated average value, the number of features for classification is reduced. The algorithm for feature reduction is given below.

Algorithm

Input:

1. D Dataset with Features F_j where j=1…. m with
   (i) Each instance $x \in D$ is assigned to one of the two classes $c(x) \in \{FP, NFP\}$;
   (ii) The value of attribute F_j for instance x is denoted as $F_j(x)$;

Method

1. Find the accuracy of D using ADA and store in variable ACC.
2. For each F_j, j=1 …….m do
   Calculate Wilk's lambda $\lambda j = \frac{ss\_withingroup}{ss\_total}$ for all the features
3. Set R $\{\lambda_j\}$ where j=1 to m
4. Calculate average , $\lambda_{avg} = \frac{\sum_{j=1}^{m} \lambda j}{m}$, $\forall$ $\lambda_j \in R$, m= number of features
5. $T \leftarrow \lambda_{avg}$ /* Fix λ_{avg} as the threshold T */
6. Reduce the feature set by considering $\lambda_j \leq T$
7. Store the reduced feature set R where $R = \{\lambda_j / \lambda_j \leq \lambda_{avg}\}$ where j is the index of reduced features
8. Find the accuracy of reduced feature set R using ADA store in ACC1.
9. If $ACC1 \geq ACC$
   Repeat steps 3 to 8 for reduced set R.
10. Else $OPS \leftarrow R$ where OPS is the optimal feature set

Output:

The optimal feature set

This paper considers four datasets; the details of data sets are given in table 4 .For each dataset, λavg is calculated for the features present. According to the algorithm presented above, the number of features is reduced and stored as subset of features as WLAT level 1. Accuracy is found for the subset features, if found good next WLAT level features are selected using the same procedure. The procedure continues until a single feature is obtained or the accuracy for different subsets levels is same. If the accuracy is same or has a slight variation for the subsets, then the subsets with minimum number of features are selected for experimental purpose and better results are achieved. The details of number of features selected through WLAT method for PC1 dataset are given in Fig. 1, the different WLAT levels are obtained and also feature set with minimum number of features is selected according to the accuracy of the feature set. From the Fig. 1, it is understood that only four features are enough for better prediction and collection of those four features are sufficient for that particular project. This will further improve the classification speed and also relatively reduces the computational cost and complexity. Also, this method can be successfully be implemented for incremental database also.

In WLAT Level 1:
λ avg = 0.811
Number of features (Reduced set) =13 (out of 21)
Accuracy = 93.4%

In WLAT Level 2:
λ avg = 0.657
Number of features (Reduced set) = 6 (out of 13)
Accuracy=93%

In WLAT Level 3:
λ avg= 0.6
Number of features (Reduced set) = 4 (out of 6)
Accuracy=93.4%

Fig. 1 Classification accuracies of original and reduced features

TABLE 1. Details of number of features selected and their Wilk's lambda and their accuracies of different datasets

Dataset name	KC2	JEDIT 3.2	PC4	PC1
Original features &accuracy	21/21	20/20	37/37	21/21
	84.4	84%	89.7%	84.4%
λ_{avg}	0.85	0.95	0.97	0.811
	0.81	0.89	0.95	0.677
WLAT level 1	12/21	9/20	17/37	13/21
	84.4%	79.4%	89.0%	84.4%
WLAT level2	6/12	5/9	6/17	6/13
	84.4%	79.8	88.7%	84.4%
WLAT level 3	4/6	3/5	6/6	4/6
	84.4%	79.0%	88.7%	84.4%

In the table1, classification accuracies for the original feature set and reduced feature set is shown for different datasets. For some datasets even though there is a slight variation between original feature set and reduced feature set (0.2-0.3), the minimum number of features are selected as optimal features because the difference is negligible. Experiments are done with those features and better performance is observed.

From the table 6, it is observed that for KC2 dataset, only four features out of twenty one are selected. In JEDIT 3.2 dataset only 5 out of 20 are selected as optimal features and for PC4 dataset only 6 features are selected out of 37 features as optimal features.

4. EXPERIMENTAL DESIGN

4.1 Feature Selection Techniques and Classification Models Considered

In this paper, many feature ranking techniques are used, and all these are based on filter-based feature ranking techniques. In these types of filters, no learners will be involved. Descriptions about these techniques are specified in the next subsections.

Feature ranking techniques evaluate attributes based on certain specified criterion and rank the attributes accordingly. However, it was observed that sometimes an attribute may not be useful by itself, but it will have greatest impact when it is combined with other attributes. Feature subset selection approach of by searching and selecting subsets of attributes that collectively have good performance. Attribute selection can also be divided into wrappers and filters [22]. This paper considers only filters instead of wrappers for feature selection for comparison with the proposed work. The reasons include that (1) the use of wrappers would be too complex for inclusion in our case study; (2) Though they give better results comparatively they are computationally costly; and (3) the wrappers depend on specific learner and this will make difficult to find best suitable wrapper since many learners are available to choose from.

4.2 Feature Selection techniques

For the experimentation, Chi square method (CS)[23], RELIEF [25] Support Vector Machines (SVM)[26] filter methods are used , WEKA, an open source data mining tool is used. All the classifiers and feature selection techniques are experimented using default parameters in WEKA [24].

4.3 Classifiers

The six classifiers used in this study are Naïve Bayesian (NB), Multi-Layer Perceptron (MLP), SMO, Classification via Regression (CvR), Radial Bias Function Network (RBFN) and Instance based K Nearest neighbor (IBK) [27-32]. These techniques can be used based on their use in software engineering. The fact is that, they do not have a built-in attribute selection capability. In general, default settings of these learners are considered as specified in WEKA [24]. However, changes to default parameters were obtained in response to significant improvement in classifier performance.

4.4 Performance Metrics

Different performance measures shown in table 3 are used in data mining to evaluate the classifier. For classifier prediction, there are four possible outcomes: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Using these, a two by two matrix called confusion matrix is formed which is described in table2. The four values |T P|, |T N|, |FP|, and |FN| are specified by the confusion matrix and they become basis for several other performance metrics. These metrics are frequently used in data mining and machine learning.

TABLE 2. Confusion matrix

	Obtained result	
	+	-
Correct result		
+	TP	FP
-	FN	TN

By combining values in the confusion matrix in table 2 composite measures are calculated. In case of imbalanced data which is proportional to the class distribution (where code is viewed in two classes – faulty or non-faulty), the above measures are very useful.

Accuracy (acc) [33], Area under Curve (AUC) [34],recall and precision are widely used in all kinds of data mining classifiers.

TABLE 3. Different performance metrics

Recall pd (probability of detection) Sensitivity	TP / (,TP + FN)
Precision	TP / (TP + FP)
pf (probability of false alarm)	FP / (FP + TN)
Specificity	TN / (TN + FP)
F-measure	2 x Recall x Precision)/ (Recall+Precision)

Accucy	(TN + TP) / (TN + FN + FP +TP)
Receiver operating characteristic	A graphical plot of the sensitivity (or pd) vs. specificity (or pf)

This paper uses ROC as the performance measure for the evaluation of the proposed method

5. EXPERIMENTAL ANALYSIS RESULTS

5.1 Software Measurement Data

Experiments are conducted and defect data collected from four real-world software projects, including NASA software projects. PC1, PC4, KC2 and JEDIT3.2 have been used in this research work. These projects include static code features and combination of static and design features along with object oriented metrics. These are combination of product, process and execution metrics. In a program module, the dependent variable is the class of that program module. A module with one or more faults is considered fp (fault prone) and nfp (not fault prone) otherwise. For JEDIT, we prepared the dataset and assigned final class by considering the number of bugs in the last column. If at least one bug is present, then that module is considered as fault prone else not fault prone. PC1, KC2 contains 21 features, JEDIT contains 20 features and 1 class and these are object oriented features and PC4 contains 37 metrics which are combinations of different metrics. Description of datasets is given in table 4.

5.2 Experimental Results

The practitioner should select number of features for classification before operating on feature ranking technique. From the literature survey, it is understood that, no guidance is provided for selecting the appropriate number of features.

TABLE 4. Dataset description

Dataset name	No of attributes with instances	No of defective%	No of not-defective%
PC1	21/947	6.94	93.05
KC2	21/487	16.5	84.5
JEDIT .2	20/333	40.3	59.7
PC4	37/1255	20	80

To construct Random Forests learners for binary classification, a recent study [35] recommended using features for imbalanced data sets where n is the total number of the independent attributes. Moreover, a preliminary investigation showed that features are appropriate for various learners. Consequently, attributes are chosen for the evaluation purpose. Accordingly the numbers of features are selected for the datasets used.

In this paper, WLAT method is proposed for feature reduction and to select optimal subset features through which search space may be reduced. WLAT is a statistical measure based on ADA. The features selected through the proposed method are compared with the other feature selection method using different classifiers. With four datasets and six classifiers, 12 subsets have been constructed and evaluated where WLAT method performs better in most of the cases or at least comparable. The advantage of WLAT lies in the extraction of features that can be reused again in future when the database gets updated without extracting the components again and again. Generally, as the database increases, the power of the features also changes and accordingly some features may become less important even if they had highest importance before updating. Hence, to achieve better classification accuracy, high power features have to be selected and feature selection must be done from the beginning, which is time consuming because same task has to be repeated for the same number of attributes. But in case of WLAT even if the database is updated, the previously selected features can be used as the best features and high accuracy can be obtained using them and time for feature selection can be reduced. This method also can be called as Hybrid search algorithm because it reduces the search space according to the WLAT method and then selects an optimal feature subset which gives highest classification/prediction accuracy. The other feature selection algorithms are based on ranking of features. Hence, as the database gets incremented ranking of features differs and feature selection process is to be performed again to select the best features.

5.3 Results of the Feature Selection Techniques

In order to investigate the performance of our algorithm, the six different models are constructed with data sets consists of selected attributes only. AUC performance metric can be used to evaluate the defect prediction models. The classifier performance results are presented in Figures 2-5. In the experiments, ten-fold cross-validation is used for model training. The values presented in the graphs represent the average AUC for every classification model constructed over the ten-fold cross-validation. All the results of four feature selection techniques over four different software data sets are reported. The results are pictorially represented and they show that proposed method is better in most of the cases and for others it is equally comparable.

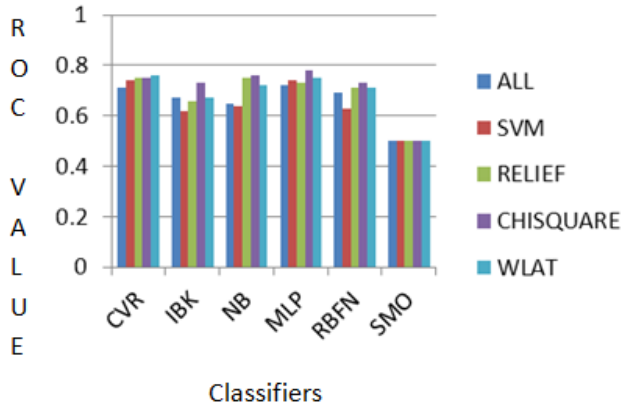

Fig. 2. Performance in terms of AUC for PC1 dataset

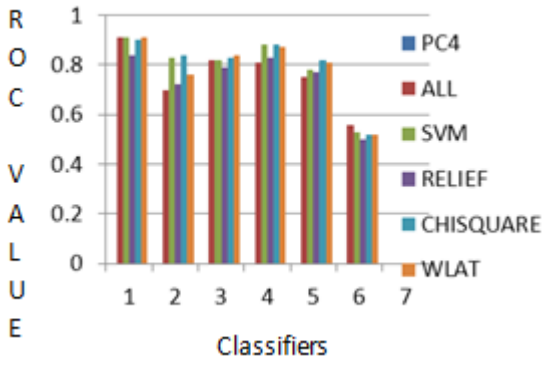

Fig. 5. Performance of PC4 in terms if AUC

Fig. 3 Performance in terms of AUC for KC2

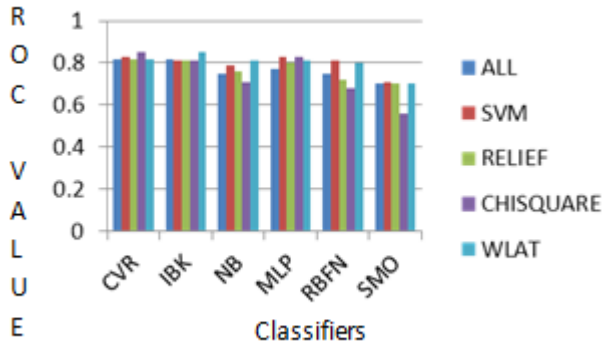

Fig. 4 Performance of JEDIT in terms of AUC

From the graphs, it is concluded that features selected from the proposed (WLAT) method achieves better AUC values for most of the classifiers, as the features depends on the grouping variable i.e., class variable as it computes the power of attributes using within group variance. Comparatively, in some cases features selected through Chi square [] feature selection technique also has performed better. Hence it can be considered as a good feature selection technique for software defect prediction. The features selected through the other two approaches, i.e., SVM and RELIEF also show good performance comparatively. The AUC values with selected features for different classifiers obtained are in the range of (0.6-0.9). This shows that the selected features are good predictors and can be used for effective defect prediction. Considering Classifiers, SMO performs worst as its AUC is 0.5 and it is constant throughout the database for all classifiers, hence it cannot be practically used for defect prediction. Naive Bayes algorithm performs better for all the datasets as it is highly affected by the percentage of defect modules. Classification via Regression, RBFN and MLP- a neural network algorithm achieves better performance comparatively. The performance of IBK depends purely on the dataset as it selects the nearest neighbors and performs prediction. Hence results may vary according to the data considered for training the model. Other than SMO all algorithm can be considered as good defect prediction models. It can also be mentioned that the classifiers performance is increased even though 85% of attributes are deleted. It is also understood that classifiers performance depends on the dataset it uses and the good feature selection technique. Though the features selected from Chi-square and Relief methods give better performance for some of the algorithms, they cannot be used for incremental database.

5.4 Analysis for Incremental Databases

In future, the database may be updated with some new records as the software keeps on changing. Updated database may be available for prediction. If the features are to be selected from the updated database, feature selection process has to be done for finding efficient features as a feature's contribution for final accuracy changes as the number of rows

are added to the database. The disadvantage of the feature ranking algorithms is they cannot be used for incremental database because the ranking of features changes as the data gets added. Reselection has to be done from scratch, which is a time consuming process. The proposed overcomes this advantage.

The proposed WLAT method allows reuse of the features without extracting again and again for the same database even though it is updated. This is because ADA Wilk's lambda of the features is calculated based on the within group statics by computing each feature a number of times. The average of Wilk's lambda of all the features is taken each and every time and the best features selected based on the accuracy. Once the best features are selected, they can be used even though the database is updated without re performing the feature selection process. This will reduce the time consuming process. Here experiments are done on three datasets. The datasets are updated by adding 10%, 20%, and 50% records to the original data, for updated database the proposed process is applied. The same number of attributes selected before updating the database has been reselected again after updating the database through WLAT method. Considering other feature selection techniques based on ranking, the different features are selected. Experimental results show that performance also increased with the same features using the proposed method. Top ranked attributes have been used for comparison according to features concept for other feature selection techniques. The below tables 5-7 show the attribute id's for each dataset which participate in the feature selection process.

In the feature selection process, the rank of attribute id's change after updating records in the database. The databases are updated by adding different number of records to them.

TABLE 5. PC1 dataset

Feature selection algorithm	Id's of Previous No. of attributes and attributes after updating the records
WLAT	(2,19,16,17)
SVM	(4,7,14,,5,2)/(4,7,14,1,9)
RELIF	(36,10,23,17,15)/(36,10,17,23,15)
Chi-square	(4,36,28,27,10)/(4,36,28,6,9)

TABLE 6. KC2 dataset

Feature selection algorithm	Id's of Previous No. of attributes and attributes after updating the records
WLAT	(1,18,16,17)/(1,18,16,17)
SVM	(11,16,20,18)/(6,20,11,8)
RELIF	(7,17,14,16)/(7,17,14,9)
Chi-square	(14,16,15,1)/(14,15,16,18)

TABLE 7:PC4 dataset

Feature selection algorithm	Id's of Previous No. of attributes and attributes after updating the records
WLAT	(18,1,7,8)
SVM	(18,3,15,16)/(21,18,16,3)
RELIF	(7,17,8,9)/(7,17,9,8)
Chi-square	(20,18,19,6)/(20,18,11,19)

From the tables 5-7, it is understood that there is slight variation in the ranking of attributes for other feature selection techniques, but still the whole process has to be done because it is not fixed that only those attributes get selected. The proposed method overcomes the above problem by using WLAT technique. Hence it can be concluded that the proposed method works well for many classifiers without reselecting the features. The ROC values for different classifiers using different feature selection techniques over the datasets are presented in table 8. The performance of the features selected through proposed method obtains better or equally comparable results with the same number of previously selected attributes. For almost all classifiers, there is an increase in the accuracy with same attributes even when the database is updated compared to original feature set. The very little difference is seen for the proposed feature selection method compared to other feature selection methods; this can be ignored when the time taken for selecting the attributes for existing methods is considered.

The main advantages of WLAT method are

i. Achieving good classification accuracy for most of the classifiers.

ii. Minimum number of features is selected for classification purpose.

iii. Re selecting or Re-extracting of features is avoided for updated database.

iv. It can be successfully used for incremental database.

TABLE 8: Performance of feature selection techniques for incremental database in terms of AUC

PC1ADDED	ALL	SVM	RELIEF	CHISQUARE	WLAT
CVR	0.71	0.81	0.83	0.81	0.81
IBK	0.67	0.78	0.79	0.79	0.79
NB	0.65	0.68	0.75	0.75	0.73
MLP	0.72	0.78	0.76	0.75	0.78
RBFN	0.69	0.66	0.72	0.73	0.71
SMO	0.5	0.5	0.5	0.5	0.5
KC2ADDED	ALL	SVM	RELIEF	CHISQUARE	WLAT
CVR	0.83	0.83	0.82	0.84	0.83
IBK	0.58	0.63	0.64	0.65	0.65
NB	0.85	0.8	0.84	0.8	0.85
MLP	0.84	0.86	0.84	0.86	0.84
RBFN	0.8	0.78	0.82	0.77	0.83
SMO	0.52	0.53	0.5	0.51	0.5
PC4ADDED	ALL	SVM	RELIEF	CHISQUARE	WLAT
CVR	0.91	0.91	0.83	0.88	0.9
IBK	0.7	0.91	0.8	0.88	0.84
NB	0.82	0.82	0.79	0.81	0.83
MLP	0.81	0.87	0.83	0.84	0.86
RBFN	0.75	0.82	0.77	0.77	0.79
SMO	0.56	0.54	0.5	0.52	0.52

6. CONCLUSION

Before training a specified defect prediction model, clever selection of software metrics improves the end result by avoiding redundant and less important features. WLAT feature dimension reduction approach is used in this paper for decreasing the number of features effectively used for defect classification. The proposed WALT method has reduced features according to different datasets and WLAT levels and the classification accuracy obtained is encouraging for different databases. The classification accuracy is compared with recently proposed feature selection techniques such as SVM, Chi-square etc.

The proposed method gives better or same classification accuracy when compared to other methods. The main benefit of the proposed WLAT method is that the preferred selected features can be reused when there is increase or decrease in database size, without extracting components every time. This is shown through the experiments conducted on the updated database. Different feature selection techniques and classifiers have been investigated and concluded that proposed method is better for most of the classifiers. Also, it is concluded that the

performance of classifiers has no significant change and sometimes better even after deletion of 85% of the attributes. Hence the best feature selection technique should be used for better classifier performance irrelevant of the data. In the future, much investigation has to be done using different datasets with the proposed method and many feature selection techniques have to be investigated. It is much better if generalized feature selection technique is proposed which is independent of the dataset.

REFERENCES

1. Khoshgoftaar T M, Bullard L A, Gao K. Attribute selection using rough sets in software quality classification. International Journal of Reliability, Quality and Safety Engineering 2009, 16(1): 73–89.
2. Lessmann S, Baesens B, Mues C, Pietsch S. Benchmarking classification models for software defect prediction: A proposed framework and novel findings. IEEE Transactions on Software Engineering, 2008, 34(4), 485–496
3. Meulen M J, Revilla M A. Correlations between internal software metrics and software dependability in a large population of small C/C++ programs. In: Proceedings of the 18th IEEE International Symposium on Software Reliability Engineering, 2007, 203–208.
4. Rodriguez D, Ruiz R, Cuadrado-Gallego J, Aguilar-Ruiz J. Detecting fault modules applying feature selection to classifiers. In: Proceedings of Eighth IEEE International Conference on Information Reuse and Integration 2007, 667–672.
5. Sunghun K, Zimmermann T, Whitehead, E J, Zeller A. Predicting faults from cached history. In: Proceedings of the 29th International Conference on Software Engineering, 2007, 489–498.
6. Pfleeger, S L. Software metrics: Progress after 25 years. IEEE Software 2008, 25 (6): 32–34.
7. Ooi C H, Chetty M, Teng, S W. Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets. Data mining and Knowledge Discovery, 2007, 329-366.
8. John G. H, Kohavi R, Pfleger K. Irrelevant Features and Subset Selection Problem. In: Proceedings of the Eleventh International Conference of Machine Learning, Morgan Kaufmann Publishers, 1994, 121-129.
9. Forman G.. An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research, 2003, 3: 1289–1305.
10. C, Serafini M, Merler S, Jurman G.. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. BMC Bioinformatics, 2003, 4: 54.
11. Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of Machine Learning Research, 2003, 3: 1157-1182.
12. Doraisamy S, Golzari S, Norowi N M, Sulaiman N, Udzir N I. A study on feature selection and classification techniques for automatic genre classification of traditional malay music. In: Proceeding of the Ninth International Conference on Music Information Retrieval, Philadelphia, 2008, 331–336.
13. Hall M A, Holmes G.. Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(6): 1437–1447.
14. Ilczuk G., Mlynarski R, Kargul, W, Wakulicz-Deja. A. New feature selection methods for qualification of the patients for cardiac pacemaker implantation. Computers in Cardiology, 2007, 423–426.
15. Rodriguez D, Ruiz R, Cuadrado-Gallego J, Aguilar-Ruiz J, Garre M. Attribute selection in software engineering datasets for detecting fault modules. In: Proceedings of 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, 418–423.
16. Chen Z, Menzies T, Port D, Boehm B. Finding the right data for software cost modeling. IEEE Software, 2005, 22(6): 38–46.
17. Pizzi N J, Demko A B, Pedrycz, W. The analysis of software complexity using stochastic metric selection. Journal of Pattern Recognition Research, 2011, 1: 19–31.
18. Jong K, Marchiori E, Sebag M, van der Vaart. A feature selection in proteomic pattern data with support vector machines," Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, San Diego, CA, 2004.
19. Gao K, Khoshgoftaar T M, Wang H, Seliya N. Choosing software metrics for defect prediction: an investigation on feature selection techniques. Software: Practice and Experience. Special Issue: Practical Aspects of Search-Based Software Engineering, 2011, 41(5): 579– 606.
20. Xinwang Liu, Guomin Zhang, Yubin Zhan, and En Zhu. An Incremental Feature Learning Algorithm Based on Least Square Support Vector Machine. FAW 2008, LNCS 5059, Springer-Verlag Berlin Heidelberg, 2008, 330–338.
21. SPSS Data Mining, Statistical Analysis Software, Predictive Analysis, Predictive Analytics, Decision Support Systems. http://www.spss.co.in
22. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(4): 491–502.
23. Han J, Kamber M. Data Mining: Concepts and Techniques. San Francisco, Morgan Kaufmann Publishers, 2001.
24. EKAhttp://www.cs.waikato.ac.nz/ml/weka.
25. Marko Robnik-Sikonja, and Igor Kononenko. An adaptation of Relief for attribute estimation in regression. In: Fourteenth International Conference on Machine Learning, 1997, 296-304.
26. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Machine Learning , 2002, 46: 389-422.
27. Aha D W, Kibler D, Albert M K. Instance-based learning algorithms. Machine Learning, 1991, 6(1): 37–66.
28. Haykin S. Neural Networks: A Comprehensive Foundation. 2nd edn, Prentice-Hall, Englewood Cliffs, NJ, 1998..
29. Shawe-Taylor J, Cristianini N. Support Vector Machine. 2nd edn, Cambridge University Press, Cambridge, 2000.
30. John G H, Langley P. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, 1995, 2: 338–345.
31. Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 1997, 29(2): 103–130.
32. Le Cessie S., Van Houwelingen J C. Ridge estimators in logistic regression, Applied Statistics, 1992, 41(1): 191–201.
33. Ma Y, Cukic B. Adequate and precise evaluation of predictive models in software engineering studies. In: Proceedings of the PROMISE workshop, 2007.
34. Promise Software Engineering. http//promise. site, Ottawa. CASE Repository
35. Khoshgoftaar T M, Golawala M, Van Hulse J. An empirical study of learning from imbalanced data using random forest. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society, 2007, 2: 310–317.