A Novel Feature Selection Method Based on Maximum Likelihood Logistic Regression for Imbalanced Learning in Software Defect Prediction

Kamal Bashir[1], Tianrui Li[1], and Mahama Yahaya[2]
[1]School of Information Science and Technology, Southwest Jiaotong University, China
[2]School of Transport and Logistics Engineering, Southwest Jiaotong University, China

Abstract: *The most frequently used machine learning feature ranking approaches failed to present optimal feature subset for accurate prediction of defective software modules in out-of-sample data. Machine learning Feature Selection (FS) algorithms such as Chi-Square (CS), Information Gain (IG), Gain Ratio (GR), RelieF (RF) and Symmetric Uncertainty (SU) perform relatively poor at prediction, even after balancing class distribution in the training data. In this study, we propose a novel FS method based on the Maximum Likelihood Logistic Regression (MLLR). We apply this method on six software defect datasets in their sampled and unsampled forms to select useful features for classification in the context of Software Defect Prediction (SDP). The Support Vector Machine (SVM) and Random Forest (RaF) classifiers are applied on the FS subsets that are based on sampled and unsampled datasets. The performance of the models captured using Area Ander Receiver Operating Characteristics Curve (AUC) metrics are compared for all FS methods considered. The Analysis Of Variance (ANOVA) F-test results validate the superiority of the proposed method over all the FS techniques, both in sampled and unsampled data. The results confirm that the MLLR can be useful in selecting optimal feature subset for more accurate prediction of defective modules in software development process.*

Keywords: *Software defect prediction· Machine learning· Class imbalance· Maximum-likelihood logistic regression.*

Received April 30, 2018; accepted January 28, 2020
https://doi.org/10.34028/iajit/17/5/5

1. Introduction

Software development is inevitably subject to defects. Predicting these defects is the holy grail of most software development entities. The cost of finding and rectifying software defects is estimated at billions of pounds per year. Thus, many research efforts are focused on finding efficient approaches that can reliably predict defective modules. Software Defect Prediction (SDP) applies historical defect data often obtained from software depositories to forecast defective modules in software development process for more assured quality service delivery. When defect prediction is considered as a binary classification problem in machine learning domain, prediction model developed with software metrics (i.e., features) identifies new software modules as possible defect or not [10, 22]. However, datasets obtained from the depositories come with high dimensional feature space, many of which are irrelevant and redundant. Irrelevant features consist of those that offer no useful information for the classification task, and redundant features present the same information as the currently selected features. Model built on training data with irrelevant and redundant features increases the model run time, deteriorate predictive performance, and increases the model complexity. Therefore, Feature Selection (FS) has become an indispensable part of data mining. FS is the process of identifying relevant features and eliminating irrelevant, redundant, or noisy data. In the perspective of classification for SDP, Yu *et al*. [28] asserted that FS aids in reducing computation requirement, minimizing the effect of the curse of dimensionality and improving the model performance.

In the perspective of supervised inductive learning in which SDP models are often situated, FS presents a set of candidates features by applying one of the three techniques:

1. The specified number of features subset that enhances an evaluation measure.
2. The minimum number of the subset that satisfies certain criteria on evaluation measures.
3. The subset with the best assurance among size and evaluation measure.

In view of the aforementioned approaches, Kumar and Minz [17] noted that appropriate use of FS algorithms improves inductive learning, either in term of generalization dimensions, learning speed, or reducing model complexity.

FS becomes critical when the number of samples is much less than the number of features. In this case, the learning becomes mainly difficult, as the search space will be sparsely populated. As a result, the search

criteria will fail to accurately discriminate between noise and relevant data [26]. Kumar and Minz [17] considered the approaches to FS under to broad categories which include individual evaluation and subset evaluation. Guyon and Elisseeff [9] referred to feature ranking as individual evaluation approach. In ranking features, the individual weight is given according to its level of relevance. In subset evaluation, candidate feature subsets are developed using search criteria. Regardless of the evaluation technique, the FS procedure generally consists of four major steps including subset generation, subset evaluation, stopping criteria and validation of the result.

In the subset generation, a heuristic search is conducted in which a candidate subset is specified in each state for evaluation in the search space. A subset generated must be assessed by certain evaluation criteria. Hence, many criteria for feature subset evaluation have been proposed in the literature to determine the suitability of the candidate subset. According to Liu and Yu [20], the FS evaluation criteria can be classified by their reliance on mining algorithms as dependent and independent criteria. Independent criteria exploit the essential characteristics of the training data without involving any mining algorithms to evaluate the goodness of a feature set or feature. And dependent criteria include predetermined mining algorithms for FS in which features are selected based on the performance of the mining algorithm used to the subset of features selected. Finally, a stop criterion is ascertained to end the selection process. The process of FS ends at validation system. Though the validation system is not intrinsic in the FS process, FS technique ought to be validated by conducting experiments and comparing results with established methods using either artificial or real-world datasets, or both.

The relation between the inductive learning approach and FS algorithm deduces a model. In this regard, there are three general approaches in the literature to deal with the FS task. First, the filter methods select the features by ranking them by their usefulness in predicting the target concept. To estimate their ranks, statistical test and correlation results (i.e., Chi-square, ANOVA, Pearsons correlation) are employed. The second approach is wrapper method, which generates different subsets of features and searches for the optimal feature subset adjusted to the particular learning algorithm [16]. The best subset is selected by testing the algorithm. Different criteria such as forward and backward selection are used to select the features for the subsets. Finally, the embedded approach is a hybrid between the ranker and the wrapper methods. For a more comprehensive review of the FS methods, interested readers can refer to [4]. The proposed FS based on Maximum Likelihood Logistic Regression (MLLR) method in this paper in the context of SDP can be categorized under embedded methods.

Logistic regression was first formulated by statistician Cox in 1958 as an approach to statistical data analysis and used extensively in several fields, including machine learning [18]. When used for prediction, logistic regression fits data to the logistic curve to develop a model for predicting future data. It requires the fitted model to well-match the data. The maximum likelihood estimation is applied to find the parameters that maximize the probability of observing the data [8]. To achieve these parameters, the likelihood function is developed to express the probability of the observed data as a function of the unknown parameters. The values that maximize this function are then chosen as the maximum likelihood estimators. This measure gives an approximation of the conditional probability that the outcome variable takes the value of 1 (faulty module), for a given feature. In order to select features, a statistical test is conducted to identify features with non-zero estimators (coefficients) at a given confidence interval. These features are selected as the optimal subset to be used to develop the model.

Despite the numerous FS techniques and the benefits, a number of data issues can pose challenges and make the selection tasks harder. Prominent among these challenges are data imbalance and noisy. For a binary class data, class imbalance occurs when the number of samples in one class (i.e., Non-Defect-Prone (NDP)) is far more than those of the other class (i.e., DP). The effect of data skews may vary according to the imbalance ration. High class imbalance, often observed in software defect datasets, makes identification of the minority class by any statistical learning method very difficult and challenging. The reason is that a high-class imbalance presents a bias in favour of the majority class. Therefore, it becomes quite hard for the majority of statistical learning methods to effectively distinguish between the minority and majority classes, yielding a task akin to searching for the proverbial needle in a haystack. The biased learning introduced by the data skew may lead to the selection of wrong features/ feature subset that are incapable of predicting the minority class samples. In SDP, where the occurrence of false-positive (i.e., wrongly predicting defective module as NDP) is rather expensive than false negative (i.e., wrongly predicting NDP module as DP), a learner's prediction that inclined to the majority class could produce unfavourable results [15].

Due to the above, we propose a hybrid FS method where data balance treatment is carried out prior to FS, and the capability of different FS methods including Chi-Square (CS), Information Gain (IG), Gain Ratio (GR), RelieF (RF), Symmetric Uncertainty (SU) and MLLR is assessed. In the next parts, we elaborate on the MLLR technique and briefly introduce the feature

ranking methods that are employed for the comparison. The superiority of the proposed method is noted and some insight in the perspective of SDP is shared.

The remainder of this paper is organized as follows: Section 2 presents the methods and techniques used in this paper. Section 3 presents the experimental design. Section 4 reports our results and discussion. Finally, the conclusion and future work are given in section 6.

2. Methodology

2.1. Feature Selection

FS as a method for reducing the attribute space of a variable set, is an essential component of both supervised and unsupervised classification and regression problems [1, 2, 11, 12, 13, 24, 25, 26, 29]. It is significant for three main reasons as outlined in section 1. In this research, we propose the MLLR FS techniques in which the optimal feature subset is through the Wald test to verify the coefficients estimated (at 95% confidence interval), based on which important features are selected. The technique of feature ranking is to score each attribute based on a particular measure, to distinguish and choose the best subset of features. This research uses five widely used filter-based feature ranking methods: CS, IG, GR, RF and SU. In the next parts, we elaborate the MLLR technique and briefly introduce the feature ranking methods that are employed for the comparison.

2.1.1. The Logistic Regression

In binary logistic regression, a dependent variable is given by Y_i (i = 1....n) ~ Bernoulli (Y/p_i) so that it takes on a value of 1 with probability p_i and 0 with probability 1-p_i over n trials. The vector of input software metrics (features) is given by xi and pi varies over this explanatory space such that

$$p_i = \left(\frac{1}{1+e^{-x_i\beta}} \right) \tag{1}$$

If Y_i is considered as a latent continuous variable Y_i (e.g., the probability of DP) distributed based on logistic density function with mean μ_i, then

$$Y_i^* \sim Logistic(Y_i^* / \mu_i) \tag{2}$$

$$\mu_i = x_i\beta, \tag{3}$$

Where (Y_i^* / μ_i) is considered one-parameter logistic PDF,

$$Y_i^* = \left(\frac{e^{-Y_i^* - \mu_i}}{\left(1+e^{-Y_i^* - \mu_i}\right)^2} \right) \tag{4}$$

Then the probability of observing the dichotomous insight of Y_i^* is given by

$$\begin{aligned} P_r(Y_i = 1/\beta) &= p_i = P_r(Y_i^* > 0/\beta) \\ &= \int_0^\infty Logistic(Y_i^* / \mu_i) dY_i^* = \frac{1}{1+e^{-x_i\beta}} \end{aligned} \tag{5}$$

This is the more general binomial case of n Bernoulli trials of Y_i over the vector x_i.

1. The maximum likelihood estimators

   The parameters for the logistic regression model discussed above are estimated by maximum likelihood, where the likelihood function is given by

$$-\sum_{i=1}^{n} \ln\left(1+e^{(1-2Y_i)x_i\beta}\right) \tag{6}$$

The principle of maximum likelihood requires that we use the value that maximizes the expression in Equations (6) as our estimate of β. To find the value of β that maximizes the likelihood function we differentiate the likelihood function with respect to β and equate the expressions to zero. These equations are as follows:

$$\sum_{i=1}^{n} [Y_i - p_i(x_i)] = 0, \tag{7}$$

and

$$\sum_{i=1}^{n} x_i [Y_i - p_i(x_i)] = 0 \tag{8}$$

The above equations are referred to as the likelihood equations. In linear regression, the likelihood equations, obtained by differentiating the sum of squared deviations function with respect to β are linear in the unknown parameters, and therefore are easily solved. For logistic regression, Equations (7) and (8) are nonlinear in β, and thus need distinct approaches for their solution. These approaches are iterative in nature and have been programmed into available computer software. McCullagh and Nelder [21] discussed the iterative approaches applied in most software programs. Specifically, they showed that the solutions to Equations 7 and 8 may be found using a generalized weighted least squares method. In this paper, we apply the generalized linear model in R statistical program to access the logistic regression model results.

When the class distribution in the data is approximately balanced, maximum likelihood estimates are reliable and asymptotically efficient. However, this is not the case when an extreme imbalance exists between classes in the data, as in crash data with a small number of fatal injury samples. The use of data with imbalanced class distribution results in low estimates of $P_r(Y_i = 1/x_i)=p_i$ due to the structure of the variance matrix shown below:

$$V\beta = \left[\sum_{i=1}^{n} p_i(1-p_i) x_i' x_i \right]^{-1} \tag{9}$$

The product of the probability of DP and NDP denoted by $p(1-p_i)$ which is a component of this matrix is affected by class imbalances in the software defect data. Therefore, it can be difficult to accurately predict given an instance, whether the software has DP or not. This is because the predicted probabilities of true DP returned by the model will be closer to 0 than to 0.5. As a result, we apply the Synthetic Minority Oversampling Technique (SMOTE) algorithm [5] to create new samples for the minority class.

The purpose is to augment the size of minority class samples and increase the sensitivity of methods that require statistical significance. This approach has been applied in many studies in the literature.

2. FS based on MLLR

One approach to test the significance of a feature in any model relates to the question of its relevance in predicting the outcome variable. This question is answered by conducting one among several statistical tests to verify the reliability of the coefficients estimated. One of the test which is considered in this study is the Wald test. Thus, we refer to FS based on the Wald test on estimated feature coefficients (at 95% confidence interval) as the MLLR FS method. The Wald test value is obtained by comparing the maximum likelihood estimate of the coefficient β_1 with the estimate of its standard error and expressed as

$$W = \frac{\beta_1}{se.\beta_1} \tag{10}$$

Under the null hypothesis that β1=0, follows a standard normal distribution. Failure to reject the null hypothesis suggest that the feature is not significant in determining the outcome. These features are accordingly left out and the feature for which their estimated coefficients are found to be significantly greater than zero are chosen as optimal subset and allowed to take part in the model development.

2.2. Feature Ranking Methods

The CS analysis is used to examine whether the two variables are independent. The idea of entropy from information theory is applied to measures IG, GR, and SU. IG measures the decrease in the weighted medium impurity of the separations, compared with that of the full set of data. RF is an instance-based feature ranking method. Its advantages are that it is not dependent on heuristics, runs in low-order polynomial time, and is noise-tolerant and robust to feature cooperation. SU emerges from the modification of IG to take care of the bias to features that have a lot of values.

2.3. Synthetic Minority Oversampling Technique (SMOTE)

A number of data sampling techniques have been studied in the literature, including both majority under sampling and minority oversampling techniques. This study applied the SMOTE, which works on creating new synthetic examples in minority classes. The synthetic examples generated operate in feature space rather than data space. The SMOTE samples are linear combinations of two similar samples from the minority class (X and X_0) and are defined as $S = S + u * (X_0 - X)$ with $0 \leq u \leq 1$. X_0 is randomly chosen among the K minority class nearest neighbours of X. The newly built examples decrease rarity in minority and make it fuller and more general. This study adopts (35:65) and (50:50) minority: majority ratio as Imbalance threshold as proposed in [21].

2.4. Classifiers

The two learners that are chosen for building the software quality prediction models are Support Vector Machine (SVM) and Random Forest (RaF). We apply the Waikato Environment for Knowledge Analysis (WEKA) tool to achieve these classifiers. The SVM is a linear discriminant classifier which assumes that the best discriminant maximizes the distance between the two classes. This is measured in the distance from the discriminant to the samples of both classes. Two changes are made to the default parameters of the SVM learner in WEKA: the complexity constant c is set to 5.0 and the buildLogisticModels parameter is set to true. By default, a linear kernel is used.

Random forests are an ensemble learning approach for classification, regression and other tasks that function by creating a multitude of decision trees at training time and coming out with the modal (classification) or mean prediction (regression) of the individual trees.

In this study, the RaF learner is adopted to construct software defect models. The choice of RaF is based on its best accuracy and capacity to efficiently run on large database relevant to current algorithms. For the implantation of RaF, default parameter settings are adopted as detailed in WEKA.

2.5. Performance Index

The Area Under the Receiver Operating Characteristics Curve (AUC) is the common proper metric applied to accurately assess the performance when imbalanced data is presented with unequal error cost [8]. The Receiver Operating Characteristic (ROC) curve plots true positive rate on the y-axis versus the false positive rate on the x-axis. The curve indicates the trade-off between detection rate and false alarm rate. The AUC, which is calibrated over the range of 0 to 1, provides a single numerical metric for evaluating model

performances. Higher values refer to better model performance and vice versa.

3. Case Study

3.1. Data Description

The data for this study is obtained from the publicly available software project data repository. The characteristics of data are presented in Table 1 where PC1 and Tomcat are publicly accessible from repository of software projects database [23], and the ML, PDE, LC and JDT are from [7].

We consider the following treatment to the datasets before the experiments:

1. Remove all nonnumeric measures.
2. Transform the post-release faults measure (which counts the number of faults in the post-release versions) into the binary class label.

In particular, those containing one or more faults are labelled as DP, whereas those with zero faults are labelled as NDP.

Table 1. Characteristics of datasets.

Datasets	#Modules	#Attribute	DP	NDP	Defect Ratio
PC1	705	38	61	644	10.55
Tomcat	858	22	77	781	10.14
ML	1862	62	245	1617	6.6
PDE	1497	62	209	1288	6.16
LC	691	62	64	627	9.79
JDT	997	62	206	791	3.84

3.2. Experimental Design

In this work, we evaluate the capability of some Machine Learning (ML)-based FS approaches and the logistic regression method to select useful variables at different levels of imbalance, for building the optimal SDP model. The stages implemented in our study is categorized for Case A (FS based on original dataset) and Case B (FS based on balanced dataset) following 4 scenarios as shown in Figure 1. The scenarios are:

- *Scenario* 1 (*S*1): FS technique selected from the original dataset.
- *Scenario* 2 (*S*2): FS technique selected from the sampled dataset.
- *Scenario* 3 (*S*3): MLLR significant features chosen from the original dataset.
- *Scenario* 4 (*S*4): MLLR significant feature selected from the sampled dataset.

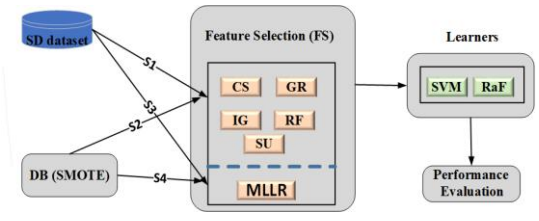


Figure 1. Framework of feature selection and data sampling scenarios.

4. Result and Discussion

This section presents the experimental findings based on SVM and RaF classification models following two Cases:

a) Learning that apply FS considering imbalanced software defect dataset.
b) Learning based on FS after dataset balancing. The research considers six software defect datasets from the depository for the analysis.

The datasets in Case B have class distributions of 50:50 and 65:35, representing the majority NDP and minority DP instances, respectively.

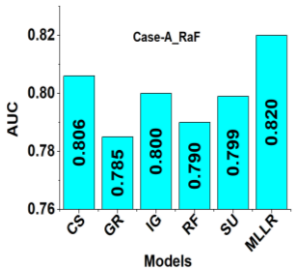
4.1. Case A

The original datasets are imbalanced with different class distributions and sizes. We deploy all the FS techniques considered, including the MLLR, to select features based on the imbalanced data. The optimal subsets found by the individual selection methods are applied as training data for the classification and prediction of software defects. The performance of the models built (SVM and RaF) is captured under AUC, and results are presented in Table 2 and shown in Figure 2. Each sample is denoted with the case letter, followed by the name of the classifier. For example, the RaF classification model for a sample from Case-A is referred to as Case-A RaF.

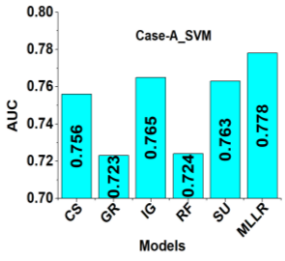
In terms of AUC, it is observed that the MLLR approach records the highest measure according to RaF classification results. AUC is comparable to a rank sum test and quantifies the separability of the classes in a dataset for the classification task. The observation here suggests that the use of MLLR FS technique can guarantee the best feature subset where predictions can be enhanced for all classes in the classification task. In this regard, the performance of CS, IG, and SU FS techniques are also remarkable even though lower than that of MLLR. In general, the worst performing FS techniques in this category are the RF and GR. Referring to Table 2, the performance of the RaF classifier for the saturated model (Normal) is worthy of note. The AUC obtained is marginally (0.02) higher than MLLR. Considering that the performance difference is insignificant relative to the benefits of reduced space and model complexity brought to bear when the classifier developed is based on FS subset, the choice of MLLR for FS is the optimal way to proceed when developing RaF classifier for SDP.

For SVM, a similar performance trend is found in RaF, except that the corresponding AUC measures for the models in Case-A-SVM are relatively lower than that of the models under Case-A-RaF. This observation only demonstrates the superior classification power of RaF to the SVM. Though not a primary objective, this observation is vital for the reason that when developing a model for SDP, the choice of the classification

algorithm is fundamental for improved model performance.



a) Case-A_RaF.

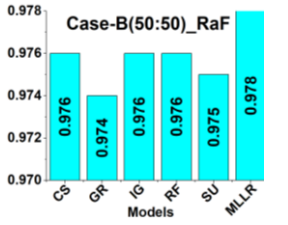


b) Case-A_SVM.

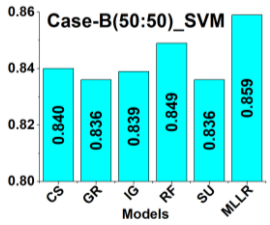
Figure 2. Average AUC over all datasets for the FS methods for Case A.

4.2. Case B

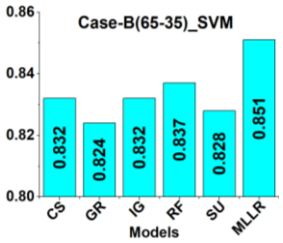
In this case, the five feature ranking methods, including MLLR, are applied after balancing data, and models are built based on the FS subset from the sampled data. For the data balance, we consider different ratios such as 65:35 and 50:50 representing the majority and the minority class samples, respectively. The classification performance (in terms of AUC) of the SVM and RaF classifiers are summarized in Table 2 and demonstrated in Figure 3. It is observed that the AUC measures of the classifiers (SVM and RaF) performance on FS based on the balanced data (50:50) are greater than FS based on sampled data (65:35). Among the FS methods, we find that MLLR performs the best according to the assessment results of both classifiers. The above findings suggest that FS performance is linked to the data imbalance ratio: as the class representation gets at par, the better the performance.



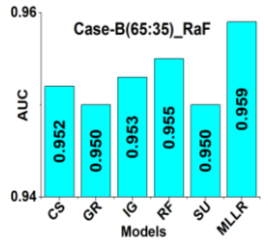
a) Case-B(50:50)_RaF.



b) Case-B(50:50)_SVM.



c) Case-B(65-35)_SVM.



d) Case-B(65:35)_RaF.

Figure 3. Average AUC over all datasets for the FS methods for Case B.

More importantly, the results show that the MLLR FS approach could guarantee improved classification performance of the SDP model. The superiority of RaF over SVM is also observed as the AUCs obtained by all the FS techniques in the RaF classification exceed their corresponding AUCs in SVM.

By the numerical value horizontal contrast in Table 2, we can see that the AUCs obtained by FS with oversampled data by SMOTE are higher than the values realized when the imbalanced data is applied. The suitability of SMOTE data balance has been reported in several studies in the literature [2, 3, 6, 14]. The contribution of our finding, in this regard, is the performance obtained by MLLR, which confirms its compatibility with oversampling and the suitable application for FS in the SDP domain. To quantify the statistical value of the performance difference by the various FS methods, the ANOVA F-test [27] is conducted. The analysis results are presented in the next section.

Table 2. The classification performance over all datasets for the two classifiers.

SVM																						
	Case-A							Case-B(65:35)							Case-B(50:50)						
Dataset	**Normal**	**CS**	**GR**	**IG**	**RF**	**SU**	**MLLR**	**NormaL**	**CS**	**GR**	**IG**	**RF**	**SU**	**MLLR**	**Normal**	**CS**	**GR**	**IG**	**RF**	**SU**	**MLLR**
JDT	0.830	0.829	0.819	0.828	0.827	0.827	0.848	0.870	0.837	0.832	0.838	0.869	0.840	0.879	0.894	0.864	0.850	0.864	0.885	0.851	0.893
LC	0.763	0.829	0.734	0.827	0.633	0.784	0.740	0.854	0.855	0.843	0.855	0.851	0.855	0.874	0.882	0.861	0.864	0.861	0.870	0.861	0.876
ML	0.768	0.658	0.599	0.685	0.740	0.622	0.772	0.818	0.806	0.786	0.806	0.820	0.786	0.822	0.832	0.814	0.815	0.814	0.827	0.815	0.829
PDE	0.729	0.723	0.727	0.707	0.681	0.713	0.761	0.795	0.782	0.786	0.783	0.786	0.783	0.799	0.803	0.788	0.787	0.788	0.791	0.783	0.807
Pc1	0.825	0.761	0.818	0.824	0.770	0.823	0.790	0.888	0.869	0.856	0.869	0.864	0.861	0.881	0.893	0.865	0.865	0.863	0.870	0.861	0.886
Tomcat	0.762	0.737	0.639	0.719	0.693	0.806	0.759	0.855	0.842	0.838	0.842	0.834	0.841	0.852	0.860	0.849	0.833	0.846	0.848	0.844	0.864
Average	0.780	0.756	0.723	0.765	0.724	0.763	0.778	0.847	0.832	0.824	0.832	0.837	0.828	0.851	0.861	0.840	0.836	0.839	0.849	0.836	0.859
RaF																						
	Case-A							Case-B(65:35)							Case-B(50:50)						
Dataset	**Normal**	**CS**	**GR**	**IG**	**RF**	**SU**	**MLLR**	**NormaL**	**CS**	**GR**	**IG**	**RF**	**SU**	**MLLR**	**Normal**	**CS**	**GR**	**IG**	**RF**	**SU**	**MLLR**
JDT	0.888	0.844	0.851	0.841	0.836	0.836	0.889	0.937	0.908	0.907	0.912	0.931	0.906	0.941	0.964	0.954	0.946	0.955	0.960	0.949	0.966
LC	0.805	0.743	0.676	0.745	0.731	0.721	0.779	0.977	0.967	0.962	0.968	0.969	0.961	0.971	0.989	0.985	0.981	0.983	0.984	0.984	0.987
ML	0.828	0.813	0.767	0.803	0.811	0.787	0.809	0.957	0.953	0.944	0.953	0.953	0.951	0.949	0.977	0.977	0.974	0.976	0.975	0.974	0.974
PDE	0.793	0.757	0.738	0.734	0.791	0.754	0.782	0.947	0.942	0.937	0.942	0.944	0.940	0.942	0.973	0.970	0.971	0.970	0.970	0.972	0.970
Pc1	0.890	0.887	0.882	0.878	0.864	0.897	0.851	0.986	0.980	0.981	0.981	0.978	0.982	0.984	0.990	0.987	0.988	0.988	0.987	0.987	0.989
Tomcat	0.835	0.794	0.795	0.798	0.704	0.797	0.810	0.974	0.964	0.967	0.964	0.954	0.961	0.965	0.986	0.982	0.983	0.982	0.981	0.982	0.980
Average	0.840	0.806	0.785	0.800	0.790	0.799	0.820	0.963	0.952	0.950	0.953	0.955	0.950	0.959	0.980	0.976	0.974	0.976	0.976	0.975	0.978

4.3. Statistical Evaluation of Models Performance

In this section, a comparative analysis of the defect prediction performance results for the FS methods is carried out through a one-way Analysis Of Variance (ANOVA) F-test [27]. The test is conducted by using SPSS Statistics 23 package. The ANOVA test evaluates the significance level of the variances in model performances represented by the AUC values. We test and validate the underlying statistical assumptions of ANOVA before the analysis. The factor of most concern (Factor A) considered in this ANOVA test is the modelling that applies the entire feature set, FS subset based on MLLR, and the five feature ranking methods in the two cases of SVM and RaF. The null hypothesis claims that the population means of the entire group are equal, whereas the alternative hypothesis is that at least one pair of means varies. Here, a performance difference is considered statistically significant if the p-value is less than or equal to 0.05. In that case, there is enough evidence to reject the null hypothesis. The ANOVA results for SVM and RaF across the six datasets are presented in Table 3, respectively. The p-value in each table is less than the value specified. This suggests that there is enough evidence to reject the null hypothesis. To further investigate which pairs of means (performance of the models) are statistically significantly different, and which are not, we carry out a multiple pairwise comparisons by applying LSD criterion.

Table 3. One-way ANOVA Results for SVM and RaF.

SVM						**RaF**					
	Sum of Squares	**df**	**Mean Square**	**F**	**Sig.**		**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Between Groups	.233	20	.012	5.261	.000	**Between Groups**	.739	20	.037	28.476	.000
Within Groups	.233	105	.002			**Within Groups**	.136	105	.001		
Total	.466	125				**Total**	.875	125			

The significance level for the LSD test is fixed at $\alpha = 0.05$. Tables 4 shows the multiple comparison results for SVM and RaF classifiers, respectively. The tables display the mean difference and the significance levels for the groups compared. Two means are significantly different if the p-value is below 0.05. To facilitate easy reference, the differences that are significant are shown in boldface. Table 4 presents the comparison results for the saturated model and the models built on FS subsets for SVM classification in Cases-A and B. The results reveal that MLLR, IG, SU and CS (arranged in order of decreasing utility) in Case-A, obtain comparative AUC values with the model built on the entire feature set (Normal). In the case of GR and RF, however, the AUC values relative to Normal are found to be significantly lower. For the comparison of Normal with oversampled data, we find that all FS methods significantly outperformed Normal except CS, IG, GR, and SU at 65:35 class distribution. A similar pattern is observed when MLLR is compared with the rest of the models in this category. Here, all the FS methods that apply data balance in case-B significantly outperformed MLLR without data balance except CS, GR, and SU at 65:35. For the comparison with the other FS methods, it is observed that MLLR performs better than all and significantly outperforms GR and RF. Also, the MLLR selection based on oversampled data (65:35 and 50:50) in case B significantly outperformed all the FS selection methods in case A including the original data (Normal) that apply the entire feature set. The results of RaF shown in Table 4 show some variations with SVM in terms of the FS performance pattern. Generally, FS selection after data balance (65:35) obtained significantly higher AUC values than the use of original imbalanced data. Unlike SVM, the results of RaF demonstrate that all the FS method, together with oversampled data, regardless of the sample distribution, are significantly better than Normal. It is also found that MLLR with or without oversampled data outperformed all the FS methods, including original data. At a class distribution of 65:35, all the FS methods perform notably better than MLLR. At 50:50, however, the performance difference is marginal. The findings here confirm the adverse impact of imbalanced learning for the MLLR FS. In this section, we have shown that the MLLR combined with data balance can identify best feature subsets that can assure better performance of the SDP model.

Table 4. The Multiple compression of one-way ANOVA Results for SVM and RaF.

SVM												
Models		Mean Diff.	Sig.		Mean Diff.	Sig.		Mean Diff.	Sig.		Mean Diff.	Sig.
Normal vs	CS	.02333	.392	Case-B(65:35)_MLLR vs	.07167*	**.010**	MLLR vs	-.00117	.966	Case-B(50:50)_MLLR vs	.07967*	**.004**
	GR	.05683*	**.039**		.09500*	**.001**		.02217	.416		.10300*	**.000**
	IG	.01450	.595		.12850*	**.000**		.05567*	**.043**		.13650*	**.000**
	RF	.05550*	**.044**		.08617*	**.002**		.01333	.625		.09417*	**.001**
	SU	.01700	.533		.12717*	**.000**		.05433*	**.048**		.13517*	**.000**
	MLLR	.00117	.966		.08867*	**.001**		.01583	.561		.09667*	**.001**
	Case-B(65:35)_NormaL	-.06717*	**.015**		.07283*	**.009**		-.06833*	**.013**		.08083*	**.004**
	Case-B(65:35)_CS	-.05233	.057		.00450	.869		-.05350	.052		.01250	.646
	Case-B(65:35)_GR	-.04400	.108		.01933	.478		-.04517	.099		.02733	.317
	Case-B(65:35)_IG	-.05267	.055		.02767	.311		-.05383	.050		.03567	.192
	Case-B(65:35)_RF	-.05783*	**.036**		.01900	.486		-.05900*	**.032**		.02700	.323
	Case-B(65:35)_SU	-.04817	.079		.01383	.612		-.04933	.072		.02183	.423
	Case-B(65:35)_MLLR	-.07167*	**.010**		.02350	.389		-.07283*	**.009**		.03150	.249
	Case-B(50:50)_Normal	-.08117*	**.004**		-.00950	.727		-.08233*	**.003**		.00800	.769
	Case-B(50:50)_CS	-.06067*	**.028**		.01100	.686		-.06183*	**.025**		-.00150	.956
	Case-B(50:50)_GR	-.05617*	**.041**		.01550	.570		-.05733*	**.037**		.01900	.486
	Case-B(50:50)_IG	-.05983*	**.030**		.01183	.664		-.06100*	**.027**		.02350	.389
	Case-B(50:50)_RF	-.06900*	**.013**		.00267	.922		-.07017*	**.011**		.01983	.467
	Case-B(50:50)_SU	-.05633*	**.041**		.01533	.574		-.05750*	**.037**		.01067	.695
	Case-B(50:50)_MLLR	-.07967*	**.004**		-.00800	.769		-.08083*	**.004**		.02333	.392
RaF												
Normal vs	CS	.03350	.110	Case-B(65:35)_MLLR vs	-.01983	.342	MLLR vs	.11883*	**.000**	Case-B(50:50)_MLLR vs	.13783*	**.000**
	GR	.05500*	**.009**		.01367	.513		.15233*	**.000**		.17133*	**.000**
	IG	.04000	.057		.03517	.094		.17383*	**.000**		.19283*	**.000**
	RF	.05033*	**.017**		.02017	.334		.15883*	**.000**		.17783*	**.000**
	SU	.04117	**.050**		.03050	.146		.16917*	**.000**		.18817*	**.000**
	MLLR	.01983	.342		.02133	.307		.16000*	**.000**		.17900*	**.000**
	Case-B(65:35)_NormaL	-.12317*	**.000**		-.14300*	**.000**		.13867*	**.000**		.15767*	**.000**
	Case-B(65:35)_CS	-.11250*	**.000**		-.13233*	**.000**		-.00433	.835		.01467	.482
	Case-B(65:35)_GR	-.10983*	**.000**		-.12967*	**.000**		.00633	.761		.02533	.226
	Case-B(65:35)_IG	-.11350*	**.000**		-.13333*	**.000**		.00900	.666		.02800	.181
	Case-B(65:35)_RF	-.11500*	**.000**		-.13483*	**.000**		.00533	.798		.02433	.245
	Case-B(65:35)_SU	-.11033*	**.000**		-.13017*	**.000**		.00383	.854		.02283	.275
	Case-B(65:35)_MLLR	-.11883*	**.000**		-.13867*	**.000**		.00850	.684		.02750	.189
	Case-B(50:50)_Normal	-.14000*	**.000**		-.15983*	**.000**		-.02117	.311		.01900	.363
	Case-B(50:50)_CS	-.13600*	**.000**		-.15583*	**.000**		-.01717	.411		-.00217	.917
	Case-B(50:50)_GR	-.13400*	**.000**		-.15383*	**.000**		-.01517	.467		.00183	.930
	Case-B(50:50)_IG	-.13583*	**.000**		-.15567*	**.000**		-.01700	.416		.00383	.854
	Case-B(50:50)_RF	-.13633*	**.000**		-.15617*	**.000**		-.01750	.402		.00200	.924
	Case-B(50:50)_SU	-.13483*	**.000**		-.15467*	**.000**		-.01600	.443		.00150	.943
	Case-B(50:50)_MLLR	-.13783*	**.000**		-.15767*	**.000**		-.01900	.363		.00300	.886
***. The mean difference is significant at the 0.05 level.**												

5. Conclusions and Future Work

Developing classification model based on FS subset is an important scientific goal. Standard machine learning algorithms assume balanced training data. Thus, imbalance learning that applies FS subset for classification is a critical challenge because most FS selection methods fail to select the optimal feature subset.

In this study, we presented a statistical method in which FS is situated on the Wald test of significance (at 95% confidence interval) for the MLLR coefficients estimated. The feature subset selected by this approach is referred to as MLLR. For clarity and logical coherence of our presentation, the research method and results discussion were considered under Cases A and B. Case A refers to the learning that applies imbalance data whereas Case B refers to learning from sampled data for the feature subset extraction. The experimental case study is founded on software data from online software historical data depository. We applied the proposed method to select features based on which the SVM and RaF classification models were developed in the context of SDP. The model performances were captured in AUC metrics. To justify the relative advantage of our proposal, we compared the performance of the MLLR method with five feature ranking methods. The ANOVA results remarkably demonstrated the superiority of the proposed MLLR over all the feature ranking methods when either of the classifiers is used in any of the two cases considered. The results also confirmed the findings reported in previous studies in terms of the advantage of FS based on sampled data over FS based on original data. Also, the performance of the defect prediction models was not affected significantly regardless of whether the training data was formed using MLLR FS subset or entire feature set in both sampled and unsampled data. The results of this study point to the fact that selecting the right feature subset for learning in classification for defect prediction is very important. In machine learning classification task, working with a smaller

dimensional feature space data for SDP modelling is more efficient than working with a dataset with high dimensional feature space. Therefore, the study recommends further investigations to tap the useful potentials of MLLR for the software development industry.

For the future research, we will evaluate the effectiveness of the proposed method in different software metrics. In addition, different data sampling techniques and different feature selection techniques will be considered in the context of this study.

Acknowledgements

This work is supported by the National Science Foundation of China (Nos. 61806170, 61876158), Sichuan Science and Technology Program (No. 2019YFS0432) and the Fundamental Research Funds for the Central Universities (No. 2682018CX25).

References

[1] Asadi S., Abdullah R., Safaei M., and Nazir S., "An Integrated SEM-Neural Network Approach for Predicting Determinants of Adoption of Wearable Healthcare Devices," *Mobile Information Systems*, pp. 1-9, 2019.

[2] Bashir K., Li T., and Yohannese C., "An Empirical Study for Enhanced Software Defect Prediction Using A Learning-Based Framework," *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 282-298, 2018.

[3] Bashir K., Li T., Yohannese C., and Mahama Y., "Enhancing Software Defect Prediction Using Supervised-Learning Based Framework," *in Proceedings of 12th International Conference on Intelligent Systems and Knowledge Engineering*, Nanjing, pp. 1-6, 2017.

[4] Chandrashekar G. and Sahin F., "A Survey on Feature Selection Methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16-28, 2014.

[5] Chawla N., Bowyer K., Hall L., and Kegelmeyer W., "SMOTE: Synthetic Minority over Sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321-357, 2002.

[6] Chubato W. and Li T., "A Combined-Learning Based Framework for Improved Software Fault Prediction," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 647-662, 2017.

[7] Czepiel S., "Maximum Likelihood Estimation of Logistic Regression Models: Theory and Implementation," *Scott A Czepiels Homepage*, pp. 1-23, 2009.

[8] DAmbros M., Lanza M., and Robbes R., "Evaluating Defect Prediction Approaches: A Benchmark and an Extensive Comparison," *Empirical Software Engineering*, vol. 17, no. 4-5, pp. 531-577, 2012.

[9] Guyon I. and Elisseeff A., "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[10] Hall T., Beecham S., Bowes D., Gray D., and Counsell S., "A Systematic Literature Review On Fault Prediction Performance in Software Engineering," *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1276-1304, 2012.

[11] Haq A., Li J., Memon M., Malik A., Ahmad T., Ali A., Nazir S., Ahad I., Shahid M., and khan J., "Feature Selection Based on L1-Norm Support Vector Machine and Effective Recognition System for Parkinsons Disease Using Voice Recordings," *IEEE Access*, vol. 7, pp. 37718-37734, 2019.

[12] Haq A., Li J., Memon M., Nazir S., and Sun R., "A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms," *Mobile Information Systems*, pp. 1-21, 2018.

[13] Janecek A., Gansterer W., Demel M., and Ecker G., "On the Relationship between Feature Selection and Classification Accuracy," *Journal of Machine Learning Research*, pp. 90-105, 2008.

[14] Khoshgoftaar T., Gao K., and Seliya N., "Attribute Selection and Imbalanced Data: Problems in Software Defect Prediction," *in Proceedings of 22nd IEEE International Conference on Tools with Artificial Intelligence*, Arras, pp. 137-144, 2010.

[15] Khoshgoftaar T., Gao K., Napolitano A., and Wald R., "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.

[16] Kohavi R. and John G., "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.

[17] Kumar V. and Minz S., "Feature Selection," *Smart CR*, vol. 4, no. 3, pp. 211-229, 2014.

[18] Kuswanto H., Asfihani A., Sarumaha Y., and Ohwada H., "Logistic Regression Ensemble for Predicting Customer Defection with Very Large Sample Size," *Procedia Computer Science*, vol. 72, pp. 86-93, 2015.

[19] Landgrebe T. and Duin R., "Efficient Multiclass Roc Approximation by Decomposition Via Confusion Matrix Perturbation Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 810-822, 2008.

[20] Liu H. and Yu L., "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, 2005.
[21] Mccullagh P., "Generalized Linear Models," *European Journal of Operational Research*, vol. 16, no. 3, pp. 285-292, 1984.
[22] Menzies T., Greenwald J., and Frank A., "Data Mining Static Code Attributes to Learn Defect Predictors," *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 2-13, 2007.
[23] Menzies T., Krishna R., and Pryor D., "The Promise Repository of Empirical Software Engineering Data," *URL http://openscience. Us/repo*, Last Visited, 2015.
[24] Nazir S., Khan M., Anwar S., Khan H., and Nazir M., "A Novel Fuzzy Logic-Based Software Component Selection Modeling," *in Proceedings of International Conference on Information Science and Applications*, Suwon, pp. 1-6, 2012.
[25] Provost F., *in Advances in Distributed and Parallel Knowledge Discovery*, MIT Press, 1999.
[26] Riza L., Zainafif A., Nazir S., and Rasim S., "Fuzzy Rule-Based Classification Systems for the Gender Prediction from Handwriting," *Telkomnika*, vol. 16, no. 6, pp. 2725-2732, 2018.
[27] Weisberg S., Berenson M., Levine D., Goldstein M., Cooper R., and Weekes A., "Intermediate Statistical Methods and Applications: A Computer Package Approach," *Journal of the American Statistical Association*, vol. 79, no. 386, pp. 471, 1983.
[28] Yu Q., Jiang S., Wang R., and Wang H., "A Feature Selection Approach Based on A Similarity Measure for Software Defect Prediction," *Frontiers of Information Technology and Electronic Engineering*, vol. 18, no. 11, pp. 1744-1753, 2017.
[29] Ziani D., "Correlation Dependencies between Variables in Feature Selection on Boolean Symbolic Objects," *The International Arab Journal of Information Technology*, vol. 16, no 6, pp. 1063-1073, 2019.

Kamal Bashir is currently a Ph.D. candidate at the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. He received his MSc. degree in Software Engineering from Khartoum University, Sudan, in 2013. His BSc. degree in Computer Science from Karary University, Sudan, in 2009. His area of research interests Includes Data Mining, Machine Learning, Software Quality Assessment.

Tianrui Li received his B.S. degree, M.S. degree and Ph.D. degree from the Southwest Jiaotong University, China in 1992, 1995 and 2002 respectively. He was a Post-Doctoral Researcher at Belgian Nuclear Research Centre (SCK • CEN), Belgium from 2005-2006, a visiting professor at Hasselt University, Belgium in 2008, the University of Technology Sydney, Australia in 2009 and the University of Regina, Canada in 2014. And, he is presently a Professor and the Director of the Key Lab of Cloud Computing and Intelligent Technique of Sichuan Province, Southwest Jiaotong University, China. Since 2000, he has co-edited 6 books, 10 special issues of international journals, 18 proceedings, received 6 Chinese invention patents and published over 360 research papers.

Mahama Yahaya is currently a Ph.D. candidate at the Transport and Logistics Engineering, Southwest Jiaotong University, Chengdu, China. He received his MSc. degree in Traffic Engineering FROM Southwest Jiaotong University, China. 2018. His BSc. degree in Geodetic Engineering from Kwame Nkrumah University of Science and Technology, Ghana, in 2007. His area of research interests Includes Machine Learning, Roads Construction Project Management, Road Traffic Survey and Data Analysis.